Zahra Ahmadloo, candidate for the degree of Master of Applied Science in Industrial Systems Engineering, has presented a thesis titled, *Predictability of Carbon Dioxide and Ethane Solubility in Ionic Liquids: A Simulation Approach*, in an oral examination held on August 4, 2016. The following committee members have found the thesis acceptable in form and content, and that the candidate demonstrated satisfactory knowledge of the subject material.

External Examiner:  *Dr. Yasser Morgan, Software Systems Engineering*  
Supervisor:  Dr. Amr Henni, Industrial Systems Engineering  
Committee Member:  Dr. Hussameldin Ibrahim, Industrial Systems Engineering  
Committee Member:  Dr. Mohamed Ismail, Industrial Systems Engineering  
Chair of Defense:  Dr. Bruno Dupeyron, Johnson-Shoyama Graduate School of Public Policy  

*Not present at defense*
Abstract

Capturing greenhouse gases using solvents is considered an efficient solution to address climate change and surging anthropogenic activity. In an attempt to find efficient solvents and avoid high experimental costs, predicting the solubility of acid gases in different solvents is attracting attention. Ionic Liquids (ILs) are considered promising solvents for the sweetening of natural gas streams. These ionic liquids can also be used to capture CO$_2$ from flue gases.

This study examines the predictability of modeling the solubility of CO$_2$ and C$_2$H$_6$ in ionic liquids based on some intrinsic properties such as critical temperature, critical pressure and acentric factor as well as process operating conditions such as temperature and pressure. Accordingly, recent experimental data has been collected for 18 ILs which have not been investigated in simulation studies, and are modeled using artificial neural network (ANN) and adaptive neuro fuzzy inference system (ANFIS).

Performance analysis suggests that the solubility of CO$_2$ and C$_2$H$_6$ in an IL, knowing the five selected parameters, can be predicted with satisfactory precision using either ANN or ANFIS. The models presented in this study outperform, in terms of accuracy, previous models in the literature for the specific ionic liquids selected. More specifically, the mean squared errors (MSE) of the generated functions using ANN analysis are $6.93 \times 10^{-5}$ and $7.94 \times 10^{-6}$ for CO$_2$ and C$_2$H$_6$, respectively. The same performance measured for the generated fuzzy inference system gave $6.72 \times 10^{-5}$ and $1.07 \times 10^{-5}$ for CO$_2$ and C$_2$H$_6$, respectively. Finally, individual and mixed-effects of the five variables on the solubility
are assessed. The mixed effect of pressure on other parameters seems to be relatively significant.
This thesis is dedicated to my beloved husband and my wonderful family

for their endless love, enormous support and constant encouragement.
Acknowledgments

I would like to express my gratitude to my supervisor, Dr. Amr Henni, whose guidance, thought provoking suggestion and encouragement supported me throughout my research.

I am deeply indebted to my friends for their constant support and generous assistance and all my friends, who in different ways, helped me in my thesis. I also thank my committee members for taking time to review this manuscript.

I acknowledge the Faculty of Graduate Studies and Research for awarding me a Graduate Scholarship, and providing me with the necessary financial means to complete this study.
# Table of Contents

Abstract.......................................................................................................................................................... i  
Acknowledgments ........................................................................................................................................ iv  
Table of Contents.......................................................................................................................................... v  
List of Tables .................................................................................................................................................. viii  
List of Figures ............................................................................................................................................... ix  
List of Appendices ....................................................................................................................................... xi  
Chapter 1: Introduction................................................................................................................................. 1  
  1.1 Problem Statement & Motivation............................................................................................................. 1  
  1.2 Approaches to CO\textsubscript{2} capturing ............................................................................................. 2  
  1.3 Ionic Liquids (ILs) as efficient solvents ................................................................................................. 4  
  1.4 Objective ............................................................................................................................................... 6  
  1.5 Scope of Study ....................................................................................................................................... 9  
  1.5.1 Artificial Neural Network (ANN) ..................................................................................................... 12  
  1.6 Thesis organization ............................................................................................................................... 13  
Chapter 2: Literature Review......................................................................................................................... 15  
  2.1 Experimental systems modeling ............................................................................................................ 15  
  2.2 Soft computing modeling ....................................................................................................................... 22  
Chapter 3: Methodology ............................................................................................................................... 31  
  3.1 Data Collection .................................................................................................................................... 31  
  3.2 Soft Computing Models ......................................................................................................................... 34
3.2.1 Artificial neural networks................................................................. 34
3.2.2 The feedforward backpropagation neural network algorithm .......... 35
3.2.3 Fuzzy logic....................................................................................... 35
3.3 Artificial Intelligence (AI)................................................................. 35
3.3.1 Artificial neural network overview.................................................. 37
3.3.2 Feed-forward neural network model .............................................. 39
3.3.3 Neural Network Training................................................................. 42
3.4 Fuzzy Logic ....................................................................................... 46
3.4.1 Fuzzy logic foundations................................................................. 50
3.4.2 Membership functions ................................................................. 50
3.4.3 Fuzzy inference process................................................................. 52
3.4.4 Neuro-adaptive learning ............................................................... 56

Chapter 4: Results and Discussion .......................................................... 59
4.1 Artificial Intelligence Fitting Results ................................................ 59
4.1.1 CO₂ solubility using ANN ............................................................. 59
4.1.2 CO₂ solubility using ANFIS .......................................................... 66
4.1.3 ANN vs ANFIS comparison......................................................... 76
4.1.4 Ethane solubility using ANN......................................................... 77
4.1.5 Ethane solubility using ANFIS ...................................................... 83
4.1.6 ANN vs ANFIS comparison......................................................... 93
4.2 Summary.................................................................................................................. 94

Chapter 5: Conclusion..................................................................................................... 96

References......................................................................................................................... 99
List of Tables

Table 1-1. ILs used in this study for modeling ......................................................... 8
Table 3-1. Ranges of temperature, pressure and CO\textsubscript{2} solubility while using certain ILs. . 31
Table 3-2. Ranges of temperature, pressure and C\textsubscript{2}H\textsubscript{6} solubility while using certain ILs.33
Table 3-3. Critical properties and acentric factors of ILs used in this study. ................ 33
Table 4-1. Weight and bias of the ANN (First hidden layer) for CO\textsubscript{2} ..................... 60
Table 4-2. Weight and bias for the ANN for the second hidden layer for CO\textsubscript{2} ........... 61
Table 4-3. Details of the trained ANFIS for CO\textsubscript{2} ........................................... 66
Table 4-4. Order of solubility & the critical properties of the ILs ................................. 74
Table 4-5. CO\textsubscript{2} ANFIS vs ANN .......................................................... 76
Table 4-6. Comparison of the performance of the other presented CO\textsubscript{2} models in the literatures ......................................................................................... 77
Table 4-7. Weight and bias values of the ethane ANN model ....................................... 79
Table 4-8. Details of the ethane trained ANFIS ....................................................... 84
Table 4-9. Order of solubility & the critical properties of the ILs ................................. 91
Table 4-10. Ethane ANFIS vs ANN ........................................................................ 93
Table 4-11. Comparison of AAD\% values for SRK and PR EOS and proposed ANN & ANFIS approach ....................................................................................... 94
List of Figures

Figure 1-1. Illustration of a post-combustion process ........................................... 3

Figure 3-1. A typical feedforward neural network. .................................................... 38

Figure 3-2. Six most popular activation functions used in neural network models ....... 41

Figure 3-3. Interpreting the fuzzy inference diagram .............................................. 55

Figure 4-1. CO₂ Function fitting neural network ..................................................... 60

Figure 4-2. CO₂ ANN Training state ....................................................................... 62

Figure 4-3. CO₂ performance improvement ............................................................. 62

Figure 4-4. CO₂ Training, validation and test regression .......................................... 63

Figure 4-5. CO₂ target-output regression ................................................................. 64

Figure 4-6. CO₂ Error histogram .......................................................................... 65

Figure 4-7. CO₂ training performance of the generated FIS ................................... 67

Figure 4-8. CO₂ checking performance of the generated FIS ................................ 67

Figure 4-9. CO₂ regression plot: Target vs Output ................................................. 68

Figure 4-10. CO₂ error histogram .......................................................................... 69

Figure 4-11. Fuzzy rules on the input and output variables for CO₂ .................... 70

Figure 4-12. Fuzzy rule for the ANFIS model for CO₂ .......................................... 71

Figure 4-13. Sensitivity analysis of solubility of CO₂ to each variable ................. 72

Figure 4-14. Solubility of carbon dioxide versus pressure at 313.15 K ............... 73

Figure 4-15. Surface view of CO₂ input parameters to the output .................... 75

Figure 4-16. Ethane function fitting neural network ............................................. 78

Figure 4-17. ANN training state for ethane parameters ....................................... 79
Figure 4-18. Ethane performance improvement ................................................................. 80
Figure 4-19. Ethane training, validation and test regression ............................................. 81
Figure 4-20. Ethane target-output regression ...................................................................... 82
Figure 4-21. Ethane error histogram .................................................................................. 83
Figure 4-22. \( \text{C}_2\text{H}_6 \) training performance of the generated FIS .................................. 85
Figure 4-23. \( \text{C}_2\text{H}_6 \) checking performance of the generated FIS .................................. 85
Figure 4-24. \( \text{C}_2\text{H}_6 \) regression plot: Target vs Output .................................................... 86
Figure 4-25. \( \text{C}_2\text{H}_6 \) error histogram .................................................................................. 87
Figure 4-26. Fuzzy rules on the input and output variables of ethane .............................. 88
Figure 4-27. Fuzzy rule for the ethane ANFIS model ....................................................... 89
Figure 4-28. Sensitivity of solubility of ethane to each parameter .................................... 90
Figure 4-29. Solubility of ethane versus pressure at 303.15 K ....................................... 91
Figure 4-30. Surface view of ethane input parameters to the output .............................. 92
List of Appendices

Appendix A: Neural Network Fitting Script Program (Carbon Dioxide) ............ 106
Appendix B: Neural Network Performance Table (Carbon Dioxide) ................. 107
Appendix C: Neural Network Predictor Function (Carbon Dioxide) .................. 110
Appendix D: Neural Network Fitting Script Program (Ethane) ...................... 113
Appendix E: Neural Network Performance Table (Ethane) ............................ 114
Appendix F: Neural Network Predictor Function (Ethane) .............................. 118
Chapter 1: Introduction

In this chapter, the problem is explained and an introduction to the main technologies used for CO2 capturing and the utilized methods to solve the problem are presented. Subsequently, the objectives of the study and the methodologies utilized to achieve our goal are discussed. The last section of this chapter describes the thesis organization.

1.1 Problem Statement & Motivation

Nowadays, burning fossil fuels such as oil, coal and gas has been meeting the demand of human beings to produce energy all around the world. Some other sources of renewable energy have been introduced to the world market such as solar, wind, hydro and geothermal, but, it seems that there is more impact in using fossil fuels than renewable energies as the progress made in using these sources of energies is rather slow. As a matter of fact, the growing use of fossil energies and the rising amount of greenhouse gases emissions will be one of the most challenging issues facing Humanity in the near future. Carbon dioxide (CO2), Methane (CH4), Nitrous Oxide (N2O), Hydro-Fluorocarbon (PFC3), Chlorofluorocarbons (CFC3), Water Vapour (H2O), Sulphur Hexafluoride (SF6) and Ethane (C2H6) are known as major greenhouse gases. Electricity, heat generation, transportation and industry contribute to the fact that CO2 is the most greenhouse gas emitted globally, despite the fact that other greenhouse gases play their own role in global warming but CO2 is the main contributor to the climate change (Henni, 2002).
As the amount of emitted greenhouse gases in the atmosphere has been increasing, the world is facing some important environmental issues such as global warming, rise in sea levels, intense floods and climate change (Henni, 2002). Hence, due to the severity of this issue, governments and environmental organizations have suggested imposing and enforcing strict regulations to mitigate the negative outcomes of using fossil fuels. According to the Canadian government (Canada, 2016), Canada’s total greenhouse gas (GHG) emissions in 2013 were 726 megatons (Mt) of carbon dioxide ($\text{CO}_2$) or 18 % (113 Mt) above the 1990 emissions of 613 Mt. Power plants, and specifically coal fired plants, are the ideal place to reduce emissions and capture these vast amounts of greenhouse gases. It should be mentioned that around 25% of the world’s energy demand is provided by coal. Along with the other GHG control projects, CCS (Carbon Dioxide Capture and Sequestration) should be widely implemented to reduce the released $\text{CO}_2$ by reducing $\text{CO}_2$ emissions mostly from power plants and other industrial sources. There are several approaches to capture $\text{CO}_2$ which are briefly explained in the following section.

1.2 Approaches to $\text{CO}_2$ capturing

Capturing $\text{CO}_2$ from major sources such as power plants and natural gas streams must be done on a continuous basis. Hence, capturing $\text{CO}_2$ requires a low cost and efficient technology. Current technologies need much energy to operate the $\text{CO}_2$ capture units and may also have harmful side effect on the environment. Oxy-fuel combustion, pre-combustion, and post-combustion are considered as the most promising $\text{CO}_2$ capture technologies.
Among the above-mentioned technologies, one of the processes which can significantly mitigate the CO\textsubscript{2} emission from the coal fired power plants is the post combustion CO\textsubscript{2} capture option. As presented in Figure 1.1, in the post combustion process, after burning the fuel and the air, CO\textsubscript{2} is captured.

![Figure 1-1. Illustration of a post-combustion process](image)

The most common commercially available and one of the most promising methods for CO\textsubscript{2} capture, is the use of aqueous amines. In the absorption process, CO\textsubscript{2} is contacted with the solvent in order to cause the removal of the solute (CO\textsubscript{2}), from the gas phase. In a later stage, the absorbed CO\textsubscript{2} will be separated from the solvent in the stripper by heating up the solution and the regenerated liquid solvent will be returned to the absorption system. In most commercially available units, the amine (solvent) enters from the top of the absorber or stripper unit and reacts chemically with the rising gas stream so that the solute (CO\textsubscript{2}) can be transferred from the gas stream to the solvent (amine). The only difficulty that industry is facing with this known technology is that regenerating the
solvents requires a large amount of energy, and solvent degradation due to oxidation and the high temperature used in the stripping section. Moreover, corrosion is the other main factor that needs to be taken it into account by adding inhibitors. In order to address all the above mentioned drawbacks in using currently available amines, researchers are investigating more energy efficient solvents that possess the appropriate physical and chemical properties.

In addition to the traditional physical and chemical solvents used in CO₂ capture, ionic liquids made an apparition as promising new solvents in the absorption process for CO₂ capture, as will be explained in detail in the next section.

1.3 Ionic Liquids (ILs) as efficient solvents

Ionic liquids are a category of salts in which the ions are poorly coordinated organic cations paired with inorganic or organic anions, this allows these solvents to be liquids below 100°C, or even at room temperature (room temperature ionic liquids, RTILs). The benefit of using ionic liquids is their high thermal and chemical stability and low vapour pressure which makes them non-volatile (Zhijun, 2012). As a result, solvent loss due to degradation is minimized, and more importantly, energy can be saved in the regeneration step making it a cheaper process than of chemical solvents. In recent years, studies related to the possible application of ionic liquids in CO₂ capture process has dramatically increased in terms of both experimental and theoretical studies. The first research group which synthesized [NH₂p-bim] [BF₄] as a substance and used the amine function in the liquid cation were Bates et al. (2002). According to the obtained results, CO₂ separation was feasible without any reduction in stability of ionic liquid. 0.5 mol
CO₂/ mol IL was the ratio of absorbed CO₂ after exposing the ionic liquid to CO₂ at atmospheric pressure. The exposed ionic liquid remained stable after five cycles of absorption and desorption (Bates, 2002). The effect of structure, properties and molecular interactions of ionic liquids on solubility and selectivity for CO₂ capture utilizing the thermodynamic model in the software COnductor-like Screening MOdel for Realistic Solvents (COSMO-RS), as well as the relationship between the properties of ionic liquids and the Henry’s law constants were studied by Sumon and Henni (2011). Results show that by increasing the molar volume, the Henry’s law constants decreased as well as an increase in polarity of ionic liquids was observed. Furthermore, by increasing the temperature, the solubility of CO₂ and selectivity both decreased. To capture CO₂, a number of ionic liquids, based on the Hildebrand solubility parameter (δ) (calculated by molecular dynamic simulation utilizing Material Studio), were examined by Sistla (2012). The computed (δ) values using the correlation and the direct molecular simulation perfectly agreed with each other. Hence, the suggested correlation function was good enough for the prediction of (δ) values. The study of absorption of CO₂ in some organic solvents like DMA, NMP, DMSO, and DMF with 2-hydroxyethyl ammonium lactate (HEAL) has been experimentally and theoretically performed by Hwang (2009). In this study, based on the film theory and using the nonlinear reaction rate equation according to the Zwitterions mechanism, a CO₂ absorption mathematical model along with the CO₂ reaction with HEAL was presented. The experimental data of the enhancement factor as for carbon dioxide absorption at various temperatures, solvents and their different HEAL concentration were well correlated by the suggested model. The use of the Henry’s law for the correlation of the gathered data for solubility of CO₂ in sulfonate ionic liquids at high
pressure was studied by Zhang et al. (2005). Shiflett et al. (2006) used the Non Random Two Liquid (NRTL) model to correlate the solubility of tetrafluoroethane in seven ionic liquids. Many other studies and investigations have been initiated to study of CO$_2$ capture using ionic liquids such as: CO$_2$ capture by enzymatic bioconversion in a membrane contactor with task specific ionic liquids (Neves et al., 2012), a Supported Liquid Membrane (SLM) based on a task specific ionic liquid to achieve the selective and facilitated CO$_2$ transport through the membrane (Hanioka et al., 2008), CO$_2$ capture by a dual amino ionic liquid with amino-functionalized imidazolium cation and taurine anion (Xue et al., 2011). Carbon dioxide separation performance of polymeric ionic liquid composite membranes based on poly (diallyldimethylammonium) bis (trifluoro-methylsulfonyl) imide, poly ([pyr11][NTf2]) (Tome et al., 2013), performance of multiple CO$_2$ binding organic liquids (CO$_2$BOLs) as solvent systems for post combustion CO$_2$ capture (Heldebrant et al., 2011), and new criteria combining efficiency, greenness, and economics for screening ionic liquids for CO$_2$ capture (Chen et al., 2013) have been proposed.

