

**PREDICTION OF CRUDE OIL PVT PROPERTIES**

**BY SOFT COMPUTING TECHNIQUES**

BY

**Munirudeen Ajadi Oloso**

A Thesis Presented to the  
DEANSHIP OF GRADUATE STUDIES

**KING FAHD UNIVERSITY OF PETROLEUM & MINERALS**

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the  
Requirements for the Degree of

**MASTER OF SCIENCE**

In

**Systems Engineering**

**June 2009**

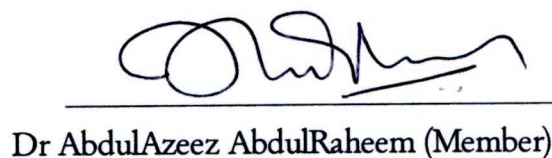
**KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS**  
**DHAHRAN 31261, SAUDI ARABIA**  
**DEANSHIP OF GRADUATE STUDIES**

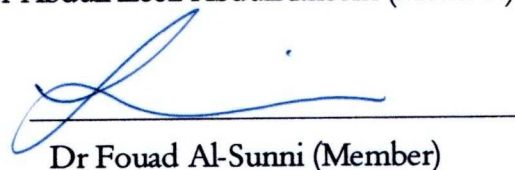
This thesis, written by Munirudeen Ajadi Oloso under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE IN SYSTEMS ENGINEERING.

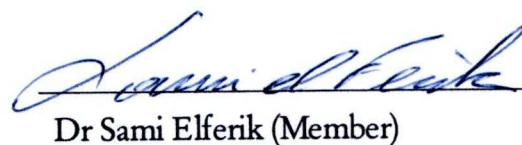
Thesis Committee

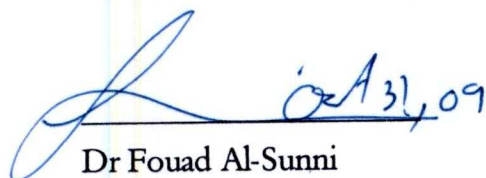
  
Dr. Amar Khoukhi (Advisor)

  
Dr. Moustafa Elshafei (Co-Advisor)

  
Dr AbdulAzeez AbdulRaheem (Member)

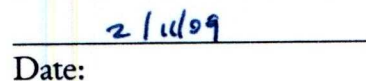
  
Dr Fouad Al-Sunni (Member)

  
Dr Sami Elferik (Member)

  
Dr Fouad Al-Sunni

Department Chairman

  
Dr. Salam Zummo  
Dean of Graduate Studies

  
Date: 2/11/09



**DEDICATION**

**DEDICATED  
TO ALLAH, THE MOST GRACIOUS**

## ACKNOWLEDGEMENT

All praises and adoration are due to Allah, to Him belongs the sovereignty of the heavens and the earth. May His blessings and mercies be upon the noblest of mankind, Muhammad (S.A.W.), his household, his companions and the generality of the true believers till the day of reckoning. I am grateful to Allah for all His favours on me since my birth, these blessings are indeed innumerable, the greatest of His bounties on me is being a Muslim.

Another great blessing of Allah on me is my father, Sheik Sirajudeen and my mother, Dhikrat. They have not only inculcated in me His teachings but they also practise them in their lives. I am more than grateful for the knowledge you imparted in me and the exceptional upbringing you have given me. Indeed, you have instilled the love for Allah (S.W.T.) and His Prophet (S.A.W.) in me. Also, I appreciate the supports of all my siblings especially my elder brothers; Dr Hamid (Computational Scientist and the Director of AMIT, NASA, Maryland), Dr Nurudeen and Mr Saheed (Senior Facilities Engineer, SHELL). I am really indebted to the entire Olosos. I appreciate you all. May Allah grant us all the companionship of His Prophet in paradise-ameen.

My warmest appreciation goes to my thesis advisor, Dr Amar Khoukhi and co-advisor, Prof. Moustafa Elshafei for their intellectual guidance, support and readiness to assist during this research. In the same vein, I appreciate the helpful suggestions of other thesis committee members: Dr Abdulazeez Abduraheem, Prof. Fouad M. Al-Sunni and Dr Sami El-Ferik. Special appreciation goes to Prof. Al-Marhoun M. for releasing the data which was used for this study.