1.4 Objective

As mentioned in the previous section, due to the highly effective characteristics of ILs in capturing acid gases in many chemical industries, measuring the solubility of acid gases such as CO$_2$ and C$_2$H$_6$ in different ionic liquids have attracted much attention to this area. But beside all these benefits, the experimental measurements have some restrictions based on the utilized equipment and laboratory set up, they are time consuming and fairly expensive. Therefore, it would be wise and logical to develop effective predictive models for the estimation of the solubility values with high precision that saves cost and time.
The Peng–Robinson equation of state (EOS), a generic van der Waals EOS, Henry’s law, an extended Henry’s law included Pitzer’s virial expansion for the excess Gibbs energy, and the general Redlich–Kwong cubic EOS, modified perturbed hard sphere chain (PHSC) EOS, soft-SAFT EOS, the truncated perturbed chain polar statistical associating fluid theory PC-PSAFT, the perturbed chain statistical associating fluid theory (PC-SAFT), the statistical associating fluid theory for potentials of variable range (SAFT-VR), group contribution models, and simple correlations are among the most widely used methods which researchers used to model the gas/ILs systems based on the aforementioned models (Jou and Mather, 2007; Mortazavi-Manesh et al., 2011). One of the most important drawbacks of applying the reported methods, is that they are restricted to correlation of these gases and ILs systems (Safamirzaei and Modarress, 2012).

In this study, the Artificial Neural Network (ANN) and Adaptive Neuro Inference Systems (ANFIS) have been utilized for modeling and predicting the solubility of carbon dioxide and ethane in 18 different ILs for which the critical properties of the ILs (pressure and temperature) are used as input parameters and the solubility of the gases in ILs is used as the target of the data used for modeling. In this study, two models are used correlate for the solubility of the carbon dioxide and ethane. To the best of the author’s knowledge neither the gathered experimental data for the mentioned gases (CO₂ and C₂H₆), nor the used ILs have been investigated using the simulation methods used here.

The following points highlight the sub-objectives of the research:
• Evaluating and validating the accuracy of Artificial Neural Network, Adapted Neuro Fuzzy Inference System modeling for the estimation of the solubility of the gases in the 18 specific ionic liquids listed below (Table 1-1).

• Modeling with the ANN, and ANFIS to implement a systematic method for the prediction of carbon dioxide solubility.

• The critical properties of the studied ionic liquids (critical pressure and temperature) and acentric factor were estimated, and the temperature and pressure are used as input values, while the solubility of the gases are considered as the outcome values of this study.

• Selecting the best network which can predict the output values based on the optimization procedure.

• Comparing the predicted values using ANN to the used thermodynamic models (goodness of fit).

Table 1-1. List of ionic liquids used in this study

<table>
<thead>
<tr>
<th>#</th>
<th>Ionic Liquids</th>
<th>Abbreviations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1-Ethyl-3-methylimidazolium L-(+)-lactate</td>
<td>[EMIM] [LACTATE]</td>
</tr>
<tr>
<td>2</td>
<td>3-Methyl-1-propylpyridinium bis[(trifluoromethyl)sulfonyl] imide</td>
<td>[PMPY] [TF2N]</td>
</tr>
<tr>
<td>3</td>
<td>1(4Sulfobutyl)3methylimidazoliumbis(trifluoromethanesulfonyl) Imide</td>
<td>[(CH2)4SO3HMIm][TF2N]</td>
</tr>
<tr>
<td>4</td>
<td>1-(4-Sulfobutyl)-3-methylimidazolium hydrogen sulfate</td>
<td>[(CH2)4SO3HM] [HSO4]</td>
</tr>
<tr>
<td>5</td>
<td>1,2,3-Tris(diethylamino) cyclopropenylium dicyan amide</td>
<td>[TDC] [DCN]</td>
</tr>
<tr>
<td>6</td>
<td>Ethyldimethylpropylammonium bis(trifluoro methyl sulfonyl)imide</td>
<td>[EMMP] [TF2N]</td>
</tr>
<tr>
<td>7</td>
<td>1,2,3-Tris(diethylamino) cyclopropenylium bis(trifluoromethanesulfonyl) imide</td>
<td>[TDC] [TF2N]</td>
</tr>
</tbody>
</table>
8. N-methyl-n-pentylpyrrolidinium bis(trifluoromethylsulfonyl)imide \([\text{p}(5)\text{mpyrr}][\text{Tf2}]

9. 1-Butyl-2,3-dimethylimidazolium bis(trifluoromethyl sulfonyl)imide \([\text{bmmim}][\text{Tf2N}]

10. Tributylmethylphosphonium bis(trifluoromethyl sulfonyl)imide \([\text{P4441}][\text{Tf2N}]

11. Triethylsulfoniumbis(trifluoromethylsulfonyl)imide \([\text{S222}][\text{Tf2N}]

12. Diethylmethyl(2methoxyethyl)ammoniumbis(trifluoromethylsulfonyl)imide \([\text{deme}][\text{Tf2N}]

13. 1-Propyl-3-methylimidazolium bis(trifluoromethylsulfonyl)imide \([\text{pmim}][\text{Tf2N}]

14. 1-Allyl-3-methylimidazolium bis-(trifluoromethyl sulfonyl)-imide \([\text{amim}][\text{Tf2N}]

15. 1-Butyl-4-methylpyridinium tetrafluoroborate \([\text{4mbp}][\text{BF4}]

16. 1-Methyl-1-propylpiperidinium bis(fluorosulfonyl)imide \([\text{mp}(3)\text{pip}][\text{FSI}]

17. N-propyl- n-methylpyrrolidinium bis(fluorosulfonyl)imide \([\text{mp}(3)\text{pyrr}][\text{FSI}]

18. N,n-diethyl-n-methyl-n-propylammonium bis(fluorosulfonyl)imide \([\text{N1223}][\text{FSI}]

1.5 Scope of Study

For the purpose of parameters calculation, models are used for estimation rather than observation, to reduce time and cost. To create a model, previously observed output data and other related input variables are required. The related input variables measurements must be easier and cheaper to calculate or measure compared to the measured response variables. For the required problem a suitable modeling technique and predictor variables have to be found from background knowledge and literature. Modellers may try many techniques and evaluate their performances to select the best prediction model (Wang et al., 2008). There are different types of modelings and methods for acid gas soubility modeling, such as thermodynamic and mathematical. The thermodynamic models
represent the system with a smaller sized model to analyze and measure the required parameters while mathematical models represent systems with numerical values which can be adopted to quantify different parameters and their components. Recently, due the development of more powerful computers and software programs, mathematical models have become more popular compared to other types of models (Benedini & Tsakiris, 2013).

As for the thermodynamic modeling, the exact experimental procedure and the obtained data are required which makes this type of modeling expensive, time consuming and in some cases, the experiments are restricted to the setup conditions of the equipments (Li et al., 1998; Huang, et al., 2010).

Inferential statistics generate predictions and comparisons about population (larger amounts of data) using information collected from a sample (smaller in size) of that population. Thus, an inferential statistical model defines a mathematical relationship between two sets of variables, the predictors and the output to be used for predictions and estimations (Lyman & Longnecker, 2010). In general, this type of model has been found to be ineffective for nonlinear and complex systems (Kisi, 2004; Hamaamin et al., 2013).

In the last three decades, different soft modeling methods have been developed that integrate biological structures and computing techniques such as artificial neural networks (ANN), genetic algorithms, fuzzy logic, and fusion methods (Li et al., 1998; Huang et al., 2010). With increases in the calls for limiting acid gas emissions, along with growing attention to climate changes and anthropogenic activity, the ability to accurately predict the solubility conditions continue to be invaluable (Smakhtin, 2001). However,
conventional methods may not remain cost-effective or robust compared to alternative soft computing methods (Huang et al., 2010).

Artificial Neural Network and Adaptive Network-based Fuzzy Inference System are used in this study to model the gathered experimental data for both carbon dioxide and ethane. In the following section, a brief background about the fuzzy set theory and an introduction to the utilized Artificial Neural Network (ANN) method and the Adaptive Network-based Fuzzy Inference System (ANFIS) are presented.

**Neuro-Adaptive Learning**

The neuro-adaptive learning method works similarly to that of neural networks. Neuro-adaptive learning techniques provide a method for the fuzzy modeling procedure to learn information about a data set. In this case, the membership function parameters are set so that the associated fuzzy inference system is able to track the given input/output data as much as possible. A network-type structure, similar to that of a neural network that maps inputs through input membership functions and associated parameters and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map (Rutkowski & Kacprezyk, 2002).

The parameters associated with the membership functions changes through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. When the gradient vector is obtained, any of several optimization routines can be applied in order to adjust the parameters to reduce some error measures. This error is usually
defined by the sum of the squared difference between actual and desired outputs. Either back propagation or a combination of least squares estimation and back propagation can be used for membership function parameter estimation (Rutkowski & Kacprezyk, 2002).

1.5.1 Artificial Neural Network (ANN)

In Artificial Neural Networks (ANNs), there are some elements that have been inspired by the human biological nervous systems, these elements which are called neurons work in a parallel computational strategy (Hagan et al., 2002).

There is considerable attention given to studies that model experimental work related to the solubility of gases in ionic liquids. Most of the experimental studies in this field of interest are along with the different methods of modeling in which the methods are restricted by the pre-defined set-up of the utilized equipment and not suitable for the real time application and besides all this, being time consuming can be considered as the drawbacks of this approach. All the suggested models such as: equation of state, group contribution methods, activity models, extended Henry’s law and simple correlations require adjustable parameters which should be optimized based on experimental data points. All the recommended models are useless without having some experimental data and the pre-defined set-up parameters (Safamirzaei et al., 2012). As a matter of fact, ANNs are based on the thermodynamic systems which are able to model the non-linear systems (Moghaddasi et al., 2010). Artificial Neural Networks (ANNs) approach of modeling applies the weight and connection among the input and target variables, and all the complexity of the process are included in a modeling using ANNs. Neural network is able to be trained and learn the behaviour of the generated data by a system (Moghaddasi et al., 2009). In comparison to the other used methods, the simplicity, speed in learning
the behaviour of the data and generating the related function to model the data and the capacity of ANNs are mentioned as the pros points of the Artificial Neural Networks.

**Process used in the ANN**

Training the Artificial Neural Network includes learning and recalling. For the learning step, the network needs to feed the right information with the right parameters. Then the network, based on the iterative learning process will learn to calculate the outputs. While for the recalling step, the network computed the output of the testing data set based on the learning results. This can be used to assess the learning performance.

### 1.6 Thesis organization

In this section, the organization of the study is presented. In chapter 2, previous studies regarding the prediction of solubility of gases in different ILs are discussed. Chapter 3 explains the methodology and procedure of the work. It starts with a description of the collected solubility data points in this study. Neural network and Inference System generation methodologies using ANFIS and ANN are also discussed in this chapter.

The results of ANN and ANFIS modeling, and ANN function generation are presented in chapter 4. In sections 4.1.1 and 4.1.2, the analysis and the results of the generated FIS and ANN models for CO$_2$ are presented. Then, a comparison between the ANN and FIS results is presented in section 4.1.3. The results of the generated models for the solubility of ethane and the comparison between the FIS and ANN models are presented in sections 4.1.4 to 4.1.6
Chapter 5 deals with the summary and conclusion, and the six appendices of the study are presented in the last section of the manuscript.
Chapter 2: Literature Review

In this chapter, a comprehensive review is presented and deals with published modeling studies of the solubility of CO$_2$ and ethane in ionic liquids.

2.1 Experimental systems modeling

The discovery of various ionic liquids which can effectively absorb CO$_2$ from flue gases and natural gas was a motivation to study the phase behavior of carbon dioxide with ionic liquids. A convenient thermodynamic model such as an equation of state is required for predicting structure–property relationship. Bazargani (2015) selected the Perturbed Hard Sphere Chain Equation of State (PHSC) to model CO$_2$ absorption in a series of ionic liquids including 1-hexyl-3- methyl- imidazolium tetrafluoroborate [HMIM][BF$_4$], 1-hexyl-3-methylimidazolium hexafluorophosphate [HMIM][PF$_6$], 1-hexyl-3-methyl imidazolium bis (trifluoromethylsulfonyl) imide [HMIM][Tf$_2$N], 1-octyl-3-methyl imidazolium tetrafluoroborate [OMIM][BF$_4$], 1-octyl-3-methylimidazolium hexafluorophosphate [OMIM][PF$_6$] and 1-octyl-3-methyl-imidazolium bis (trifluoromethyl-sulfonyl) imide [OMIM][Tf$_2$N]. Using three molecular-based parameters plus regression of only one binary interaction parameter was sufficient for PHSC EoS to achieve the modeling purposes.

While the solubility of carbon dioxide in 1-ethyl-3-methyl imidazolium L-(+)-lactate ([EMIM][LACTATE]), 3-methyl-1-propyl pyridinium bis[(trifluoro methyl sulfonyl) imide ([PMPY][TF$_2$N]), 1-(4-sulfobutyl)-3-methyl imidazolium bis (trifluoro methanesulfonyle) imide ([(CH$_2$)$_4$SO$_3$HMIm] [TF$_2$N]), 1-(4-sulfobutyl)-3-methyl
imidazolium hydrogen sulfate ([(CH₂)₄SO₃HMIm] [HSO₄]), has been experimentally studied by Zoubeik (2016) using a gravimetric microbalance method. Carbon dioxide + IL systems were studied at (313.15, 323.15 and 333.15) K over a pressure range of 100 mbar to 20000 mbar. Experimental densities, Henry's Law constants, entropies and enthalpies of absorption were also reported. The results obtained showed that CO₂ solubility diminished in the following sequence: [PMPY] [Tf₂N] > [EMIM] [LACTATE] > [(CH₂)₄SO₃HMIm] [TF₂N] > [(CH₂)₄SO₃HMIm] [HSO₄]. It was found that [PMPY] [Tf₂N] shows comparable CO₂ solubility with ionic liquids that are considered promising such as [HMIM] [Tf₂N], which makes this ionic liquid an attractive solvent for gas separation processes. CO₂ solubility in the ionic liquids was well correlated using Peng–Robinson equation of state with a quadratic mixing rule and the non-random two-liquid (NRTL) model.

High-pressure solubility of carbon dioxide in ionic liquids for tris (pentafluoroethyl) trifluorophosphate ([eFAP]) anion, combined with 1-ethyl-3-methyl imidazolium ([emim]), 1-butyl-3-methylimidazolium ([bmim]) and 1-hexyl-3-methyl imidazolium ([hmim]) cation, was determined experimentally by da Ponte (2016) at 313.15 K and at pressures up to 10 MPa. Measurements of the volume expansion of the liquid phase upon carbon dioxide dissolution are also reported. The obtained results show that the solubility of CO₂ in ionic liquids with the fluoroalkyl phosphate anion, as well as the CO₂-induced volume expansion of the liquid, are among the highest so far reported for ionic liquids where CO₂ dissolves through “physical” mechanisms.

The other three bis(trifluoromethylsulfonyl)imide-based ionic liquids i. e. n-methyl-n-pentyl pyrrolidinium bis (trifluoromethylsulfonyl)imide ([p(5)mpyr][Tf₂N]), 1-butyl-
2,3 dimethyl imidazolium bis (trifluoromethyl sulfonyl) imide ([bmmim][Tf$_2$N]), and tributyl- methyl phosphonium bis (trifluoromethyl sulfonyl) imide ([P$_{4441}$][Tf$_2$N]) were measured by Tagiuri et al. (2014) at 298.15, 313.15, and 343.15 K and pressures up to 1.9 MPa. The data were well correlated with the Soave–Redlich–Kwong equation of state (SRK-EoS). Henry's law constants for CO$_2$ in the three ILs ([p(5)mpyrr][Tf$_2$N], [bmmim][Tf$_2$N], and [P$_{4441}$][Tf$_2$N]) at 298.15 K were (3.1, 3.5, and 3.3) MPa, respectively.

Experimental values for the solubility of carbon dioxide, methane and nitrous oxide in ionic liquids are measured at different temperatures and at pressures close to atmospheric are considered as the other works that are available in the literature. Carbon dioxide is the most soluble gas studied with mole fraction solubility of the order of 10$^{-2}$. Nitrous oxide is one order of magnitude less soluble than carbon dioxide, whereas methane is the least soluble gas. A group contribution method aimed at estimating the Henry's law constant of CO$_2$ in ILs as a function of temperature was developed. Such a method makes it possible to predict the solubility of CO$_2$ in ionic liquids when experimental data are unavailable (Chen et al., 2014).

Moreover, the solubility of carbon dioxide in three different ionic liquids {1,2,3-Tris(diethylamino) cyclopropenylium dicyanamide [TDC] [DCN], ethylidimethyl propylammonium bis (trifluoromethyl sulfonyl) imide [EMMP] [TF$_2$N], and 1,2,3 tris (diethylamino) cyclopropenylium bis (trifluoro methane sulfonyl) imide [TDC] [TF$_2$N]} has been studied experimentally using a gravimetric microbalance. CO$_2$ + IL systems were studied at 313.15, 323.15 and 333.15 K and over a pressure range of 10.1 to 20 bar. The experimental density, Henry's law constant, entropy and the enthalpy of absorption
for CO$_2$ are also reported. The results obtained showed that CO$_2$ solubility diminished in the following sequence: [TDC][TF$_2$N] > [EMMP] [TF2N] > [TDC][DCN]. It was found that [TDC][TF$_2$N] shows comparable CO$_2$ solubility to other well-known ionic liquids such as [hmim][TF$_2$N], which makes it an attractive solvent for CO$_2$ removal processes. The experimental solubility of CO$_2$ in the three solvents were well correlated using the standard Peng–Robinson (PR) EoS, the Soave–Redlich–Kwong (SRK) EoS with quadratic mixing rules, and the Non-Random Two-Liquid (NRTL) model. (Zoubeik & Henni, 2014)

1-Butyl-3-methylimidazolium bis (trifluoro methy lsulfonyl) imide, 1-ethyl-3-methyl imidazolium diethyl phosphate, trihexyl tetrade cyl phosphonium dicyanamide, and trihexyl tetrade cyl phosphonium bis (2,4,4-trimethylpentyl) phosphinate were studied for the solubility of gas mixtures containing carbon dioxide (CO$_2$) and methane (CH$_4$) by Ramdin et al. (2014). The experiments involve bubble-point measurements in the temperature range 303.15 to 363.15 K and at pressures up to 14 MPa using a visual synthetic method. The influence of the gas composition on the bubble-point pressure was investigated for three gas mixtures containing 25 mol% CO$_2$–75 mol% CH$_4$, 50 mol% CO$_2$–50 mol% CH$_4$ and 75 mol% CO$_2$–25 mol% CH$_4$. The Peng–Robinson (PR) equation of state (EoS) in combination with van der Waals mixing rules, using only binary interaction parameters, was applied to predict the ternary vapor–liquid equilibrium data and to extract the real CO$_2$/CH$_4$ selectivity (i.e., the selectivity of CO$_2$ in the presence of CH$_4$ in the ternary system CO$_2$–CH$_4$–IL). This real selectivity does not differ significantly from the ideal selectivity (i.e., the ratio of the pure gas Henry constants) even for mole fractions of IL as low as 0.7 and regardless of the gas phase composition.
The solubility in three ionic liquids which contained three different numbers of cyanide anions, 1-ethyl-3-methylimidazolium thiocyanate ([c$_2$mim][SCN]), 1-ethyl-3-methylimidazolium dicyanamide ([c$_2$mim][N(CN)$_2$]), 1-ethyl-3-methyl imidazolium tricyanomethanide ([c$_2$mim][C(CN)$_3$]) was measured in order to investigate the effects of cyanide anions on the solubility of CO$_2$. The solubility of CO$_2$ was determined by measuring the bubble point pressure at a temperature range from 303.15 to 373.15 K in 10 K intervals. Also, the measured data were correlated with the PR-EoS (Peng–Robinson equation of state) incorporated with the conventional van der Waals one fluid mixing rule. The critical properties of ionic liquids were estimated using the modified Lydersen–Joback–Reid method. As a result, the calculated data were comparatively well correlated to the experimental results and, as is commonly known, the solubility of CO$_2$ was observed to increase with increasing pressure and with decreasing temperature. Moreover, the results also show that the highest solubility was obtained by [c$_2$mim][C(CN)$_3$] among those three experimented ionic liquids while [c$_2$mim][SCN] has the lowest. It implies that the CO$_2$ solubility is affected by the number of cyanide anions contained in ionic liquid. From this result, it is concluded that the cyanide anion enhances the CO$_2$ solubility in ionic liquid and that the ionic liquid which contains more cyanide anion has higher CO$_2$ solubility (Kim et al., 2014).

Monte Carlo simulations are used to calculate the solubility of natural gas components in ionic liquids (ILs) and Selexol, which is a mixture of poly (ethylene glycol) dimethyl ethers. The solubility of the pure gases carbon dioxide (CO$_2$), methane (CH$_4$), ethane (C$_2$H$_6$), and sulfur dioxide (SO$_2$) in the following ILs: 1-alkyl-3-methylimidazolium bis(trifluoromethylsulfonyl) imide ([Cnmim] [Tf$_2$N], n = 4, 6), 1-
ethyl-3-methyl imidazolium diethylphosphate ([emim][dep]) have been computed at 313.15 K and at several pressures. The gas solubility trend observed in the experiments and simulations is: SO$_2$ > CO$_2$ > C$_2$H$_6$ > CH$_4$. Overall, the Monte Carlo simulation results are in quantitative agreement with the measured experimental data. Molecular simulation is an excellent tool to predict gas solubility in solvents and may be used as a screening tool to navigate through the large number of theoretically possible ILs (Ramdin et al., 2014).

High pressure solubility measurement of carbon dioxide in three phosphonium-based ionic liquids and modeling using the NRTL model was performed. A synthetic method was used to measure vapor–liquid, vapor–liquid–liquid and liquid–liquid equilibria of carbon dioxide in the ionic liquids trihexyl-tetradecylphosphonium bromide [thtdp][Br], trihexyl tetra decyl phosphonium dicyanamide [thtdp][dca] and trihexyltetradecyl phosphonium bis (2,4,4-trime-thylpentyl)phosphinate [thtdp][phos] at a temperature range from 271 to 363 K and up to 90 MPa. Furthermore, the densities and viscosities of these ILs have been measured in a temperature range of 293 to 363 K. The solubility of carbon dioxide in these ILs (on mole fraction basis) is significantly larger than most of the commonly used fluorinated imidazolium ionic liquids. The latter statement, however, does not hold for the [Br] and [dca] based IL if the solubility is compared on molality (mole/mass solvent) basis, where the solubility differences among physical ILs tend to vanish indicating a strong molecular weight effect. The solubility of carbon dioxide in [thtdp][phos], both on mole fraction and molality basis, is among the highest compared to all other physical ILs reported so far in the literature. The Peng–Robinson equation of state in combination with Wong–Sandler mixing rules incorporating the NRTL Gibbs
excess energy model was applied to represent the experimental data with acceptable accuracy (Ramdin et al., 2013)

Yim and Lim (2013) measured the solubility of carbon dioxide in five following ionic: 1-hexyl-3-methylimidazolium bis (trifluoromethylsulfonyl)imide ([HMIM][Tf₂N]), 1-hexyl-3-methylimidazolium trifluoromethanesulfonate ([HMIM][TfO]), 1-hexyl-3-methyl imidazolium methylsulfate ([HMIM][MeSO₄]), 1-hexyl-3-methylimidazolium tetrafluoroborate ([HMIM][BF₄]) and 1-hexyl-3-methyl imidazolium hexa-fluoro phosphate ([HMIM][PF₆]). The experimental measurements ranged from 303.15 to 373.15 K in 10 K intervals. The solubility of carbon dioxide (CO₂) was determined by measuring the bubble point pressure at a fixed temperature. Experimental data were correlated by using the PR-EoS and Modified-Joback–Reid method. The solubility for all the ionic liquids increased with higher pressure conditions and decreased with higher temperatures. The bubble point pressure increased linearly with increasing temperature at a fixed mole fraction of CO₂. In addition, we investigated the anion exerting influence of the fluorine content on the solubility of carbon dioxide in 5 different ionic liquids in this study. As a result, the solubility of CO₂ in [HMIM][Tf₂N] was found to be the highest, while the solubility of CO₂ in [HMIM][MeSO₄] proved to be the lowest. Furthermore, they also compared the solubility of CO₂ in [HMP]-cation based ionic liquids and examined the solubility data of CO₂ with respect to [Cnmpy][Tf₂N]-based ionic liquid systems. [HMIM] cation-based ionic liquid systems and the [BMP] cation-based ionic liquid systems were compared. As a result, it is concluded that the solubility depended much more on the anion combination than on the cation combination for the systems studied.
2.2 Soft computing modeling

Elimination of carbon dioxide from gas mixtures is a common commercial step in natural gas refineries. Nowadays, room-temperature ionic liquids, which are a relatively novel type of compounds have gained attention in recent years and have potential to be considered as a substitution for conventional volatile organic solvents. With no issues with flammability, thermal stability, or corrosion, and with a low vapor pressure are some of the properties of ILs, which make them interesting of interest to a growing number of researchers.

Information about the capacity and the rate of absorption is a crucial factor for the consideration of ILs as potential solvents in industrial processes. Because of some difficulties associated with experimental measurements and expenses spent on ILs, developing predictive methods, for the prediction of the phase behavior of such types of systems, is an active area of research. Thermodynamic models are relatively complex and, and are in general valid for a specific system.

Due to such difficulties there is a need to develop general models capable of predicting the phase behavior of systems such as CO₂ with various kinds of ILs. Four different methods based on artificial intelligence using radial basis function have been proposed to predict CO₂ solubility in different ionic liquids. The results showed that the predicted values are in good agreement with the experimental data and the maximum absolute error deviation for the best predictor was no more than 3.5%. A comparison between the developed models and previously published ones reveals the superiority of the proposed models in this study (Tatar et al., 2016)
Sedghamiz et al. (2014) investigated the solubility of carbon dioxide and hydrogen sulfide, in different ionic liquids (ILs) by applying the artificial neural networks (ANNs). According to the economic benefits of CO$_2$ as an inexpensive and non-toxic source of carbon, many studies dealt with capturing of CO$_2$ from different streams using ILs due to specific properties such as a negligible vapor pressure. Solubility is a key parameter in the phase equilibria calculations. According to the complexity of ILs structure, the phase behavior modeling for these systems is complicated. ANNs are nonlinear mathematical models which relate the inputs to the outputs. A total of 2930 and 664 solubility data for CO$_2$ and H$_2$S are used. The network was trained, validated and tested using 70, 15 and 15 percent of total data with one hidden layer through hyperbolic tangent sigmoid transfer function. Optimum neurons were 23 and 14 for CO$_2$ and H$_2$S solubility, respectively. AAD% and $R^2$ were 3.58 percent and 0.9947 for CO$_2$ and 2.07 and 0.9987 for H$_2$S system. In addition, the Peng–Robinson EoS with and without optimized $k_{ij}$ and an empirical correlation with different constants were used to compare their deviations with the ANN model. Results showed that the ANN model can correlate the solubility of acid gases in ILs with a high accuracy and its error was the lowest among the three approaches used.

Ionic liquids (ILs) are promising solvents for industrial applications including gas sweetening processes. For a safe and economical design, the prediction of carbon dioxide solubility by a trustworthy model is essential. Baghban (2015), using the pressure and temperature of system and the critical properties such as critical temperature ($T_c$) and critical pressure ($P_c$) and also acentric factor ($\omega$) and molecular weight (Mw) of pure ionic liquids, developed a multi-layer perceptron artificial neural network (MLP-ANN)
utilizing log-sigmoid transfer function and an adaptive neuro-fuzzy interference system (ANFIS) to estimate the carbon dioxide solubility in presence of various ILs over a wide range of pressure, temperature and concentrations. To this end, 728 experimental data points, collected from the literature, were used for implementation of these models. To verify the proposed models, a regression analysis has been conducted on the experimental and predicted solubility of carbon dioxide in ILs. Moreover, a comparison between experimental carbon dioxide solubility and predicted values of carbon dioxide solubility by thermodynamic models based on Peng–Robinson (PR) and Soave–Redlich–Kwong (SRK) equation of states was performed by Baghban (2015). For MLP-ANN, coefficient of determination ($R^2$) between experimental and predicted values was 0.9972 and the mean squared errors (MSEs) was 0.000133 and the values of $R^2 = 0.9336$ and MSE = 0.002942 were obtained for ANFIS model while, the values of $R^2$ and MSEs for PR-EOS were 0.7323 and 0.002702, respectively, and values of $R^2 = 0.6989$ and MSE = 0.005578 were obtained for the SRK-EOS model. The ability and better performance of the MLP-ANN model as an accurate correlation for estimating carbon dioxide solubility in ILs was demonstrated against other alternative models.

Furthermore, Artificial Neural Network procedure was used by Eslamimanesh et al. (2011) to represent the solubility of supercritical CO$_2$ in 24 ionic liquids. An optimized Three-Layer Feed Forward Neural Network and Sigmoid (Hyperbolic) Transfer Function using critical properties of ionic liquids and operational temperature and pressure has been developed. The model was applied for 1128 data points of 24 ionic liquids and results led to squared correlation coefficients of 0.993, and an average absolute deviation of 3.6%.
The potential use of a model based on an artificial neural network (ANN) was also investigated to predict the solubility of acid gases (H$_2$S and CO$_2$) in 32 single and mixed amines and ionic liquid (IL) solutions over a wide range of operating conditions. The temperature, partial pressure of acid gas (H$_2$S or CO$_2$), overall mass concentration, apparent molecular weight, critical temperature and critical pressure of solution were chosen as input variables of the proposed network. A collection of 733 experimental data points for H$_2$S solubility (including training, testing and validation data points) have been gathered from the literature to develop the network. The best parameters of the developed ANN containing the number of neurons, numbers of hidden layer and transfer function were acquired by using these data points. To evaluate the network accuracy, regression analysis with a data set including 169 data points for H$_2$S solubility which were not considered in the training, testing and validation stages was applied. Furthermore, the extrapolation capability of the network was investigated by an extra data set (114 data points for CO$_2$ solubility). The optimized network was trained by the Levenberg–Marquardt back-propagation algorithm with two hidden layers including 8 and 4 neurons and Tan-sigmoid transfer function for the hidden and output layers. The model results show that developed ANN model has ability to estimate accurately the solubility of acid gases in different solutions with Mean Relative Error (MRE) value of 3.104 and correlation coefficient ($R^2$) of 0.997 (Hamzehie et al., 2015).

Prediction of the solubility of CO$_2$ in ethanol–[EMIM][Tf$_2$N] ionic liquid mixtures was performed by Mirarab et al. (2014) using soft computing modeling. The solubility equilibrium data of CO$_2$ were measured at 0, 20, 50 and 80 and 100 wt. % mixture of [EMIM][Tf$_2$N] ionic liquid, at 313.2 and 333.2 K, and in a pressure range of 0 to 7 MPa.
A feed-forward multilayer perceptron (MLP) neural network with Levenberg-Marquardt learning algorithm was developed for prediction task. The ANN model was trained, using 70% of all solubility data, validated and tested with 15% and 15%, respectively. An optimization procedure was performed based on genetic algorithm to select the best ANN architecture. Therefore, a three layer feed-forward neural network with Levenberg-Marquardt back-propagation training algorithm was developed and designed with four variables as inputs and one variable as output, 15 neurons in the hidden layer, log-sigmoid transfer function in the hidden layer and linear transfer function in the output layer. Based upon the statistical analysis, results obtained demonstrated that there is a very little difference between the predicted and experimental data, giving very low value of average absolute deviation (AAD) and a high value of least square ($R^2$) that was very close to one, indicating high accuracy of this model to predict output variable. The results also proved that the developed ANN model outperformed the Peng-Robinson model.

Prediction of acid gases (carbon dioxide and hydrogen sulfide) loading capacities by employing artificial neural network (ANN) model in 51 single and blended alkanolamines, ionic liquids, and amino acid salt solutions, was implemented. In order to evaluate the extrapolation capability of the ANN, new experimental data on CO$_2$ solubility in aqueous solutions of potassium glycinate blended with piperazine (PZ) and 2-amino-2-methyl-1-propanol (AMP), at different temperatures and pressures, were measured. For developing the ANN, solution pH, total mass concentration, partial pressure of CO$_2$ and H$_2$S, apparent molecular weight, critical temperature, critical pressure and temperature were assumed as the inputs. A set of 2982 experimental data points for CO$_2$ and H$_2$S loading capacities have been collected from the literature to
create the suggested ANN. The best structure of the suggested network was achieved by employing these literature data points. The network was trained by using the Levenberg–Marquardt back-propagation, and consists of 9 and 6 neurons in first and second hidden layers, respectively. For the hidden and output layers, a Tan-sigmoid transfer function was also utilized. The output results of developed network show that suggested network that is created with solubility data of a single amine, blended alkanolamines, ionic liquids and amino acid salt solutions has the capability to predict accurately CO₂ and H₂S loadings with an Average Relative Deviation (ARD %) equal to 2.82, a Mean Square Error (MSE) value of 3.7468 × 10⁻⁵ and correlation coefficient (R²) equals to 0.9984 (Hamzehie & Najibi, 2016).

The equilibrium solubility data for CO₂ in aqueous solution of AMP have been determined in a temperature range from 293 K to 323 K, partial pressures from 17.47 kPa to 69.87 kPa and at concentrations of AMP from 1 M to 4 M. The experimental results show that the solubility of CO₂ in AMP increases with partial pressure and decreases with temperature and concentration of solvent. Two different models were used to analyze the solubility of CO₂ in AMP including the Deshmukh–Mather thermodynamic model and the artificial neural network. The modeling results indicate that the neural network modeling provides a better prediction of experimental CO₂ loadings than the Deshmukh–Mather model when compared with experimental results. Therefore, this new modeling method can be useful in predicting the results of CO₂ absorption and its accuracy is comparable with some widely used thermodynamic models (Pahlavanzadeh et al., 2011).

Finally, multiple Linear Regression (MLR), Multiple Quadratic Regression (MQR), Radial Basis Network (RB), and Multilayer Perceptron (MLP) Neural Network (NN)
models were explored by Torrecilla et al. (2008) for the prediction of the CO$_2$ solubility in 1-n-ethyl-3-methy limidazolium hexafluoro phosphate, 1-n-hexyl-3-methyl imidazolium hexafluoro phosphate, 1-n-butyl-3-methyl imidazolium tetrafluoro-borate, 1-n-hexyl-3-methylimidazolium tetrafluoroborate and 1-n-octyl-3-methyl imidazolium tetrafluoro-borate ionic liquids (ILs) at sub and supercritical conditions. The model fitting performance was analyzed by calculating statistical parameters such as the adjusted correlation coefficient, mean prediction error (MPE) and the estimated standard deviation. To verify how well the models fit the data, the analysis of residuals from the interpolative models by graphical and numerical methods and central tendency and statistical dispersion test were performed. For every model tested, the MPE values in all five studied systems were less than 12, 8, 2.2, and 2.1% for MLR, MQR, MLP and RB models, respectively. Taking the numerical analysis of residuals from the non-linear models into account, there is no correlation between residuals and CO$_2$ solubility. The MPE values were calculated by the non-linear models were less than 3.3%.

Phase equilibria data are required to estimate the capacity of a geological formation to sequester CO$_2$. Sadeghi et al. (2015) performed both thermodynamic and neural network modeling, on CO$_2$ solubility in brine. Brine is approximated by a NaCl solution. The Redlich–Kwong equation of state and Pitzer expansion are used to develop the thermodynamic model. The EoS constants were adjusted using a genetic algorithm optimization. A novel approach based on a neural network model is utilized as well. The temperature range in which the presented model is valid is 283 to 383 K, and for pressure between 0 and 600 bar, covering the temperature and pressure conditions for geological sequestration. A two-layer network consisting of 5 neurons in its hidden layer, was
chosen as the optimum topology. The regression coefficient for the neural network model was calculated giving $R^2 = 0.975$. In addition, the neural network model showed lower mean absolute percentage error (3.41%) compared to the thermodynamic model (3.55%).

As all other studies in the soft computing modeling field, Safamirzaei and Modarress (2012) proposed a modeling method based on neural network technique and molecular properties to model the solubility of carbon dioxide, carbon monoxide, argon, oxygen, nitrogen, methane and ethane in 1-butyl-3-methylimidazolium tetrafluoroborate. The molecular weight and acentric factor (i.e. sphericity of molecule) are two network inputs which represent the structure of the gas molecule. Absolute temperature and pressure are two other inputs which exhibit macroscopic condition of studied system. Low deviations during the training, validating and testing stages confirmed that the model is reliable within the studied range. Also, the proposed method is able to provide reliable gas solubility estimations based on available solubility data of other gases. This unique capability, which confirms the superiority of applied method over the traditional methods, enables researchers and engineers to provide acceptable gas solubility estimations without performing long time experiments.