In the same vein, I appreciate all my friends in Nigeria, at KFUPM and in the Kingdom at large. You have all made my stay in this Kingdom a memorable one. I cannot but mention specifically Mr Ismail Adebisi of SAUDI ARAMCO for his brotherly advice and supports. You sharpen my will every now and then. Truly, you love for your brother what you love for yourself.

Lastly, I thank Allah for blessing me with the joy of my heart. You came into my life when I was in dire need of a partner to share my “heart” with. Indeed, I have chosen you solely for your “deen”. Qudrah, “*UhibbukifILLAH*”. May Allah bless our union in this life and grant it continuity in the life to come. Amin.

## TABLE OF CONTENTS

DEDICATION.....	iii
ACKNOWLEDGEMENT .....	iv
TABLE OF CONTENTS.....	vi
LIST OF TABLES.....	ix
LIST OF FIGURES .....	xi
THESIS ABSTRACT .....	xv
CHAPTER ONE .....	1
INTRODUCTION .....	1
1.1. Overview .....	1
1.2. Problem Statement.....	3
1.3. Thesis Objectives .....	4
1.4. Scope of the Thesis.....	5
1.5. Thesis Organization.....	5
CHAPTER TWO .....	7
LITERATURE REVIEW .....	7
2.1. Introduction .....	7
2.2. Empirical Models for Predicting PVT Properties and Their Evaluations .....	8
2.3 Artificial Intelligence/Soft Computing Techniques for Predicting PVT Properties.....	14
CHAPTER THREE .....	21
SOFT COMPUTING TECHNIQUES .....	21
3.1. Artificial Neural Networks (ANN).....	21
3.1.1. Benefits of Neural Networks .....	21
3.1.2. Model of a Neuron.....	24
3.1.3. ANN Architectures.....	26
3.1.4. Learning Algorithm for Neural Networks .....	30
3.1.5. Drawbacks of Artificial Neural Network .....	34
3.2. Overview of Support Vector Machines (SVM).....	35
3.2.1. Support Vector Regression (SVR) .....	36
3.2.2. Description of SVR Parameters.....	39
3.3. Functional Networks .....	40
3.3.1. Background and Definition of Functional Networks .....	40
3.3.2. Differences between Functional and Neural Networks .....	41

3.3.3. Methods of Selecting Functional Network Models .....	43
3.3.4. Development of Functional Network Model.....	46
3.4. Adaptive Network Fuzzy Inference System (ANFIS).....	49
3.4.1. Development of Fuzzy Inference System.....	49
3.4.2 Learning Algorithm for ANFIS.....	50
3.5. Genetic Algorithm.....	53
3.5.1. General Overview of Genetic Algorithm .....	53
3.5.2. Steps for Implementing a Genetic Algorithm.....	56
CHAPTER FOUR.....	58
DATA ACQUISITION, APPROACH AND IMPLEMENTATION .....	58
4.1. Introduction .....	58
4.2. PVT Data Acquisition and Processing .....	59
4.3. Approach and Problem Formulation .....	61
4.3.1. Viscosity Curve Prediction.....	61
4.3.2. Gas/Oil Ratio Curve Prediction.....	64
4.4. Implementation of the Soft Computing Techniques.....	66
4.4.1. ANN and DE+ANN Implementation .....	67
4.4.2. SVR and FN Implementation .....	68
4.4.3. ANFIS and GA+ANFIS Implementation .....	72
CHAPTER FIVE .....	75
EXPERIMENTAL RESULTS AND DISCUSSION .....	75
5.1. Introduction .....	75
5.2. Results and Discussion for Viscosity Curve Prediction .....	76
5.2.1. ANN and DE+ANN for Viscosity Curve Prediction.....	76
5.2.2. SVR and FN for Viscosity Curve Prediction.....	81
5.2.3. ANFIS and GA+ANFIS for Viscosity Curve Prediction .....	86
5.2.4. Performances of All the Techniques for Viscosity Curve Prediction.....	91
5.3. Results and Discussion for Gas/Oil Ratio Curve Prediction .....	93
5.3.1. ANN and DE+ANN for Gas/Oil Ratio Curve Prediction.....	94
5.3.2. SVR and FN for Gas/Oil Ratio Curve Prediction.....	99
5.3.3. ANFIS and GA+ANFIS for Gas/Oil Ratio Curve Prediction .....	104
5.3.4. Performance of All the Techniques for Gas/Oil Ratio Curve Prediction .....	109