Finally, Mirarab et al. (2014) proposed an artificial neural network (ANN) technique as a new approach to predict the solubility of CO$_2$ in ethanol–[EMIM][Tf$_2$N] ionic liquid mixtures. The solubility equilibrium data of CO$_2$ were measured at 0, 20, 50 and 80 and 100 wt. % mixture of [EMIM][Tf$_2$N] ionic liquid, temperatures of 313.2 and 333.2 K, and pressure range of 0 to 7 MPa. A feed-forward multilayer perceptron (MLP) neural network with Levenberg-Marquardt learning algorithm was developed for prediction task. The ANN model was trained, validated, and tested using 70% of all solubility.
optimization procedure was performed based on genetic algorithm to select the best ANN architecture. Therefore, a three layer feed-forward neural network with Levenberg-Marquardt back-propagation training algorithm was developed and designed with four variables as inputs and one variable as output, 15 neurons in the hidden layer, log-sigmoid transfer function in the hidden layer and linear transfer function in the output layer. Based upon statistical analysis, results obtained demonstrated that there is a very little difference between predicted and experimental data of CO₂ capture rate giving very low value of average absolute deviation (AAD) and high value of least square (R²) very close to one, demonstrating a high accuracy of this model to predict the output variables. The results also proved that the developed ANN model outperformed the Peng-Robinson model.
Chapter 3: Methodology

In this chapter, the data sets for both carbon dioxide and ethane from different sources are provided. In the rest of the chapter, artificial neural network and adaptive neuro fuzzy systems which have been employed in this study are explained in detail.

3.1 Data Collection

Collecting a wide range of real and newly obtained data from various recent experiments was a crucial step to create an appropriate network. A set of 514 data for 15 different ionic liquids is collected for the solubility of CO₂ and 90 data points for three ILs were collected for the solubility of ethane, all in different ranges of temperature and pressure which the following tables representing all related information for each selected ILs for carbon dioxide and ethane.

Table 3-1. Ranges of temperature, pressure and CO₂ solubility while using certain ILs

<table>
<thead>
<tr>
<th>No.</th>
<th>Ionic Liquid</th>
<th>Temp range (K)</th>
<th>Pressure range (MPa)</th>
<th>CO₂ solubility range (mole fraction)</th>
<th>No. of data</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[EMIM][LACTATE]</td>
<td>313.15-333.15</td>
<td>0.05-1.89</td>
<td>0.043-0.283</td>
<td>24</td>
<td>(Zoubeik, et al., 2016)</td>
</tr>
<tr>
<td>2</td>
<td>[PMPY] [TF2N]</td>
<td>313.15-333.15</td>
<td>0.0096-1.90</td>
<td>0.00219-0.317</td>
<td>39</td>
<td>(Zoubeik, et al., 2016)</td>
</tr>
<tr>
<td>3</td>
<td>[(CH2)4SO3HMIm][TF2N]</td>
<td>313.15-323.15</td>
<td>0.0098-1.9</td>
<td>0.00132-0.254</td>
<td>18</td>
<td>(Zoubeik, et al., 2016)</td>
</tr>
<tr>
<td>4</td>
<td>[(CH2)4SO3HMIm][HSO4]</td>
<td>313.15-323.15</td>
<td>0.0099-1.70</td>
<td>0.00056-0.0614</td>
<td>14</td>
<td>(Zoubeik, et al., 2016)</td>
</tr>
<tr>
<td>No.</td>
<td>Ionic Liquid</td>
<td>Temp range (K)</td>
<td>Pressure range (MPa)</td>
<td>CO2 solubility range (mole fraction)</td>
<td>No. of data</td>
<td>Reference</td>
</tr>
<tr>
<td>-----</td>
<td>-------------------</td>
<td>----------------</td>
<td>----------------------</td>
<td>--------------------------------------</td>
<td>-------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>[TDC] [DCN]</td>
<td>313.15-333.15</td>
<td>0.01-1.90</td>
<td>0.00175-0.272</td>
<td>38</td>
<td>(Zoubeik &amp; Henni, 2014)</td>
</tr>
<tr>
<td>6</td>
<td>[EMMP] [TF2N]</td>
<td>313.15-333.15</td>
<td>0.0098-1.9</td>
<td>0.0018-0.3165</td>
<td>39</td>
<td>(Zoubeik &amp; Henni, 2014)</td>
</tr>
<tr>
<td>7</td>
<td>[TDC] [TF2N]</td>
<td>313.15-333.15</td>
<td>0.0097-1.89</td>
<td>0.0023-0.36</td>
<td>39</td>
<td>(Zoubeik &amp; Henni, 2014)</td>
</tr>
<tr>
<td>8</td>
<td>[p(5)mppyrr][TF2N]</td>
<td>298.15-343.15</td>
<td>0.0097-1.90</td>
<td>0.002-0.406</td>
<td>36</td>
<td>(Tagiuri, et al., 2014)</td>
</tr>
<tr>
<td>9</td>
<td>[bmmim][TF2N]</td>
<td>298.15-343.15</td>
<td>0.0099-1.89</td>
<td>0.002-0.382</td>
<td>36</td>
<td>(Tagiuri, et al., 2014)</td>
</tr>
<tr>
<td>10</td>
<td>[P4441][TF2N]</td>
<td>298.15-343.15</td>
<td>0.0099-1.89</td>
<td>0.003-0.393</td>
<td>36</td>
<td>(Tagiuri, et al., 2014)</td>
</tr>
<tr>
<td>11</td>
<td>[S222][TF2N]</td>
<td>313.15-333.15</td>
<td>0.0095-1.9006</td>
<td>0.002-0.3033</td>
<td>39</td>
<td>(Nonthanas in, et al., 2014)</td>
</tr>
<tr>
<td>12</td>
<td>[deme][TF2N]</td>
<td>313.15-333.15</td>
<td>0.00985-1.90044</td>
<td>0.0023-0.3202</td>
<td>39</td>
<td>(Nonthanas in, et al., 2014)</td>
</tr>
<tr>
<td>13</td>
<td>[pmim][TF2N]</td>
<td>313.15-333.15</td>
<td>0.0099-1.9003</td>
<td>0.0023-0.316</td>
<td>39</td>
<td>(Nonthanas in, et al., 2014)</td>
</tr>
<tr>
<td>14</td>
<td>[amim][TF2N]</td>
<td>313.15-333.15</td>
<td>0.01-1.8998</td>
<td>0.0021-0.315</td>
<td>39</td>
<td>(Nonthanas in, et al., 2014)</td>
</tr>
<tr>
<td>15</td>
<td>[4mbp][BF4]</td>
<td>313.15-333.15</td>
<td>0.0095-1.9001</td>
<td>0.001-0.2028</td>
<td>39</td>
<td>(Nonthanas in, et al., 2014)</td>
</tr>
</tbody>
</table>
Table 3-2. Ranges of temperature, pressure and C₂H₆ solubility while using certain ILs

<table>
<thead>
<tr>
<th>No.</th>
<th>Ionic Liquid</th>
<th>Temp range (K)</th>
<th>Pressure range (MPa)</th>
<th>C₂H₆ solubility range (mole fraction)</th>
<th>No. of data</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[mp(3)pip][FSI]</td>
<td>303.15-343.15</td>
<td>0.0002-1.0124</td>
<td>0-0.0485</td>
<td>27</td>
<td>(Nath et al., 2016)</td>
</tr>
<tr>
<td>2</td>
<td>[mp(3)pyrr][FSI]</td>
<td>303.15-343.15</td>
<td>0.0001-1.0176</td>
<td>0-0.454</td>
<td>27</td>
<td>(Nath et al., 2016)</td>
</tr>
<tr>
<td>3</td>
<td>[N1223][FSI]</td>
<td>303.15-343.15</td>
<td>0.0002-1.0177</td>
<td>0-0.0474</td>
<td>36</td>
<td>(Nath et al., 2016)</td>
</tr>
</tbody>
</table>

The critical properties including critical temperature (T_c), critical pressure (P_c), and the acentric factor (ω) are the information for the ionic liquids which is presented in the following table 3.3.

Table 3-3. Critical properties and acentric factors of ILs used in this study

<table>
<thead>
<tr>
<th>Compound</th>
<th>Tc (K)</th>
<th>Pc (MPa)</th>
<th>ω</th>
</tr>
</thead>
<tbody>
<tr>
<td>[EMIM] [LACTATE]</td>
<td>912.7</td>
<td>2.824</td>
<td>0.9702</td>
</tr>
<tr>
<td>[PMPY] [TF2N]</td>
<td>1234.2</td>
<td>2.755</td>
<td>0.3070</td>
</tr>
<tr>
<td>[(CH2)4SO3HMIm] [TF2N]</td>
<td>1612.8</td>
<td>3.27</td>
<td>0.377</td>
</tr>
<tr>
<td>[(CH2)4SO3HMIm] [HSO4]</td>
<td>1433.0</td>
<td>2.588</td>
<td>0.8437</td>
</tr>
<tr>
<td>[TDC] [DCN]</td>
<td>1073.7</td>
<td>1.615</td>
<td>1.0726</td>
</tr>
<tr>
<td>[EMMP] [TF2N]</td>
<td>1038.7</td>
<td>2.588</td>
<td>0.3334</td>
</tr>
<tr>
<td>[TDC] [TF2N]</td>
<td>1255.7</td>
<td>1.803</td>
<td>0.5876</td>
</tr>
<tr>
<td>[p(5)mpyr] [TF2N]</td>
<td>1221.9</td>
<td>1.828</td>
<td>0.2603</td>
</tr>
<tr>
<td>[bmmim][TF2N]</td>
<td>1255.8</td>
<td>2.031</td>
<td>0.3139</td>
</tr>
<tr>
<td>[P4441][TF2N]</td>
<td>1155</td>
<td>1.173</td>
<td>0.5207</td>
</tr>
<tr>
<td>[S222][TF2N]</td>
<td>1191.1</td>
<td>2.19</td>
<td>0.1561</td>
</tr>
<tr>
<td>[deme][TF2N]</td>
<td>1080.5</td>
<td>2.35</td>
<td>0.3915</td>
</tr>
</tbody>
</table>
### 3.2 Soft Computing Models

Soft computing is an automated intelligent estimation to solve complex problems by accepting good enough results, which may not be solvable by hard computing. Soft computing provides approximate solutions with low cost of calculation and is faster compared to hard models. Soft computing can solve complex problems approximately if exact solution is not available or too complex to analytically determine. During the last three decades different soft modeling methods have been developed which are the integration of biological structures and computing techniques such as artificial neural networks (ANN), genetic algorithms, fuzzy logic, and fusion methods (Li et al., 1998).

#### 3.2.1 Artificial Neural Networks

Artificial neural networks (ANN) system updates its structure based on external and internal information that feed to the system network during the learning phase. This mimics the working of the human brain neurons network which learns from experience without any previous knowledge or information on the system (Jain & Martin, 1998; Nagy & Watanabe, 2002).
3.2.2 The Feedforward Backpropagation Neural Network Algorithm

Although the long-term goal of the neural-network community remains the design of autonomous machine intelligence, the main modern application of artificial neural networks is in the field of pattern recognition (Joshi et al., 1997). In the sub-field of data classification, neural-network methods have been found to be useful alternatives to statistical techniques such as those which involve regression analysis or probability density estimation (Holmström et al., 1997). The potential utility of neural networks in the classification of multisource satellite-imagery databases has been recognized for well over a decade, and today neural networks are an established tool in the field of remote sensing.

3.2.3 Fuzzy Logic

Fuzzy logic models uses approximate reasoning (IF-THEN) rules, which works like human decision making “rule of thumb.” The final product is a graded statement rather than strict results such as true or false (Jain & Martin, 1998 and Sen, 2010).

3.3 Artificial Intelligence (AI)

Artificial Intelligence is a universal field which encompasses a huge variety of subfields, from more general areas such as learning and perception to more specific tasks such as playing chess, writing poetry, and diagnosing diseases (Russell & Norvig, 2003).

McCarthy was the first one to introduce AI as a new field in 1955. Afterwards McCarthy and some other researchers resumed their research in this field in spite of all the difficulties they encountered. Three decades later, AI became a science and a revolution that occurred both in its content and methodology (Russell & Norvig, 2003).
There are several definitions for AI according to different researchers; however, it can generally be defined as “the study and design of intelligent agents.” An intelligent agent is something that acts intelligently in an environment according to the circumstances and its goal. This agent is flexible to the change of environments and goals. It learns from experience, and, within a certain limitation, it is able to make appropriate choices (Poole et al., 1998).

Artificial Neural Network (ANN), Fuzzy Logic (FL), and Genetic Algorithms (GA) are among the most popular artificial intelligence techniques that are used to solve engineering problems. Incorporation of several virtual or artificial intelligence tools in a hybrid manner can generate a successful intelligent application. Virtual intelligence tools are able to complement each other by amplifying each other’s effectiveness (Mohaghegh, 2011).

As one of the sub-categories of the artificial intelligence method, Artificial Neural Networks (ANNs) have a great potential for performing accurate analysis and predictions from historical data. The ANNs are particularly useful where mathematical modeling cannot be considered as a practical option. This can be due to the fact that all the parameters involved in the process are not known and/or there is a very complex relationship between the parameters of the system, which is too complicated to be presented through mathematical formulations (Holdaway, 2014).

The artificial neural network is a powerful tool that has been successfully used in a wide variety of problems in different areas such as medical, engineering, financial, business, etc. However, in different industries, many cases of unsuccessful neural
network implementations have been reported, which can be traced back to inappropriate neural network design and general misconceptions about how they work (Reid, 2014).

In the current research, the artificial neural network method is employed to develop a prediction function. This technique is explained in detail throughout the upcoming sections.

3.3.1 Artificial Neural Network Overview

Neural networks are members of a family of computational architectures inspired by biological brains (McClelland et al., 1986; Luger and Stubblefield, 1993). Such architectures are commonly called "connectionist systems", and are composed of interconnected and interacting components called nodes or neurons. Neural networks are characterized by a lack of explicit representation of knowledge; there are no symbols or values that directly correspond to classes of interest. Rather, knowledge is implicitly represented in the patterns of interactions between network components (Lugar & Stubblefield, 1993). A graphical depiction of a typical feedforward neural network is given in Figure 3.1. The term “feedforward” indicates that the network has links that extend in only one direction. Except during training, there are no backward links in a feedforward network, in other words, all links proceed from input nodes toward output nodes.
Figure 3-1. A typical feedforward neural network.

The method used in this study is the feed-forward neural network which will be explained in details in the following section.

Individual nodes in a neural network emulate biological neurons by taking input data and performing simple operations on the data, selectively passing the results on to other neurons. The output of each node is called "activation". Weight values are associated with each vector and node in the network, and these values constrain how input data are related to output data. Weight values associated with individual nodes are also known as biases. Weight values are determined by the iterative flow of training data through the network (weight values are established during a training phase in which the network learns how to identify particular classes by their typical input data characteristics) (Leverington, 2009).
3.3.2 Feed-forward Neural Network Model

The feed-forward neural network is known as the most widely used types of the artificial neural network in which the information flows in one direction from input toward output. The simplest feed-forward artificial neural network consists of a single perceptron that is only capable of learning linear separable problems. A simple multilayer feed-forward artificial neural network is used to solve more complex problems.

The basic neural network model can be described as a series of functional transformations. Neural networks use basis functions in the form of following equation (Bishop, 2006):

\[ y(x, w) = f \left( \sum_{j=1}^{M} \omega_j \phi_j(x) \right) \]  (3.1)

Each basis function, itself, is a nonlinear function of a linear combination of the inputs.

Considering \( x_1, x_2, \ldots, x_D \) as inputs, in order to develop the neural network, \( M \) linear combinations of the input variables are constructed through the following formulation:

\[ a_j = \sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)} \]  (3.2)

where, \( a_j \) is known as activation with \( j = 1, 2, \ldots, M \), and subscript (1) is the representative of the first layer of the network. The parameters \( \omega_{ji}^{(1)} \) are referred to as weights, and \( \omega_{j0}^{(1)} \) are called the biases.

Each activation is transformed using a nonlinear activation function \( h(a_j) \) which then gives:

\[ z_j = h(a_j) \]  (3.3)
These quantities correspond to the hidden units. The nonlinear function of $h(.)$ can be a sigmoidal function or the $\text{'tanh'}$ function. In order to obtain the output unit activation, these values are again linearly combined.

$$a_k = \sum_{j=1}^{M} \omega_{kj}^{(2)} z_j + \omega_{k0}^{(2)} \quad (3.4)$$

Equation 3-4 is the transformation related to the second layer of the network in which, $k = 1, 2, \ldots, K$ and $K$ is the number of outputs. The same as Equation 3.2, the $\omega_{j0}^{(1)}$'s are the bias parameters.

Finally, in order to calculate the network output, $y_k$ the output unit activations are transformed through an appropriate activation function. Several activation functions are available for different systems. The linear, unit step, ramp, logistic sigmoid, hyperbolic tangent, and Gaussian transfer function are the most commonly used. The diagram below shows the schematics of the six activation functions (Reid, 2014).
Figure 3-2. Six most popular activation functions used in neural network models (Reid, 2014)

The choice of activation function is determined based on the nature of the data and also the type of problem (Bishop, 2006). The logistic sigmoid function, $\delta$ which is the activation function used in this work, is defined as:

$$\delta(a) = \frac{1}{1+\exp(-a)}$$

(3.5)

Considering the sigmoid function as the activation function, each output unit is transformed through $y_k = (a_k)$ formula.
By combining the various stages mentioned above (Equation 3.2 to Equation 3.6),
the overall neural network function can be represented in the following form:

\[
y_k(x, w) = \delta\left(\sum_{j=1}^{M} \omega_{kj}^{(2)} h\left(\sum_{i=1}^{D} \omega_{ji}^{(1)} x_i + \omega_{j0}^{(1)}\right) + \omega_{k0}^{(2)}\right)
\] (3.6)

As indicated in Equation 3-6 the neural network model is a nonlinear function of a
set of input \(\{x_i\}\) to a set of output \(\{y_i\}\), and it is controlled by a set of adjustable
parameters which are the weights.

### 3.3.3 Neural Network Training

Neural network training will be performed once the network structure is determined. In
general, neural network is considered as a class of non-linear parametric function. The
neural network training process comprises several steps that lead to the determination of
the involved network parameters. A simple analogy to this process is curve fitting
through which a sum of squares of the error function is minimized. Therefore, in a neural
network, given a set of input vectors \(\{x_n\}, n = 1, 2, \ldots, N\), and the corresponding set of
target vectors \(\{t_n\}\), the goal is to minimize the error function which can be denoted as
follows:

\[
E(w) = \frac{1}{2} \sum_{n=1}^{N} ||y(x_n, w) - t_n||^2
\] (3.7)

The next step is to find the weight vector such that the error function is minimized.
In most techniques, an initial value is chosen for the weight vector \((0)\), and in subsequent
steps the weights are updated through:

\[
w^{\tau+1} = w^{\tau} + \Delta w^{\tau}
\] (3.8)

Where, \(\tau\) is an indicator for the iteration step. Different algorithms use different
choices for updating the weight vector. However, many of them apply gradient
information, and therefore the value of $\nabla(w)$ should be evaluated at the new iteration step (Bishop, 2006).

The method is utilized to train the neural network in this study was back propagation technique which will be discussed in the following section.

**Back Propagation Technique**

In the employment of the backpropagation algorithm, each iteration of training involves the following steps: 1) a particular case of training data is fed through the network in a forward direction, producing results at the output layer, 2) error is calculated at the output nodes based on known target information, and the necessary changes to the weights that lead into the output layer are determined based upon this error calculation, 3) the changes to the weights that lead to the preceding network layers are determined as a function of the properties of the neurons to which they directly connect (weight changes are calculated, layer by layer, as a function of the errors determined for all subsequent layers, working backward toward the input layer) until all necessary weight changes are calculated for the entire network. The calculated weight changes are then implemented throughout the network, the next iteration begins, and the entire procedure is repeated using the next training pattern. In the case of a neural network with hidden layers, the backpropagation algorithm is given by the following equations (Gallant, 1993).

Assuming a particular input pattern $\mathbf{n}$, the error function takes the following form.

$$E_n = \frac{1}{2} \sum_k (y_{nk} - t_{nk})^2 \quad (3.9)$$

Where, $y_{nk} = y_k(x_n, w)$. The gradient of this error function is calculated with respect to a weight $w_{ij}$ through following Equation.
\[
\frac{\partial E_n}{\partial w_{ij}} = (y_{nj} - t_{nj})x_{ni} \quad (3.10)
\]

In general feed-forward network, each unit calculates the weighted some of its inputs as denoted below:

\[
a_j = \sum_i w_{ij}z_i \quad (3.11)
\]

where, \(z_i\) represents the activation of a unit, or an input that has a connection to unit \(j\), and \(w_{ij}\) is the weight associated with that connection. The summation in Equation 3.11 is transformed through a nonlinear activation function to generate the activation \(z_j\) of unit \(j\).

\[
z_j = h(a_j) \quad (3.12)
\]

The forward propagation is the process through which the input vectors are supplied to the network, and the activations of the hidden and output units are calculated by successive application of Equations 3.11 and 3.12.

Applying the chain rule to the gradient of the error function, the partial derivative can be rewritten as:

\[
\frac{\partial E_n}{\partial w_{ij}} = \frac{\partial E_n}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} \quad (3.13)
\]

Now, a new notation is introduced as \(\delta_j \equiv \frac{\partial E_n}{\partial a_j}\)

Using Equation 3.11, the second term can be written as \(\frac{\partial a_j}{\partial w_{ij}} = z_i\)

By substituting these two terms into Equation 3.13, the derivative can be obtained by multiplying the value of \(\delta\) for the unit at the output end of the weight by the value of \(z\) for the unit at the input end of the weight.
\[ \frac{\partial E_n}{\partial w_{ij}} = \delta_j z_i \]  

(3.14)

What is calculated for the output layer is:

\[ \delta_k = y_k - t_k \]  

(3.15)

By the use of the chain rule, the hidden layer can be calculated by:

\[ \delta_j = \sum_k \frac{\delta E_n}{\delta a_k} \frac{\delta a_k}{\delta a_j} \]  

Where, the sum applies on all of unit \( k \) to which unit \( j \) sends connections.

By the definition of the \( \delta \) and the equations 3.11, 3.12, the back propagation formula is as follows:

\[ \delta_j = h'(a_j) \sum_k w_{kj} \delta_k \]  

(3.17)

The procedure to apply the back propagation method can be summarized in the following steps (Bishop, 2006):

1. The input vector is applied to the network and it is propagated forward through the network using equation 3.11 and equation 3.12. This results in finding the activations for all the hidden and output units.
2. \( \delta_k \)’s are evaluated for all the output units using equation 3.15.
3. The \( \delta \)'s are back propagated using equation 3.12 to obtain \( \delta_j \)'s for each hidden unit in the network.
4. Equation 3.14 is applied to evaluate the required derivatives.
5. The weights are updated and the same procedure is repeated until the stopping condition is met.
3.4 Fuzzy Logic

In recent years, the number and variety of applications of fuzzy logic have increased significantly. The applications range from consumer products such as cameras, camcorders, washing machines, and microwave ovens to industrial process control, medical instrumentation, decision-support systems, and portfolio selection.

Fuzzy logic has two different meanings. In a narrow sense, fuzzy logic is a logical system, which is an extension of multivalued logic. However, in a wider sense fuzzy logic (FL) is almost synonymous with the theory of fuzzy sets, a theory which relates to classes of objects with un-sharp boundaries in which membership is a matter of degree. In this perspective, fuzzy logic in its narrow sense is a branch of FL. Even in its more narrow definition, fuzzy logic differs both in concept and substance from traditional multivalued logical systems.

In this study, fuzzy logic should be interpreted as FL, that is, fuzzy logic in its wide sense. The basic concept underlying FL is that of a linguistic variable, that is, a variable whose values are words rather than numbers. In effect, much of FL may be viewed as a methodology for computing with words rather than numbers. Although words are inherently less precise than numbers, their use is closer to human intuition. Furthermore, computing with words exploits the tolerance for imprecision and thereby lowers the cost of solution.

Another basic concept in FL, which plays a central role in most of its applications, is that of a fuzzy if-then rule or, simply, fuzzy rule. Although rule-based systems have a long history of use in Artificial Intelligence (AI), what is missing in such systems is a
mechanism for dealing with fuzzy consequents and fuzzy antecedents. In fuzzy logic, this mechanism is provided by the calculus of fuzzy rules. The calculus of fuzzy rules serves as a basis for what might be called the Fuzzy Dependency and Command Language (FDCL). Although FDCL is not used explicitly in the toolbox, it is effectively one of its principal constituents. In most of the applications of fuzzy logic, a fuzzy logic solution is, in reality, a translation of a human solution into FDCL (Sugeno, 1985).

A trend that is growing in visibility relates to the use of fuzzy logic in combination with neuro computing and genetic algorithms. More generally, fuzzy logic, neuro computing, and genetic algorithms may be viewed as the principal constituents of what might be called soft computing. Unlike the traditional, hard computing, soft computing accommodates the imprecision of the real world. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost. In the future, soft computing could play an increasingly important role in the conception and design of systems which MIQ (Machine IQ) is much higher than that of systems designed by conventional methods (Sugeno, 1985).

Among various combinations of methodologies in soft computing, the one that has highest visibility at this juncture is that of fuzzy logic and neuro computing, leading to neuro-fuzzy systems. Within fuzzy logic, such systems play a particularly important role in the induction of rules from observations. An effective method developed by Jang et al. for this purpose and is called ANFIS (Adaptive Neuro-Fuzzy Inference System) (Jang et al., 1993).
Fuzzy logic is all about the relative importance of precision: How important is it to be exactly right when a rough answer will do? Fuzzy logic is a fascinating area of research because it does a good job of trading off between significance and precision—something that humans have been managing for a very long time. In this sense, fuzzy logic is both old and new because, although the modern and methodical science of fuzzy logic is still young, the concepts of fuzzy logic rely on age-old skills of human reasoning (Zadeh, 1988).

Fuzzy logic is a convenient way to map an input space to an output space. Mapping input to output is the starting point for everything. Between the input and the output, the preceding figure shows a black box that can contain any number of things: fuzzy systems, linear systems, expert systems, neural networks, differential equations, interpolated multidimensional lookup tables, or even a spiritual advisor, just to name a few of the possible options (Zadeh, 1988). However, fuzzy is often the very best way as it is faster and cheaper.

**Why choosing fuzzy logic**

*Fuzzy logic is conceptually easy to understand:* The mathematical concepts behind fuzzy reasoning are very simple. Fuzzy logic is a more intuitive approach without the far-reaching complexity (Mamdani, 1977).

*Fuzzy logic is flexible:* With any given system, it is easy to layer on more functionality without starting again from scratch.
**Fuzzy logic is tolerant of imprecise data:** Everything is imprecise if it is looked closely enough, but more than that, most things are imprecise even on careful inspection. Fuzzy reasoning builds this understanding into the process rather than tacking it onto the end.

**Fuzzy logic can model nonlinear functions of arbitrary complexity:** Fuzzy system can be created to match any set of input-output data. This process is made particularly easy by adaptive techniques like Adaptive Neuro-Fuzzy Inference Systems (ANFIS).

**Fuzzy logic can be built on top of the experience of experts:** In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets the user rely on the experience of people who already understand the system.

**Fuzzy logic can be blended with conventional control techniques:** Fuzzy systems don't necessarily replace conventional control methods. In many cases fuzzy systems augment them and simplify their implementation.

**Fuzzy logic is based on natural language:** The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic. Because fuzzy logic is built on the structures of qualitative description used in everyday language, fuzzy logic is easy to use (Mamdani, 1977).

The last statement is perhaps the most important one and deserves more discussion. Natural language, which is used by ordinary people on a daily basis, has been shaped by thousands of years of human history to be convenient and efficient. Sentences written in ordinary language represent a triumph of efficient communication (Mamdani, 1977).
3.4.1 Fuzzy Logic Foundations

Fuzzy inference is a method that interprets the values in the input vector and, based on some set of rules, assigns values to the output vector. Fuzzy logic starts with the concept of a fuzzy set. A fuzzy set is a set without a crisp, clearly defined boundary. It can contain elements with only a partial degree of membership (Schweizer & Sklar, 1963).

Any statement can be fuzzy. The major advantage that fuzzy reasoning offers is its ability to reply to a yes-no question with a not-quite-yes-or-no answer. Humans do this kind of thing all the time (think how rarely you get a straight answer to a seemingly simple question), but it is a rather new trick for computers (Schweizer & Sklar, 1963).

Reasoning in fuzzy logic is just a matter of generalizing the familiar yes-no (Boolean) logic. If you give true the numerical value of 1 and false the numerical value of 0, this value indicates that fuzzy logic also permits in-between values like 0.2 and 0.7453 (Sugeno, 1977).

3.4.2 Membership Functions

A membership function (MF) is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership) between 0 and 1. The input space is sometimes referred to as the universe of discourse, a fancy name for a simple concept. The only condition a membership function must really satisfy is that it must vary between 0 and 1. The function itself can be an arbitrary curve whose shape we can define as a function that suits us from the point of view of simplicity, convenience, speed, and efficiency (Sugeno, 1977).

Summary of Membership Functions are as follows:
Fuzzy sets describe vague concepts (e.g., fast runner, hot weather, and weekend days).

A fuzzy set admits the possibility of partial membership in it. (e.g., Friday is sort of a weekend day, the weather is rather hot).

The degree an object belongs to a fuzzy set is denoted by a membership value between 0 and 1.

A membership function associated with a given fuzzy set maps an input value to its appropriate membership value.

The most important thing to realize about fuzzy logical reasoning is the fact that it is a superset of standard Boolean logic. In other words, if you keep the fuzzy values at their extremes of 1 (completely true), and 0 (completely false), standard logical operations will hold (Zadeh, 1975).

If-Then Rules

Fuzzy sets and fuzzy operators are the subjects and verbs of fuzzy logic. These if-then rule statements are used to formulate the conditional statements that comprise fuzzy logic.

A single fuzzy if-then rule assumes the form:

if \( x \) is \( A \) then \( y \) is \( B \)

where, \( A \) and \( B \) are linguistic values defined by fuzzy sets on the ranges (universes of discourse) \( X \) and \( Y \), respectively. The if-part of the rule "\( x \) is \( A \)" is called the
antecedent or premise, while the then-part of the rule "y is B" is called the consequent or conclusion (Zadeh, 1988).

Interpreting if-then rules is the following three-part process:

1. **Fuzzify inputs**: Resolve all fuzzy statements in the antecedent to a degree of membership between 0 and 1. If there is only one part to the antecedent, then this is the degree of support for the rule.

2. **Apply fuzzy operator to multiple part antecedents**: If there are multiple parts to the antecedent, apply fuzzy logic operators and resolve the antecedent to a single number between 0 and 1. This is the degree of support for the rule.

3. **Apply implication method**: Use the degree of support for the entire rule to shape the output fuzzy set. The consequent of a fuzzy rule assigns an entire fuzzy set to the output. This fuzzy set is represented by a membership function that is chosen to indicate the qualities of the consequent. If the antecedent is only partially true, (i.e., is assigned a value less than 1), then the output fuzzy set is truncated according to the implication method.

In general, one rule alone is not effective. Two or more rules that can play off one another are needed. The output of each rule is a fuzzy set. The output fuzzy sets for each rule are then aggregated into a single output fuzzy set. Finally the resulting set is defuzzified, or resolved to a single number.

### 3.4.3 Fuzzy Inference Process

Fuzzy inference is the process of formulating the mapping from a given input to an output using fuzzy logic. The mapping then provides a basis from which decisions can be
made, or patterns discerned. The process of fuzzy inference involves all of the pieces that are described in Membership Functions, Logical Operations, and If-Then Rules. The parallel nature of the rules is one of the more important aspects of fuzzy logic systems. Instead of sharp switching between modes based on breakpoints, logic flows smoothly from regions where the system's behavior is dominated by either one rule or another (Yager & Filev, 1994).

Fuzzy inference process comprises of five parts:

- Fuzzification of the input variables
- Application of the fuzzy operator (AND or OR) in the antecedent
- Implication from the antecedent to the consequent
- Aggregation of the consequents across the rules
- Defuzzification

A fuzzy inference diagram displays all parts of the fuzzy inference process from fuzzification through defuzzification.

Step 1. Fuzzify Inputs

The first step is to take the inputs and determine the degree to which they belong to each of the appropriate fuzzy sets via membership functions. The input is always a crisp numerical value limited to the universe of discourse of the input variable (in this case the interval between 0 and 10) and the output is a fuzzy degree of membership in the qualifying linguistic set (always the interval between 0 and 1). Fuzzification of the input amounts to either a table lookup or a function evaluation.
Step 2. Apply Fuzzy Operator

After the inputs are fuzzified, you know the degree to which each part of the antecedent is satisfied for each rule. If the antecedent of a given rule has more than one part, the fuzzy operator is applied to obtain one number that represents the result of the antecedent for that rule. This number is then applied to the output function. The input to the fuzzy operator is two or more membership values from fuzzified input variables. The output is a single truth value.

Step 3. Apply Implication Method

Before applying the implication method, the rule's weight must be determined. Every rule has a weight (a number between 0 and 1), which is applied to the number given by the antecedent. Generally, this weight is 1 and thus has no effect at all on the implication process. After proper weighting has been assigned to each rule, the implication method is implemented. A consequent is a fuzzy set represented by a membership function, which weights appropriately the linguistic characteristics that are attributed to it. The consequent is reshaped using a function associated with the antecedent (a single number). The input for the implication process is a single number given by the antecedent, and the output is a fuzzy set. Implication is implemented for each rule.

Step 4. Aggregate All Outputs

Because decisions are based on the testing of all of the rules in a FIS, the rules must be combined in some manner in order to make a decision. Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single
fuzzy set. Aggregation only occurs once for each output variable, just prior to the fifth and final step, defuzzification. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable.

Step 5. Defuzzify

The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number. As much as fuzziness helps the rule evaluation during the intermediate steps, the final desired output for each variable is generally a single number. However, the aggregate of a fuzzy set encompasses a range of output values, and so must be defuzzified in order to resolve a single output value from the set (Zadeh, 1965).

Fuzzy Inference Diagram

The fuzzy inference diagram is the composite of all the smaller diagrams presented so far in this section. It simultaneously displays all parts of the fuzzy inference process you have examined. Information flows through the fuzzy inference diagram as shown in the following figure:

![Fuzzy Inference Diagram](image)

Figure 3-3. Interpreting the fuzzy inference diagram
In this figure, the flow proceeds up from the inputs in the lower left, then across each row, or rule, and then down the rule outputs to finish in the lower right. This compact flow shows everything at once, from linguistic variable fuzzification all the way through defuzzification of the aggregate output (Sugeno, 1985).

It should be also mentioned that in this study, Sugeno-type fuzzy inference has been utilized. The main characteristic of the so-called Sugeno, or Takagi-Sugeno-Kang, method of fuzzy inference is that the Sugeno output membership functions are either linear or constant. A typical rule in a Sugeno fuzzy model has the form:

If Input 1 = x and Input 2 = y, then Output is z = ax + by + c

3.4.4 Neuro-Adaptive Learning

The shape of the membership functions in fuzzy models depends on many parameters, and changing these parameters changes the shape of the membership function. In some modeling situations, we cannot discern what the membership functions should look like simply from looking at data. Rather than choosing the parameters associated with a given membership function arbitrarily, these parameters could be chosen so as to tailor the membership functions to the input/output data in order to account for these types of variations in the data values. In such cases, you can use the neuro-adaptive learning techniques (Zadeh, 1973).

The neuro-adaptive learning method works similarly to that of neural networks. Neuro-adaptive learning techniques provide a method for the fuzzy modeling procedure to learn information about a data set. In this case, the membership function parameters
are set so that the associated fuzzy inference system is able to track the given input/output
data as much as possible. A network-type structure similar to that of a neural network,
which maps inputs through input membership functions and associated parameters and
then through output membership functions and associated parameters to outputs, can be
used to interpret the input/output map (Zadeh, 1973).

The parameters associated with the membership functions changes through the
learning process. The computation of these parameters (or their adjustment) is facilitated
by a gradient vector. This gradient vector provides a measure of how well the fuzzy
inference system is modeling the input/output data for a given set of parameters. When
the gradient vector is obtained, any of several optimization routines can be applied in
order to adjust the parameters to reduce the error. This error measure is usually defined
by the sum of the squared difference between actual and desired outputs. Either back
propagation or a combination of least squares estimation and back propagation can be
used for membership function parameter estimation (Zadeh, 1973).

As the sigmoid function introduces non-linearity in the network, it is used as
opposed to continuous and differentiable. Its derivative is very fast to compute (as
opposed to the derivative of tanh, which has similar properties) and convention/
interpretability, and has a limited range (from 0 to 1, exclusive). To this end, in this
research, the sigmoid function is used in neural networks to give logistic neurons real-
valued output that is a smooth and bounded function of their total input.

Moreover, the structure of neural-fuzzy network is formed by combination of neural
networks and fuzzy systems. Hence this structure utilizes both the trainable property of
the neural networks and the inference ability of fuzzy systems which increases the power and accuracy in uncertain situations therefore utilizing ANFIS would be the other method to model the uncertain and nonlinear inputs in order to obtain the smoother results.
Chapter 4: Results and discussion

In this chapter, the results for the ANN and ANFIS modeling are presented and their performances are compared to each other.

4.1 Artificial Intelligence Fitting Results

Two different data sets are chosen for modeling the solubility of carbon dioxide and ethane in 18 different ionic liquids. Artificial intelligence for the estimation of the solubility of aforementioned gases in ionic liquids is applied using the following methods: 1) Artificial Neural Network (ANN) and 2) Adaptive Neuro Fuzzy Inference Systems (ANFIS). In the following sections, the results are presented and discussed for each method in detail.