CHAPTER SIX.....	112
CONCLUSION.....	112
6.1 Summary .....	112
6.2. Contribution to Knowledge .....	113
6.3. Recommendations .....	114
Nomenclature.....	115
REFERENCES .....	116



## LIST OF TABLES

Table 4.1: Description of Data Set A.....	60
Table 4.2 Distribution of the Fitting Coefficients.....	66
Table 5.1: Statistical Performance Measures of ANN and DE+ANN Models for Viscosity Curve Prediction .....	77
Table 5.2: Sample Predicted Viscosity Curve Parameters by ANN and DE+ANN Models .....	77
Table 5.3: Statistical Performance Measures of SVR and FN Models for Viscosity Curve Prediction .....	81
Table 5.4: Sample Predicted Viscosity Curve Parameters by SVR and FN Models.....	82
Table 5.5: Statistical Performance Measures of ANFIS and GA+ANFIS Models for Viscosity Curve Prediction .....	87
Table 5.6: Sample Predicted Viscosity Curve Parameters by ANFIS and GA+ANFIS Models.....	87
Table 5.7: Time Complexity of All Models for Viscosity Curve Prediction.....	93
Table 5.8: Statistical Performance Measures of ANN and DE+ANN Models for Gas/Oil Ratio Curve Prediction.....	95
Table 5.9: Sample Predicted Gas/Oil Ratio Curve Parameters by ANN and DE+ANN Models.....	95
Table 5.10: Statistical Performance Measures of SVR and FN Models for Gas/Oil Ratio Curve Prediction .....	100
Table 5.11: Sample Predicted Gas/Oil Ratio Curve Parameters by SVR and FN Models .....	100

Table 5.12: Statistical Performance Measures of ANFIS and GA+ANFIS Models for Gas/Oil Ratio Curve Prediction .....	105
Table 5.13: Sample Predicted Gas/Oil Ratio Curve Parameters by ANFIS and GA+ANFIS Models .....	105
Table 5.14: Time Complexity of All Models for Gas/Oil Ratio Curve Prediction.....	111

## LIST OF FIGURES

Figure 3.1: Nonlinear Model of a Neuron .....	24
Figure 3.2: Single Layer Feedforward Network.....	27
Figure 3.3: Fully Connected Multilayer Feedforward Network.....	28
Figure 3.4: Recurrent Network with no Self-feedback Loops and no Hidden Neurons...	29
Figure 3.5: Recurrent Network with Hidden Neurons.....	29
Figure 3.6: Creation of margins between two data sets by support vectors .....	35
Figure 3.7: SVR maps input vectors to a higher dimensional space .....	36
Figure 3.8: The error function.....	37
Figure 3.9: The Support Vector Regression Tube .....	37
Figure 3.10: A Standard Neural Network.....	43
Figure 3.11: A Standard Functional Network.....	43
Figure 3.12: Fuzzy Reasoning and Its Equivalent ANFIS.....	51
Figure 3.13: Pseudo-code for a Simple Genetic Algorithm.....	56
Figure 4.1: A Typical Viscosity Curve .....	62
Figure 4.2: A fitted Viscosity Curve.....	63
Figure 4.3: A Typical Gas/Oil Ratio Curve .....	65
Figure 4.4: A fitted Gas/Oil Ratio Curve.....	65
Figure 5.1: Viscosity vs Pressure Plot for Sample Well TR1 .....	78
Figure 5.2: Viscosity vs Pressure Plot for Sample Well TR2.....	78
Figure 5.3: Viscosity vs Pressure Plot for Sample Well TR3.....	79
Figure 5.4: Viscosity vs Pressure Plot for Sample Well TS1 .....	79
Figure 5.5: Viscosity vs Pressure Plot for Sample Well TS2 .....	80