4.1.1 CO₂ solubility using ANN

Introducing the inputs and outputs is the first step to develop an ANN network. To this end, the critical properties of the chosen ionic liquids which are T_c for critical temperature, and P_c that stands for critical pressure, and \( \omega \) representing the acentric factor are used to help the model for recognizing the ionic liquids, temperature and pressure were chosen as the inputs of the model while the solubility of carbon dioxide was selected as the network output.

Modeling the ANN network has been performed with 70% randomly selected data of the database for training the network, and the rest 15% for validation and 15% for testing the modeled network. In order to provide the best optimized network, the program has been executed with one and two layers and with different numbers of neurons and their
performances were obtained. The script code for applying neural network method is presented in appendix A. The performance of the employed models is presented in appendix B. According to the collected performances which the MSE was the considered parameter to select the best model, the best observed performance belongs to the model with two hidden layers containing 6 neurons in each layer.

An illustration of the selected model is as follows:

![Figure 4-1. CO₂ function fitting neural network](image)

The weights and bias values for the first and the second layer of the generated ANN network are presented in the following tables (4.1 & 4.2) in which the value α, in table 4.2, represents the assigned weights for the solubility of the carbon dioxide:

<table>
<thead>
<tr>
<th>Neuron</th>
<th>( T_c )</th>
<th>( P_c )</th>
<th>( \omega )</th>
<th>( T )</th>
<th>( P )</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.04282</td>
<td>-0.05015</td>
<td>0.012871</td>
<td>-0.33052</td>
<td>0.16049</td>
<td>-1.09044</td>
</tr>
<tr>
<td>2</td>
<td>1.633165</td>
<td>0.91987</td>
<td>2.255374</td>
<td>-0.05703</td>
<td>0.17635</td>
<td>-1.22438</td>
</tr>
<tr>
<td>3</td>
<td>1.464972</td>
<td>0.686434</td>
<td>-4.28151</td>
<td>2.389583</td>
<td>-3.73876</td>
<td>1.063735</td>
</tr>
<tr>
<td>4</td>
<td>-0.00206</td>
<td>0.022418</td>
<td>-0.00832</td>
<td>0.140853</td>
<td>0.202787</td>
<td>0.368349</td>
</tr>
<tr>
<td>5</td>
<td>-0.02875</td>
<td>-0.6616</td>
<td>2.00408</td>
<td>-5.22046</td>
<td>9.176857</td>
<td>5.018038</td>
</tr>
<tr>
<td>6</td>
<td>1.042688</td>
<td>-1.37956</td>
<td>-1.42055</td>
<td>0.179072</td>
<td>-0.07089</td>
<td>2.172378</td>
</tr>
</tbody>
</table>
Using bias makes it easier for the neural network to work efficiently. Biases are values associated with each node in the input and hidden of a network, but in practice are treated in exactly the same manner as other weights. The use of biases in a neural network increases the capacity of the network to solve problems. Accordingly, the weights and the bias values were obtained from the generated network.

Table 4-2. Weight and bias for the ANN for the second hidden layer for CO$_2$

<table>
<thead>
<tr>
<th>Neurons</th>
<th>2nd hidden layers weights</th>
<th>Output layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>First layer</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>-3.290</td>
<td>-2.107</td>
</tr>
<tr>
<td>2</td>
<td>-1.426</td>
<td>0.985</td>
</tr>
<tr>
<td>3</td>
<td>-0.660</td>
<td>-0.403</td>
</tr>
<tr>
<td>4</td>
<td>-0.743</td>
<td>0.608</td>
</tr>
<tr>
<td>5</td>
<td>-0.441</td>
<td>-0.682</td>
</tr>
<tr>
<td>6</td>
<td>-1.818</td>
<td>1.290</td>
</tr>
</tbody>
</table>

The training and validation state is provided in figure 4.2. The training state plot shows the progress of other training variables, such as the gradient magnitude, the number of validation checks, etc. validation fail is the iteration when the validation MSE increases and Matlab will stop training procedure if the failure happens in 6 iterations in a row.
Figure 4-2. CO₂ ANN Training state

The figure below illustrates performance improvements during training, validation and testing process. As it can be seen, the performance of the training, validation and testing improves until epoch 74 in which the performance stay still after epoch 74.

Figure 4-3. CO₂ performance improvement

In order to validate the network performance, complementary investigations have been performed in Figure 4.4. The following regression plots display the network outputs with respect to targets for training, validation, and test sets. For a perfect fit, the data
should fall along a 45 degree line, where the network outputs are equal to the targets. For this case, the fit is reasonably good for all data sets, with $R^2$ values in each case of 0.99 or above.

Figure 4-4. CO$_2$ Training, validation and test regression
Regression figure illustrates the correlation between the target values for the experimental data set and the output values generated by the ANN. MSE value and coefficient of determination \((R^2)\) for the test data set are \(6.64 \times 10^{-5}\) and 0.9938, respectively. According to figure 4-5, the estimated outputs are in the excellent agreement with the experimental target data set.

![Figure 4-5. CO\(_2\) target-output regression](image)

The error histogram is shown in figure 4.6 for additional verification of network performance. The blue bars represent the training data, the green bars represent the validation data, and the red bars represent the testing data. The histogram can give us an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, you can see that while most errors fall between -0.0140 and
0.01926, there are training points with errors of -0.023 and 0.0442. These outliers are also visible on the testing regression plot. As no evidence was observed suggesting that the outliers are occurred due to experimental error, none of them has been omitted or slipped in this study.

![Error Histogram with 20 Bins](image)

Figure 4-6. CO₂ Error histogram

After fitting a satisfactory artificial neural network, a neural network function has been generated in Matlab (version R2016 a). The complex function is presented in appendix C.

The Modeled ANFIS structure for CO₂ solubility will be discussed further in detail in the following section.
4.1.2 CO$_2$ solubility using ANFIS

In order to generate the FIS structure for CO$_2$ data points, a Sugeno-type FIS structure using subtractive clustering and Gaussian membership function were utilized. Among the 514 data point, 470 data were assigned for training the FIS and the rest were employed to test and check the generated FIS structure. As for training the membership function parameters, the hybrid optimization method was selected to emulate the training data as this method is a combination of least-squares and gradient method. The rule extraction method first uses the subclust function to determine the number of rules and antecedent membership functions and then uses linear least squares estimation to determine each rule's consequent equations. The summarized details of the FIS structure which is trained to predict the solubility of the carbon dioxide is presented in the table 4.3.

Table 4-3. Details of the trained ANFIS for CO$_2$

<table>
<thead>
<tr>
<th>Type</th>
<th>Value/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership function</td>
<td>Gaussmf</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>128</td>
</tr>
<tr>
<td>Number of linear parameters</td>
<td>60</td>
</tr>
<tr>
<td>Number of non-linear parameters</td>
<td>100</td>
</tr>
<tr>
<td>Total number of parameters</td>
<td>160</td>
</tr>
<tr>
<td>Number of training parameters</td>
<td>470</td>
</tr>
<tr>
<td>Number of checking parameters</td>
<td>45</td>
</tr>
<tr>
<td>Number of Fuzzy rules</td>
<td>10</td>
</tr>
<tr>
<td>Train Epoch</td>
<td>26</td>
</tr>
<tr>
<td>Optimization method</td>
<td>Hybrid</td>
</tr>
</tbody>
</table>

The training process of the generated FIS applied to 1000 epochs and after the 26$^{th}$ iteration the performance of the FIS reached the minimum amount and the value of the
checking data set’s error remained constant. The mean squared error of the training and
checking data points was obtained $7.23 \times 10^{-5}$ and $1.23 \times 10^{-5}$, respectively. The
following figures illustrate the training and the checking processes of the generated FIS
structure.

![Figure 4-7. CO$_2$ training performance of the generated FIS](image)

![Figure 4-8. CO$_2$ checking performance of the generated FIS](image)

Figure 4.9 shows the regression plot of the estimated output results and the
experimental target values of the solubility of the carbon dioxide. According to the results
shown in the regression plot, the experimental targets and the obtained output by the FIS are in well agreement with each other, and the coefficient of determination ($R^2$) value was 0.996.

![Regression Plot](image)

Figure 4-9. CO$_2$ regression plot: Target vs Output

The residuals histogram of the predicted and the experimental values are shown in figure 4.10. The histogram can give an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, it can be seen that while most errors fall between -0.02 and 0.02, there are points with errors between -0.08 to -0.1 and 0.04 – 0.06. These outliers are also visible on the testing regression plot.
Figure 4-10. CO$_2$ error histogram

Figure below briefly represent the rules and the membership functions have been employed to each input and the outputs. This function displays a high level diagram of an FIS. Inputs and their membership functions appear to the left of the FIS structural characteristics, while outputs and their membership functions appear on the right.
Figure 4.11. Fuzzy Rules on the input and output variables for CO₂

Figure 4.12 displays a roadmap of the whole fuzzy inference process. A single figure window with 60 plots nested in it. Each rule is a row of plots, and each column is a variable. The rule numbers for this generated FIS are 10.

- The first five columns of plots (the 10 yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule.
- The sixth column of plots (the 10 blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule.

Each of the characterizations of each of the variables is specified with respect to the input index line in this manner. The aggregation occurs down the sixth column, and the resultant aggregate plot is shown in the single plot appearing in the lower right corner of
the plot field. The defuzzified output value is shown by the thick line passing through the aggregate fuzzy set.

Figure 4-12. Fuzzy rule for the ANFIS model for CO₂

Sensitivity of CO₂ solubility to each factor in depicted in figure 4.13. The diagram suggests that the reliability of solubility of carbon dioxide to the critical temperature is limited as it can be seen in the output axel of the shown Tₑ versus the output. Furthermore, CO₂ dissolves better in a specific IL as the temperature is reduced and the pressure is raised. Moreover, as it is shown in the figure, the solubility of the acid gas increased where the critical pressure is in optimum level, between the 1.5 to 2 MPa. Acentric factor does not have a steady effect on CO₂ solubility in ILs. More specifically, the diagram suggests that there is a worst level for acentric factor.
Figure 4-13. Sensitivity analysis of solubility of CO$_2$ to each variable
In order to represent the results of the sensitivity analysis, the figure for the solubility of carbon dioxide in ILs versus the pressure at the fixed temperature (313.15 K) is depicted in the figure 4.14. Regarding the obtained results from the sensitivity analysis and the shown figure, for the ILs with the critical pressure between the 1.5 to 2.5 (MPa), attained the highest solubility of the carbon dioxide in

![Figure 4-14. Solubility of carbon dioxide versus pressure at 313.15 K](image)

As it was mentioned in the individual effect of the input parameters on solubility of CO₂, the solubility of the acid gas is increased while critical pressure of the chosen ILs is in between 1.5 to 2.5 MPa while according to the mixed effect figures of the input parameters, in the lower the critical pressure, the higher critical temperature lead the solubility to be raised. In the other world, according to the following able, there are some
ILs which did not follow the pattern of the aforementioned individual effect and that is because of the mixed effect of the critical temperature on the critical pressure that led the lower solubility.

Table 4-4. Order of solubility & the critical properties of the ILs

<table>
<thead>
<tr>
<th>#</th>
<th>ILs</th>
<th>P_c</th>
<th>T_c</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[TDC] [TF2N]</td>
<td>1.803</td>
<td>1255.7</td>
</tr>
<tr>
<td>2</td>
<td>[p(5)mpyr][Tf2N]</td>
<td>1.828</td>
<td>1221.9</td>
</tr>
<tr>
<td>3</td>
<td>[P4441][Tf2N]</td>
<td>1.173</td>
<td>1155</td>
</tr>
<tr>
<td>4</td>
<td>[deme][Tf2N]</td>
<td>2.35</td>
<td>1080.5</td>
</tr>
<tr>
<td>5</td>
<td>[EMMP] [TF2N]</td>
<td>2.588</td>
<td>1038.7</td>
</tr>
<tr>
<td>6</td>
<td>[pmim][Tf2N]</td>
<td>3.00</td>
<td>1259.1</td>
</tr>
<tr>
<td>7</td>
<td>[amim][Tf2N]</td>
<td>3.12</td>
<td>1266.8</td>
</tr>
<tr>
<td>8</td>
<td>[PMPY] [TF2N]</td>
<td>2.755</td>
<td>1234.2</td>
</tr>
<tr>
<td>9</td>
<td>[bmmim][Tf2N]</td>
<td>2.031</td>
<td>1255.8</td>
</tr>
<tr>
<td>10</td>
<td>[S222][Tf2N]</td>
<td>2.19</td>
<td>1119.1</td>
</tr>
<tr>
<td>11</td>
<td>[EMIM] [LACTATE]</td>
<td>2.824</td>
<td>912.7</td>
</tr>
<tr>
<td>12</td>
<td>[TDC] [DCN]</td>
<td>1.615</td>
<td>1073.7</td>
</tr>
<tr>
<td>13</td>
<td>[(CH2)4SO3HMIm]</td>
<td>3.27</td>
<td>1612.8</td>
</tr>
<tr>
<td>14</td>
<td>[4mbp][BF4]</td>
<td>1.89</td>
<td>625.8</td>
</tr>
<tr>
<td>15</td>
<td>[(CH2)4SO3HMIm]</td>
<td>2.588</td>
<td>1433.0</td>
</tr>
</tbody>
</table>
Figure 4-15. Surface view of CO$_2$ input parameters to the output
In order to examine mixed effects of the parameters, a three-dimensional sensitivity analysis is considered in figure 4.15. As expected, CO\textsubscript{2} solves better in a specific IL as the pressure is raised and the temperature is reduced. The diagram also suggests that the effect of critical temperature on solubility of CO\textsubscript{2} is quite limited. Moreover, the mixed effect of acentric factor and critical temperature seems complex.

4.1.3 ANN vs ANFIS comparison

Regarding the comparison between the performance of the presented carbon dioxide solubility prediction models by ANN and ANFIS, the mean squared error (MSE), standard deviation (STD), coefficient of determination (R\textsuperscript{2}) and the mean absolute error (MAE) were computed which all the mentioned parameters were summarized in the table 4.5:

<table>
<thead>
<tr>
<th>Analysis</th>
<th>ANN</th>
<th>ANFIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>6.93 \times 10^{-5}</td>
<td>6.72 \times 10^{-5}</td>
</tr>
<tr>
<td>MAE</td>
<td>0.0102</td>
<td>0.0042</td>
</tr>
<tr>
<td>STD</td>
<td>0.0168</td>
<td>0.0082</td>
</tr>
<tr>
<td>R\textsuperscript{2}</td>
<td>0.992813</td>
<td>0.992992</td>
</tr>
</tbody>
</table>

Table 4.5 represents main performance measures of ANN and ANFIS modeling analysis. The ANN and ANFIS models both performed very well in modeling and predicting the solubility of CO\textsubscript{2}. For instance, MAE and R\textsuperscript{2} of both models show that we can predict the solubility of carbon dioxide in different ILs with great precision.

According to the obtained statistical results shown in table 4.5, the performance of the presented models in comparison to the other introduced models’ performances in the
literature are much better. In the following table the performance of the other presented models in the literatures are summarized in brief.

<table>
<thead>
<tr>
<th>Study</th>
<th>ANN</th>
<th>ANFIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current study</td>
<td>6.93 × 10^{-5}</td>
<td>0.992813</td>
</tr>
<tr>
<td>Baghban, et al., 2015</td>
<td>0.000133</td>
<td>0.9972</td>
</tr>
<tr>
<td>Ahmadi et al., 2014</td>
<td>6.6507 × 10^{-5}</td>
<td>0.997594</td>
</tr>
<tr>
<td>Bahmani et al., 2015</td>
<td>0.00005</td>
<td>0.98612</td>
</tr>
<tr>
<td>Hamzehie et al., 2016</td>
<td>0.6328 × 10^{-3}</td>
<td>0.9975</td>
</tr>
<tr>
<td>Norouzbahari et al., 2015</td>
<td>0.6931 × 10^{-3}</td>
<td>0.9977</td>
</tr>
<tr>
<td>(Shafiei, et al., 2014)</td>
<td>0.00025</td>
<td>0.95151</td>
</tr>
<tr>
<td>Hamzehie et al., 2014</td>
<td>1.406 × 10^{-4}</td>
<td>0.995</td>
</tr>
</tbody>
</table>

### 4.1.4 Ethane solubility using ANN

To develop an ANN network, introducing the inputs and outputs is the first step. Therefore, the critical properties of the ionic liquids which are $T_c$ for critical temperature, $P_c$ stands for critical pressure and $\omega$ represents the acentric factor to help the model for Characterizing the ionic liquids, temperature and pressure were chosen as the inputs of the model and the solubility of carbon dioxide was selected as the network output.

To assure that the data are randomly assigned for training, validation and testing purposes, the data set is reordered randomly before being allocated. Thus, modeling the
ANN network has been performed with 70% data points for training the network and the rest (15%) for validation and 15% for testing the modeled network, which have not been used and tested during the training process of the network. In order to provide the best optimized network, the program has been executed with one and two layers and with different number of neurons and their performances based on parameter MSE, have been collected. The script code for applying neural network method is presented in appendix D. The performance of different employed models has been showed in appendix E. According to the collected performances which the MSE was the considered parameter to select the best model, the best observed performance belongs to the model with ONE hidden layer with 5 neurons. As can be seen, an illustration of the selected model is as follows:

![Network Diagram](image)

Figure 4-16. Ethane Function fitting neural network

As for detail explanation of the following table, each input that is sent into the neuron must be weighted by multiplying it by some value (often a number between -1 and 1), when creating the network. Afterwards, the created network will require a third input, typically referred to as a bias input. A bias input always has the value of 1 and is also weighted. The weights and bias values for the layer of the generated ANN network is presented in the following table (4.7), in which the value $\alpha$, represents the assigned weights for the solubility of carbon dioxide
Table 4-7. Weight and bias values of the ethane ANN model

<table>
<thead>
<tr>
<th>Neuron</th>
<th>1st Layer’s weights</th>
<th>Output layer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Weights</td>
<td></td>
</tr>
<tr>
<td></td>
<td>T&lt;sub&gt;c&lt;/sub&gt;</td>
<td>P&lt;sub&gt;c&lt;/sub&gt;</td>
</tr>
<tr>
<td>1</td>
<td>-1.257</td>
<td>1.526</td>
</tr>
<tr>
<td>2</td>
<td>-2.115</td>
<td>-0.438</td>
</tr>
<tr>
<td>3</td>
<td>1.711</td>
<td>0.224</td>
</tr>
<tr>
<td>4</td>
<td>-0.322</td>
<td>-0.685</td>
</tr>
<tr>
<td>5</td>
<td>0.177</td>
<td>-0.312</td>
</tr>
</tbody>
</table>

The training and validation state is provided in figure 4.17. Gradient is a value of backpropagation gradient on each iteration. As it can be seen, the validation fail is the iteration where the performance of validation increases its value and the utilized software automatically stops the training process after six failures in a row.

Figure 4-17. ANN Training state for ethane parameters

Figure 4.18 shows performance improvements during training as well as validation and testing process. As the epochs were increasing, the performance of the training,
validation and testing improved until epoch 19 where the performance remains the same after the mentioned epoch.

![Graph](image.png)

**Figure 4-18. Ethane performance improvement**

In order to validate the network performance, complementary investigations have been performed in Figure 4.19. The following regression plots display the network outputs with respect to targets for training, validation, and test sets. For a perfect fit, the data should fall along a 45 degree line, where the network outputs are equal to the targets. For this problem, the fit is reasonably good for all data sets, with coefficient of determination ($R^2$) values in each case of 0.9585 or higher.
Regression figure illustrates the correlation between the target values for the experimental data set and the output values generated by the ANN. MSE value and coefficient of determination ($R^2$) for the test data set are $1.53 \times 10^{-6}$ and 0.9585 respectively.

According to the figure 4.20, the estimated outputs are in the perfect agreement with the experimental target data set.
The error histogram is depicted in figure 4.20 for additional verification of network performance. The blue bars represent training data, the green bars represent validation data, and the red bars represent testing data. The histogram can give us an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, there are training points with errors of 0.00464, 0.005937 and 0.01891. These outliers are also visible on the test regression plot. As discussed earlier for CO$_2$ model, there is also no evidence was observed for ethane data set suggesting that the outliers are occurred due to experimental error, none of them has been omitted or slipped in this study.
After fitting a satisfactory artificial neural network, a neural network function has been generated in Matlab. The complicated function is presented in appendix F.

The Modeled ANFIS structure for C\textsubscript{2}H\textsubscript{6} solubility will be discussed further in detail in the following section.

4.1.5 Ethane solubility using ANFIS

In order to generate the FIS structure for C\textsubscript{2}H\textsubscript{6} data points, a Sugeno-type FIS structure using subtractive clustering and Gaussian membership function were utilized. Among the 90 data point, 65 data were assigned for training the FIS and the rest were employed to test and check the generated FIS structure. As for training the membership function parameters, the hybrid optimization method was selected to emulate the training data.
making this method a combination of least-squares and gradient method. The rule extraction method first uses the subclust function to determine the number of rules and antecedent membership functions, and then uses linear least squares estimation to determine each rule's consequent equations. The summarized details of the FIS structure which is trained to predict the solubility of the carbon dioxide is presented in table 4.8 below:

Table 4-8. Details of the ethane trained ANFIS

<table>
<thead>
<tr>
<th>Type</th>
<th>Value/Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Membership function</td>
<td>Gaussmf</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>116</td>
</tr>
<tr>
<td>Number of linear parameters</td>
<td>54</td>
</tr>
<tr>
<td>Number of non-linear parameters</td>
<td>90</td>
</tr>
<tr>
<td>Total number of parameters</td>
<td>144</td>
</tr>
<tr>
<td>Number of training parameters</td>
<td>65</td>
</tr>
<tr>
<td>Number of checking parameters</td>
<td>30</td>
</tr>
<tr>
<td>Number of Fuzzy rules</td>
<td>9</td>
</tr>
<tr>
<td>Train Epoch</td>
<td>14</td>
</tr>
<tr>
<td>Optimization method</td>
<td>Hybrid</td>
</tr>
</tbody>
</table>

The training process of the generated FIS applied to 1000 epochs, and after the 14th iteration, the performance of the FIS reached the minimum amount and the value of the checking data set error remained constant. The mean squared error of the training and checking data points was obtained $7.84 \times 10^{-6}$ and $1.29 \times 10^{-5}$. The following figures illustrate the training and the checking processes of the generated FIS structure.
Figure 4-22. $\text{C}_2\text{H}_6$ training performance of the generated FIS

Figure 4-23. $\text{C}_2\text{H}_6$ checking performance of the generated FIS
Figure 4.24 shows the regression plot of the estimated output results and the experimental target values of the solubility of the Ethane. According to the results shown in the regression plot, the experimental targets and the obtained output by the FIS are in good agreement with each other and the coefficient of determination ($R^2$) value was 0.956.

The residuals histogram of the predicted and the experimental values is shown figure 4.25. The histogram can give us an indication of outliers, which are data points where the fit is significantly worse than the majority of data. In this case, it can be seen that while most errors fall between -0.005 and 0.005, there are points with errors between 0.01. These outliers are also visible on the testing regression plot.
The figure below briefly represents the rules and the membership functions that have been employed to each input and the outputs. This function displays a high level diagram of an FIS. Inputs and their membership functions appear to the left of the FIS structural characteristics, while the outputs and their membership functions appear on the right.
Figure 4.26. Fuzzy Rules on the input and output variables of ethane

Figure 4.27 displays a roadmap of the whole fuzzy inference process, it is a single figure window with 54 plots nested in it. Each rule is a row of plots, and each column is a variable. The rule numbers for this generated FIS are 9.

- The first five columns of plots (the 45 yellow plots) show the membership functions referenced by the antecedent, or the if-part of each rule.
- The sixth column of plots (the 9 blue plots) shows the membership functions referenced by the consequent, or the then-part of each rule.

Each of the characterizations of each of the variables is specified with respect to the input index line in this manner. The aggregation occurs down the sixth column, and the resultant aggregate plot is shown in the single plot appearing in the lower right corner of the plot field. The defuzzified output value is shown by the thick line passing through the aggregate fuzzy set.
Sensitivity of ethane solubility to each factor is reported in figure 4.28. The diagram suggests that ILs with lower critical temperature, higher critical pressure and higher acentric factor are better solvents for ethane. Furthermore, ethane absorbs better in a specific IL as the temperature is reduced. It seems that the effect of pressure change on ethane solubility in ILs is not steady. According to the diagram, as pressure rises, solubility soars to its peak; however, more pressure rise causes a sharp fall to a local minimum. Any pressure increase after that point leads to slight improvement in solubility.
Figure 4-28. Sensitivity of solubility of ethane to each parameter

In order to represent the results of the sensitivity analysis, the predicted solubility of ethane in three different ionic liquids versus the pressure at the fixed temperature (303.15 K) is drawn and shown in figure 4.29. According to the figure and the results of the sensitivity analysis, the lower the critical temperature, the higher the solubility of ethane will be.
Figure 4-29. Solubility of ethane versus pressure at 303.15 K

Table 4.9 shows the calculated amount of the critical properties (critical temperature and critical pressure) of the named ILs for solubility of ethane.

Table 4-9. Order of solubility & the critical properties of the ILs

<table>
<thead>
<tr>
<th>#</th>
<th>ILs</th>
<th>$T_c$</th>
<th>$P_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>[N1223][FSI]</td>
<td>1059.82</td>
<td>3.801</td>
</tr>
<tr>
<td>2</td>
<td>[mp(3)pyrr][FSI]</td>
<td>1217.07</td>
<td>4.443</td>
</tr>
<tr>
<td>3</td>
<td>[mp(3)pip][FSI]</td>
<td>1240.28</td>
<td>4.017</td>
</tr>
</tbody>
</table>

In order to examine the mixed effects of the variables, a three-dimensional sensitivity analysis is considered as well. Figure 4.30 depicts the mixed effect of the factors that show significant cross-effect.
Figure 4-30. Surface view of ethane input parameters to the output
The acentric factor seems to be significantly effective only at low pressure. Moreover, the diagram suggests that for ILs with low acentric factor, a higher pressure always has a favorable effect on the solubility of ethane. The same trend can be observed regarding the mixed-effect of pressure and temperature on solubility.

4.1.6 ANN vs ANFIS comparison

Regarding the comparison between the performance of the presented carbon dioxide solubility prediction models by ANN and ANFIS, the mean squared error (MSE), standard deviation (STD), coefficient of determination ($R^2$), and the mean absolute error (MAE) were computed, and all the mentioned parameters are summarized in the table 4.10.

<table>
<thead>
<tr>
<th>Analysis</th>
<th>ANN</th>
<th>ANFIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>$7.94 \times 10^{-6}$</td>
<td>$1.07 \times 10^{-5}$</td>
</tr>
<tr>
<td>MAE</td>
<td>0.0014</td>
<td>0.0379</td>
</tr>
<tr>
<td>STD</td>
<td>0.0028</td>
<td>0.0041</td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.9585</td>
<td>0.921196</td>
</tr>
</tbody>
</table>

Table 4-10. Ethane ANFIS vs ANN

Table 4.10 represents the main performance measures of ANN and ANFIS modeling analysis. As tabulated, ANN and ANFIS models both performed well in modeling and predicting the solubility of C$_2$H$_6$ and MSE and $R^2$ confirm this fact.

Moreover, in table 4.11, the average absolute deviation values for SRK and PR EoS and the proposed ANN & ANFIS approaches are compared to each other. The results suggest that the models used to predict CO$_2$ solubility performed better than the other thermodynamic models based on the obtained ADD% values.
Table 4-11. Comparison of AAD% values for SRK and PR EOS and the proposed ANN & ANFIS approaches

<table>
<thead>
<tr>
<th>Compound</th>
<th>PR EOS</th>
<th>SRK EOS</th>
<th>ANN</th>
<th>ANFIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[mp(3)pip][FSI]</td>
<td>3.35</td>
<td>-</td>
<td>3.29</td>
<td>3.31</td>
</tr>
<tr>
<td>[mp(3)pyrr][FSI]</td>
<td>3.06</td>
<td>-</td>
<td>1.53</td>
<td>1.59</td>
</tr>
<tr>
<td>[N1223][FSI]</td>
<td>3.24</td>
<td>-</td>
<td>3.06</td>
<td>3.10</td>
</tr>
<tr>
<td>[PMPY][Tf2N]</td>
<td>5.1</td>
<td>-</td>
<td>0.72</td>
<td>0.82</td>
</tr>
<tr>
<td>[(CH2)4SO3HMIm][TF2N]</td>
<td>6.63</td>
<td>-</td>
<td>1.14</td>
<td>1.40</td>
</tr>
<tr>
<td>[EMMP][TF2N]</td>
<td>2.2</td>
<td>0.83</td>
<td>1.07</td>
<td>1.00</td>
</tr>
<tr>
<td>[TDC][TF2N]</td>
<td>3.1</td>
<td>1.0</td>
<td>0.61</td>
<td>0.68</td>
</tr>
<tr>
<td>[S222][Tf2N]</td>
<td>2.3</td>
<td>2.1</td>
<td>0.89</td>
<td>0.91</td>
</tr>
<tr>
<td>[deme][Tf2N]</td>
<td>2.3</td>
<td>2.2</td>
<td>0.89</td>
<td>0.82</td>
</tr>
<tr>
<td>[pmim][Tf2N]</td>
<td>1.9</td>
<td>1.8</td>
<td>0.54</td>
<td>0.50</td>
</tr>
<tr>
<td>[amim][Tf2N]</td>
<td>2.2</td>
<td>2.1</td>
<td>0.61</td>
<td>0.59</td>
</tr>
<tr>
<td>[4mbp][BF4]</td>
<td>1.0</td>
<td>1.0</td>
<td>0.99</td>
<td>0.98</td>
</tr>
</tbody>
</table>

4.2 Summary

In this work, artificial intelligence has been applied as a technique to develop predictive models to be substituted for the other thermodynamic models. These models using ANN and ANFIS methods generate the functions which are able to compute and predict the output (CO$_2$ and C$_2$H$_6$ mole fraction) of the solubility of acid gases in ionic liquids.

The following can be pointed out as the most important findings of this study:

- Neural networks can be considered as an efficient tool capable of learning the behaviour of a system through the provided data. These data must be collected
wisely so the system behaviour can be captured and captured by the network. To this end, new and recent experimental CO$_2$ solubility data were collected and modeled utilizing ANN and ANFIS.

- In the cases where a portion of data must be selected due to the high dimensionality of the data set, application of proper data sampling method is of significant importance. Therefore, in this study more simulation runs are required to be included in the training (70% of the collected data) so that the variability of the system can be properly introduced to the network through sufficient sampled data, and the rest 30% of data were allocated for training and testing of the network (each 15%).

- As for the case of the solubility of ethane in ILs over a new set of collected experimental data, the neural network (ANN & ANFIS) can be more efficiently trained to capture the effect of variabilities of the ILs.

- A sensitivity analysis that included individual and mixed-effects of the chosen variables was studied to represent the dependency of solubility of CO$_2$ and ethane to each input parameter.
Chapter 5: Conclusion

In this study, the prediction of the solubility of $\text{CO}_2$ and $\text{C}_2\text{H}_6$ in ILs for natural gas sweetening and flue gas treatment utilizing the ANN and ANFIS models has been performed. The ability of the ANN and ANFIS to predict the solubility of gases ($\text{CO}_2$ and $\text{C}_2\text{H}_6$) in different ionic liquids was assessed in this study. The critical properties of the ILs such as critical temperature ($T_c$) and critical pressure ($P_c$), acentric factor ($\omega$) and operating conditions ($P$ and $T$) were considered as inputs. Recent solubility data points of carbon dioxide and ethane, collected from the literature, which have not been investigated in previous simulation studies, were allocated as the output parameters of the models. A total of 514 data points for the solubility of carbon dioxide and 90 data points for the solubility of ethane were included in this study.

As for selecting the best neural networks and FIS for both carbon dioxide and ethane, the mean squared error (MSE), standard deviation (STD), and the mean absolute error (MAE) were the parameters selected to help choose the best model. To this end, statistical factors showed that both models (ANN & ANFIS) were able to estimate the solubility of the mentioned gases with very low mean squared errors. Furthermore, according to the sensitivity analysis, ILs with lower critical temperatures were better solvents of $\text{CO}_2$. Moreover, $\text{CO}_2$ dissolves better in a given IL at low temperature and high pressure, as is the case for most physical solvents. While sensitivity analysis shows that ILs with lower critical temperature, higher critical pressure and higher acentric factor are better solvents for ethane. In addition, like $\text{CO}_2$, ethane also has a higher solubility in
a specific IL at a lower temperature. The effect of pressure changes on ethane solubility in ILs is not obvious. As the pressure rose, solubility seemed to reach a peak value.

The ability to estimate and predict the solubility of other gases in other systems within a wide range of temperature and pressure is one of the major advantages of using the ANN and ANFIS modeling in comparison to the other thermodynamic models. Using this type of soft computing modeling can reduce time and cost of experimental studies, with no further restriction on the operating conditions, by applying and utilizing the discussed ANN & ANFIS networks, the solubility in different systems of acid and other gases in ILs can be predicted. On the other hand, there are some points which also can be considered as challenges in this study such as: the selection of experimental data in this study entails restrictions and can lead to errors, and the functions generated using either ANN or ANFIS also contain noises.

**Future work**

While this study has provided potential models to predict the solubility of specific gases (CO₂ and C₂H₆) in different ionic liquids, many opportunities remain to extend the scope of this study. Although the main factors which can affect the solubility of acid gases in ionic liquids have been examined and modeled in this study but there are still some other characteristics of ILs that can be used such as molecular weight of the pure ionic liquids and the ionic strength that can be considered as the other effective parameters in solubility of gases in ionic liquids. Also, due to recent growing attention to the solubility of acid gases in mixtures of ILs and amines, the implementation of soft computing such as artificial neural network and adaptive neuro fuzzy systems to the
solubility of gases on the above-mentioned chemical solvents might be a possible future study in this area.
References


Kim, J. E., Joo Kim, H., & Lim, J. S. (2014). Solubility of CO2 in ionic liquids containing cyanide anions: [c2mim][SCN], [c2mim][N(CN)2], [c2mim][C(CN)3], *Fluid Phase Equilibria*, vol 367, 151–158.


Appendix A: Neural Network Fitting Script Program (Carbon Dioxide)

```matlab
x=x';
t=y';
trainFcn='trainlm';
hiddenLayerSize=[6 6];
net=fitnet(hiddenLayerSize, trainFcn);
net.input.processFcns={'removeconstantrows','mapminmax'};
net.output.processFcns={'removeconstantrows','mapminmax'};
net.divideFcn='dividerand';
net.divideMode='sample';
net.divideParam.trainRatio=70/100;
net.divideParam.valRatio=15/100;
net.divideParam.testRatio=15/100;
net.performFcn='mse';

net.plotFcns={'plotperform','plottrainstate','ploterrhist',..., 'plotregression','plotfit'};

[net,tr]=train(net,x,t);

y=net(x);
e=gsbsubtract(t,y);
performance=perform(net,t,y);

trainTargets=t.*tr.trainMask{1};
valTargets=t.*tr.valMask{1};
testTargets=t.*tr.testMask{1};
trainPerformance=perform(net,trainTargets,y)
valPerformance=perform(net,valTargets,y)
testPerformance=perform(net,testTargets,y)

view(net)
figure,plotperform(tr)
figure,plottrainstate(tr)
figure,ploterrhist(e)
figure,plotregression(t,y)
figure,plotfit(net,x,t)
```
Appendix B: Neural Network Performance Table (Carbon Dioxide)

One layer performances:

<table>
<thead>
<tr>
<th>No. of neurons in 1&lt;sup&gt;st&lt;/sup&gt; layer</th>
<th>No. of neurons in 2&lt;sup&gt;nd&lt;/sup&gt; layer</th>
<th>Performance (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>0</td>
<td>0.000469598</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0.000335969</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0.000412632</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0.000255149</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0.000259865</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0.000229614</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0.000245193</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0.000218231</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>0.000203438</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>0.000209428</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>0.000188306</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>0.000285824</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>0.000189915</td>
</tr>
</tbody>
</table>

Two layer performances:

<table>
<thead>
<tr>
<th>No. of neurons in 1&lt;sup&gt;st&lt;/sup&gt; layer</th>
<th>No. of neurons in 2&lt;sup&gt;nd&lt;/sup&gt; layer</th>
<th>Performance (MSE)</th>
<th>No. of neurons in 1&lt;sup&gt;st&lt;/sup&gt; layer</th>
<th>No. of neurons in 2&lt;sup&gt;nd&lt;/sup&gt; layer</th>
<th>Performance (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>0.000785743</td>
<td>2</td>
<td>9</td>
<td>0.000912198</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.000869758</td>
<td>2</td>
<td>10</td>
<td>0.000616053</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>0.000870963</td>
<td>2</td>
<td>11</td>
<td>0.000831746</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>0.000862344</td>
<td>2</td>
<td>12</td>
<td>0.000865731</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>0.000580947</td>
<td>3</td>
<td>2</td>
<td>0.000300143</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>0.000817823</td>
<td>3</td>
<td>3</td>
<td>0.000366428</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>0.000601786</td>
<td>3</td>
<td>4</td>
<td>0.000391174</td>
</tr>
<tr>
<td>No. of neurons in 1\textsuperscript{st} layer</td>
<td>No. of neurons in 2\textsuperscript{nd} layer</td>
<td>Performance (MSE)</td>
<td>No. of neurons in 1\textsuperscript{st} layer</td>
<td>No. of neurons in 2\textsuperscript{nd} layer</td>
<td>Performance (MSE)</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-------------------</td>
<td>---------------------------------------------</td>
<td>---------------------------------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>0.000403929</td>
<td>7</td>
<td>4</td>
<td>0.000180765</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>0.00033457</td>
<td>7</td>
<td>5</td>
<td>0.000203495</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>0.000341453</td>
<td>7</td>
<td>6</td>
<td>0.000240699</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>0.000378719</td>
<td>7</td>
<td>7</td>
<td>0.000211083</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>0.000572531</td>
<td>7</td>
<td>8</td>
<td>0.000287458</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>0.000270764</td>
<td>7</td>
<td>9</td>
<td>0.000190232</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
<td>0.000261947</td>
<td>7</td>
<td>10</td>
<td>9.79E-05</td>
</tr>
<tr>
<td>3</td>
<td>12</td>
<td>0.000256244</td>
<td>7</td>
<td>11</td>
<td>0.00015798</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>0.000335702</td>
<td>7</td>
<td>12</td>
<td>0.000216715</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0.000273382</td>
<td>8</td>
<td>2</td>
<td>0.000227237</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0.000284094</td>
<td>8</td>
<td>3</td>
<td>0.000239196</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.000385514</td>
<td>8</td>
<td>4</td>
<td>0.000197506</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>0.000225573</td>
<td>8</td>
<td>5</td>
<td>0.000302263</td>
</tr>
<tr>
<td>4</td>
<td>7</td>
<td>0.00023849</td>
<td>8</td>
<td>6</td>
<td>0.00027205</td>
</tr>
<tr>
<td>4</td>
<td>8</td>
<td>0.000283093</td>
<td>8</td>
<td>7</td>
<td>0.000207455</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>0.000278961</td>
<td>8</td>
<td>8</td>
<td>0.000108692</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>0.000274657</td>
<td>8</td>
<td>9</td>
<td>0.000126536</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>0.000256115</td>
<td>8</td>
<td>10</td>
<td>0.000183665</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>0.000287367</td>
<td>8</td>
<td>11</td>
<td>0.000113617</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>0.000243224</td>
<td>8</td>
<td>12</td>
<td>0.0001635</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0.000251312</td>
<td>9</td>
<td>2</td>
<td>0.000188658</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>0.000284929</td>
<td>9</td>
<td>3</td>
<td>0.000174729</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0.000230056</td>
<td>9</td>
<td>4</td>
<td>0.000253789</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.000188897</td>
<td>9</td>
<td>5</td>
<td>0.000230996</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>0.000268324</td>
<td>9</td>
<td>6</td>
<td>0.000200081</td>
</tr>
<tr>
<td>5</td>
<td>8</td>
<td>0.000126779</td>
<td>9</td>
<td>7</td>
<td>9.38E-05</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>0.000265611</td>
<td>9</td>
<td>8</td>
<td>0.000245571</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>0.000294814</td>
<td>9</td>
<td>9</td>
<td>0.000291545</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>0.000162291</td>
<td>9</td>
<td>10</td>
<td>0.000312324</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>0.000257981</td>
<td>9</td>
<td>11</td>
<td>0.000230873</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>0.000274311</td>
<td>9</td>
<td>12</td>
<td>0.000162894</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>0.000282345</td>
<td>10</td>
<td>2</td>
<td>0.000227034</td>
</tr>
</tbody>
</table>
Appendix C: Neural Network Predictor Function (Carbon Dioxide)

function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
% ====== NEURAL NETWORK CONSTANTS ======
% Input 1
x1_step1.xoffset = [625.8;1.173;0.1561;298.15;0.0095];
x1_step1.gain = [0.00202634245187437;0.953743443013829;2.1822149817239;0.0444444444444444;1.05747609534183];
x1_step1.ymin = -1;
% Layer 1
b1 = [-1.090439204654515;-1.224381569566243;1.0637347971039071;0.3683486134671895;5.018037980816883;2.1723779137255805];
IW1_1 = [-0.0428157373706877 -0.050149701243448858 0.01287063859104411 -0.33052135714678965 -0.33052135714678965 0.91987041543638681 2.25537404735284 -0.057027939630530253 0.17635521368280849;1.1464971742324111 0.86843393573234124 -4.2815095202765754 2.3895834416376278 -1.7387597012710883 -0.002060714708998536 0.022417653644285166 -0.0083209196615490876 0.14085332487767885 0.202786691393605;1.028748907950540277 -0.66160176732710163 2.004804583860239 -5.2204574162605573 9.176856717127633;1.0426879388492163 -1.3795631433247126 -1.4205520787707828 0.17907226701425272 -0.70886352623242568];
% Layer 2
b2 = [4.4223975020079518;1.5382971433369479;0.94917115244091999;0.18209276142041211;0.65484231277461902;1.7882503186655034];
LW2_1 = [-3.2903677063024466 -2.1068743044016642 8.1241755779404023 0.2476739597684416 0.0675603665151359 0.9581444717217276;1.425968053985159 0.984546223045134317 0.22844180387658711 -1.1119618920014387 -0.03544419961322171 0.9931045719120567 -0.65986157910179177 -0.40342367835703408 0.35291499267717336 2.5155437795778557 -0.049656807795065663 1.1815254118816858;0.74772222307067459 0.60802738968139092 -0.65744946967399764 -1.9239393537426379 0.068355315855737286 0.43940702346927463;0.44110535195954312 -0.6817764451323326 0.088962007750709701 -0.44727168906087977 0.01769156370333467 -0.34490481006759649;1.8176149177112455 1.290295855373119 1.7068985954294691 1.62987666629013599 -0.02280760979319983 -0.628120581020713];
% Layer 3
b3 = 1.0421122639977318;
LW3_2 = [-3.342112694711199 -2.2001558325446671 -1.3675428961558831 -1.4568761307121991 -3.4644742690888206 0.27815109352721312];
% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 4.93291239147593;
y1_step1.xoffset = 0.00056;
% ======= SIMULATION =======

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS
    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = tansig_apply(repmat(b2,1,Q) + LW2_1*a1);

    % Layer 3
    a3 = repmat(b3,1,Q) + LW3_2*a2;

    % Output
    Y{1,ts} = mapminmax_reverse(a3,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(3,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end
% ===== MODULE FUNCTIONS ========
% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
    y = bsxfun(@minus,x,settings.xoffset);
    y = bsxfun(@times,y,settings.gain);
    y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
    a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
    x = bsxfun(@minus,y,settings.ymin);
    x = bsxfun(@rdivide,x,settings.gain);
    x = bsxfun(@plus,x,settings.xoffset);
end
Appendix D: Neural Network Fitting Script Program (Ethane)

\[
x' = x';
\]

\[
t' = y';
\]

\[
\text{trainFcn} = 'trainlm';
\]

\[
\text{hiddenLayerSize} = 5;
\]

\[
\text{net} = \text{fitnet} \left( \text{hiddenLayerSize}, \text{trainFcn} \right);
\]

\[
\text{net}.\text{input}.\text{processFcns} = \{ '\text{removeconstantrows}', '\text{mapminmax}' \};
\]

\[
\text{net}.\text{output}.\text{processFcns} = \{ '\text{removeconstantrows}', '\text{mapminmax}' \};
\]

\[
\text{net}.\text{divideFcn} = '\text{dividerand}';
\]

\[
\text{net}.\text{divideMode} = '\text{sample}';
\]

\[
\text{net}.\text{divideParam}.\text{trainRatio} = 70/100;
\]

\[
\text{net}.\text{divideParam}.\text{valRatio} = 15/100;
\]

\[
\text{net}.\text{divideParam}.\text{testRatio} = 15/100;
\]

\[
\text{net}.\text{performFcn} = '\text{mse}';
\]

\[
\text{net}.\text{plotFcns} = \{ '\text{plotperform}', '\text{plottrainstate}', '\text{ploterrhist}', ...
\]

\[
'\text{plotregression}', '\text{plotfit}' \};
\]

\[
[\text{net}, \text{tr}] = \text{train} \left( \text{net}, x, t \right);
\]

\[
y = \text{net} \left( x \right);
\]

\[
e = \text{gsubtract} \left( t, y \right);
\]

\[
\text{performance} = \text{perform} \left( \text{net}, t, y \right);
\]

\[
\text{trainTargets} = t .* \text{tr}.\text{trainMask} \{ 1 \};
\]

\[
\text{valTargets} = t .* \text{tr}.\text{valMask} \{ 1 \};
\]

\[
\text{testTargets} = t .* \text{tr}.\text{testMask} \{ 1 \};
\]

\[
\text{trainPerformance} = \text{perform} \left( \text{net}, \text{trainTargets}, y \right);
\]

\[
\text{valPerformance} = \text{perform} \left( \text{net}, \text{valTargets}, y \right);
\]

\[
\text{testPerformance} = \text{perform} \left( \text{net}, \text{testTargets}, y \right);
\]

\[
\text{view} \left( \text{net} \right);
\]

\[
\text{figure, plotperform(tr)}
\]

\[
\text{figure, plottrainstate(tr)}
\]

\[
\text{figure, ploterrhist(e)}
\]

\[
\text{figure, plotregression(t, y)}
\]

\[
\text{figure, plotfit(net, x, t)}
\]
### Appendix E: Neural Network Performance Table (Ethane)

#### One layer performances:

<table>
<thead>
<tr>
<th>No. of neurons in 1st layer</th>
<th>No. of neurons in 2nd layer</th>
<th>Performance (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>7.95E-06</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1.28E-05</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1.15E-05</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>1.05E-05</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>1.12E-05</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>8.82E-06</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>1.91E-05</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>1.30E-05</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>9.96E-05</td>
</tr>
<tr>
<td>14</td>
<td>0</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>15</td>
<td>0</td>
<td>1.01E-05</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1.41E-05</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>1.01E-05</td>
</tr>
</tbody>
</table>

#### Two layer performances:

<table>
<thead>
<tr>
<th>No. of neurons in 1st layer</th>
<th>No. of neurons in 2nd layer</th>
<th>Performance (MSE)</th>
<th>No. of neurons in 1st layer</th>
<th>No. of neurons in 2nd layer</th>
<th>Performance (MSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>1.36E-05</td>
<td>5</td>
<td>8</td>
<td>9.51E-06</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>2.98E-05</td>
<td>5</td>
<td>9</td>
<td>9.54E-06</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1.60E-05</td>
<td>5</td>
<td>10</td>
<td>1.01E-05</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1.26E-05</td>
<td>5</td>
<td>11</td>
<td>1.90E-05</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1.14E-05</td>
<td>5</td>
<td>12</td>
<td>8.88E-06</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>9.99E-06</td>
<td>6</td>
<td>1</td>
<td>9.09E-06</td>
</tr>
<tr>
<td>in 1st layer</td>
<td>in 2nd layer</td>
<td>(MSE)</td>
<td>1st layer</td>
<td>2nd layer</td>
<td>(MSE)</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>----------</td>
<td>-----------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>1.32E-05</td>
<td>8</td>
<td>5</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>1.33E-05</td>
<td>8</td>
<td>6</td>
<td>9.40E-06</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>1.20E-05</td>
<td>8</td>
<td>7</td>
<td>7.50E-06</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>8.98E-06</td>
<td>8</td>
<td>8</td>
<td>9.62E-06</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>1.10E-05</td>
<td>8</td>
<td>9</td>
<td>1.59E-05</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>1.11E-05</td>
<td>8</td>
<td>10</td>
<td>1.28E-05</td>
</tr>
<tr>
<td>6</td>
<td>9</td>
<td>1.04E-05</td>
<td>8</td>
<td>11</td>
<td>1.84E-05</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>9.76E-06</td>
<td>8</td>
<td>12</td>
<td>1.16E-05</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>9.61E-06</td>
<td>9</td>
<td>1</td>
<td>1.29E-05</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>1.07E-05</td>
<td>9</td>
<td>2</td>
<td>1.03E-05</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>1.06E-05</td>
<td>9</td>
<td>3</td>
<td>1.06E-05</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>9.62E-06</td>
<td>9</td>
<td>4</td>
<td>1.15E-05</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1.08E-05</td>
<td>9</td>
<td>5</td>
<td>1.76E-05</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8.73E-06</td>
<td>9</td>
<td>6</td>
<td>1.35E-05</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>1.05E-05</td>
<td>9</td>
<td>7</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>9.42E-06</td>
<td>9</td>
<td>8</td>
<td>1.32E-05</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>9.98E-06</td>
<td>9</td>
<td>9</td>
<td>8.91E-06</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9.83E-06</td>
<td>9</td>
<td>10</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>7</td>
<td>9</td>
<td>5.26E-05</td>
<td>9</td>
<td>11</td>
<td>2.28E-05</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>8.28E-06</td>
<td>9</td>
<td>12</td>
<td>1.44E-05</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>9.88E-06</td>
<td>10</td>
<td>1</td>
<td>9.42E-06</td>
</tr>
<tr>
<td>7</td>
<td>12</td>
<td>1.07E-05</td>
<td>10</td>
<td>2</td>
<td>1.15E-05</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>2.79E-05</td>
<td>10</td>
<td>3</td>
<td>1.07E-05</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1.12E-05</td>
<td>10</td>
<td>4</td>
<td>4.32E-05</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>9.30E-06</td>
<td>10</td>
<td>5</td>
<td>4.62E-05</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1.47E-05</td>
<td>10</td>
<td>6</td>
<td>1.25E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>10</td>
<td>7</td>
<td>9.96E-06</td>
<td>12</td>
<td>9</td>
<td>9.96E-06</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>1.05E-05</td>
<td>12</td>
<td>10</td>
<td>1.05E-05</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>9.55E-06</td>
<td>12</td>
<td>11</td>
<td>1.27E-05</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>1.06E-05</td>
<td>12</td>
<td>12</td>
<td>1.27E-05</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>1.41E-05</td>
<td>13</td>
<td>1</td>
<td>1.02E-05</td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>1.01E-05</td>
<td>13</td>
<td>2</td>
<td>1.13E-05</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1.06E-05</td>
<td>13</td>
<td>3</td>
<td>1.14E-05</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>8.95E-06</td>
<td>13</td>
<td>4</td>
<td>9.37E-06</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>1.12E-05</td>
<td>13</td>
<td>5</td>
<td>1.35E-05</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>1.07E-05</td>
<td>13</td>
<td>6</td>
<td>9.45E-06</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>9.06E-06</td>
<td>13</td>
<td>7</td>
<td>1.09E-05</td>
</tr>
<tr>
<td>11</td>
<td>6</td>
<td>1.05E-05</td>
<td>13</td>
<td>8</td>
<td>1.08E-05</td>
</tr>
<tr>
<td>11</td>
<td>7</td>
<td>3.45E-05</td>
<td>13</td>
<td>9</td>
<td>1.17E-05</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>1.31E-05</td>
<td>13</td>
<td>10</td>
<td>1.03E-05</td>
</tr>
<tr>
<td>11</td>
<td>9</td>
<td>1.42E-05</td>
<td>13</td>
<td>11</td>
<td>1.66E-05</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>1.01E-05</td>
<td>13</td>
<td>12</td>
<td>1.26E-05</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>1.87E-05</td>
<td>14</td>
<td>1</td>
<td>1.47E-05</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>1.32E-05</td>
<td>14</td>
<td>2</td>
<td>1.10E-05</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>9.49E-06</td>
<td>14</td>
<td>3</td>
<td>1.45E-05</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>1.50E-05</td>
<td>14</td>
<td>4</td>
<td>1.11E-05</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>1.30E-05</td>
<td>14</td>
<td>5</td>
<td>2.19E-05</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>1.35E-05</td>
<td>14</td>
<td>6</td>
<td>8.79E-06</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>1.38E-05</td>
<td>14</td>
<td>7</td>
<td>1.16E-05</td>
</tr>
<tr>
<td>12</td>
<td>6</td>
<td>1.04E-05</td>
<td>14</td>
<td>8</td>
<td>1.23E-05</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>1.11E-05</td>
<td>14</td>
<td>9</td>
<td>9.57E-06</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>1.04E-05</td>
<td>14</td>
<td>10</td>
<td>1.04E-05</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>9.03E-06</td>
<td>14</td>
<td>12</td>
<td>1.00E-05</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1.12E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1.03E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1.03E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>1.29E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>5</td>
<td>1.00E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>6</td>
<td>1.10E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>9.07E-06</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>1.03E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>1.21E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>1.24E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>11</td>
<td>1.36E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>12</td>
<td>1.05E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1.12E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1.03E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1.03E-05</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix F: Neural Network Predictor Function (Ethane)

```matlab
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)

% ====== NEURAL NETWORK CONSTANTS ======

% Input 1
x1_step1.xoffset = [1059.82;3.801;0.2229;303.15;0.0001];
x1_step1.gain = [0.011082784295689;3.11526479750779;20.8986415882968;0.05;1.96540880503145];
x1_step1.ymin = -1;

% Layer 1
b1 = [1.7245921631495611; -0.27263556316529602; 0.2495775281713364; -0.063636402574045958; 1.8683656006497706];
IW1_1 = [-1.2565512899668729 1.5260133893298713 -0.20158022149300248 -0.13569311185302479 -0.52486990791137667; -2.1147063189730169 -0.4382512585979208 1.6845091101921541 1.3706282911742651 -2.5256909810854138; 1.7107271050066624 0.2235216325504952 -1.2479402827660766 -0.70305367753942627 1.8222165292968531; -0.32229476568987747 -0.68517450100274113 -1.6286969789728816 0.1853408039764175 -0.39190430127652615; 0.17730475948235508 -0.31193840525563665 -0.54790442391335326 1.2458655355969617 1.0657482103378435];

% Layer 2
b2 = -0.32529140095626291;
LW2_1 = [-0.18532562344690715 1.2009347325939426 2.340872659695151 -2.1777919226065379 0.43988542691514748];

% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 42.1940928270042;
y1_step1.xoffset = 0;

% ===== SIMULATION ========

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
```

118
Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end

% ===== MODULE FUNCTIONS ========
% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
    y = bsxfun(@minus,x,settings.xoffset);
    y = bsxfun(@times,y,settings.gain);
    y = bsxfun(@plus,y,settings.ymin);
end
% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end