Figure 5.6: Viscosity vs Pressure Plot for Sample Well TS3 .....	80
Figure 5.7: Viscosity vs Pressure Plot for Sample Well TR1 .....	83
Figure 5.8: Viscosity vs Pressure Plot for Sample Well TR2 .....	83
Figure 5.9: Viscosity vs Pressure Plot for Sample Well TR3 .....	84
Figure 5.10: Viscosity vs Pressure Plot for Sample Well TS1 .....	84
Figure 5.11: Viscosity vs Pressure Plot for Sample Well TS2 .....	85
Figure 5.12: Viscosity vs Pressure Plot for Sample Well TS3 .....	85
Figure 5.13: Viscosity vs Pressure Plot for Sample Well TR1 .....	88
Figure 5.14: Viscosity vs Pressure Plot for Sample Well TR2 .....	88
Figure 5.15: Viscosity vs Pressure Plot for Sample Well TR3 .....	89
Figure 5.16: Viscosity vs Pressure Plot for Sample Well TS1 .....	89
Figure 5.17: Viscosity vs Pressure Plot for Sample Well TS2 .....	90
Figure 5.18: Viscosity vs Pressure Plot for Sample Well TS3 .....	90
Figure 5.19: The Root Mean Square Error of all the Models for Viscosity Curve Prediction (Training) .....	91
Figure 5.20: The Root Mean Square Error of all the Models for Viscosity Curve Prediction (Testing) .....	92
Figure 5.21: The Average Absolute Percent Relative Error of all the Models for Viscosity Curve Prediction (Training) .....	92
Figure 5.22: The Average Absolute Percent Relative Error of all the Models for Viscosity Curve Prediction (Testing) .....	93
Figure 5.23: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1 .....	96
Figure 5.24: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2 .....	96

Figure 5.25: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3 .....	97
Figure 5.26: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1 .....	97
Figure 5.27: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2 .....	98
Figure 5.28: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3 .....	98
Figure 5.29: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1 .....	101
Figure 5.30: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2 .....	101
Figure 5.31: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3 .....	102
Figure 5.32: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1 .....	102
Figure 5.33: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2 .....	103
Figure 5.34: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3 .....	103
Figure 5.35: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1 .....	106
Figure 5.36: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2 .....	106
Figure 5.37: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3 .....	107
Figure 5.38: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1 .....	107
Figure 5.39: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2 .....	108
Figure 5.40: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3 .....	108
Figure 5.41: The Root Mean Square Error of all the Models for Gas/Oil ratio Curve Prediction (Training).....	109
Figure 5.42: The Root Mean Square of all the Models for Gas/Oil ratio Prediction (Testing).....	110
Figure 5.43: The Average Absolute Percent Relative Error of all the Models for Gas/Oil ratio Curve Prediction (Training).....	110

Figure 5.44: The Average Absolute Percent Relative Error of all the Models for Gas/Oil  
ratio Curve Prediction (Testing) ..... 111

## THESIS ABSTRACT

**NAME:** Munirudeen Ajadi Oloso

**TITLE OF STUDY:** Prediction of Crude Oil PVT Properties  
by Soft Computing Techniques

**MAJOR FIELD:** Systems Engineering

**DATE OF DEGREE:** June, 2009

Characterization of Pressure-Volume-Temperature (PVT) properties of crude oil is important for many types of petroleum calculations, such as, determination of hydrocarbon flowing properties, gas-lift and pipeline design, calculation of oil recovery both from natural depletion and recovery techniques. Two of these important properties are the oil viscosity and gas/oil ratio. An experimental analysis which is both time-consuming and costly is used to determine these properties over the entire range of pressures.

To solve the problem of going through these rigorous laboratory experimentations which gulp valuable production resources, time and money, equations of states (EOS) and empirically derived correlations have been used to predict these reservoir fluid properties. These two methods were used for a long time until Soft Computing (SC) /Artificial Intelligence (AI) techniques, basically Neural Networks, were introduced to improve the prediction performances. However, all the prediction methods up to date are for predicting single or multi-data points, even for PVT properties that are generated as curves

In this study, we have developed a new approach for predicting PVT properties that need to be described by curves over specific ranges of reservoir pressures. This approach is demonstrated with oil viscosity and gas/oil ratio curves. First, a thorough study of the target reservoir properties based on the data collected from PVT laboratory analyses of crude oil were carried out. Also, a statistical analysis was conducted on the data to detect the outliers. We then explored the capabilities of different Soft Computing techniques for predicting these properties. Different prediction models using Support Vector Regression (SVR), Functional Networks (FN), Adaptive Neuro-Fuzzy Inference

Systems (ANFIS) and Artificial Neural Networks (ANN) and also two hybrid models: Differential Evolution Algorithm with ANN (DE+ANN) and Genetic Algorithm with ANFIS (GA+ANFIS) have been developed. A very small root mean square error and absolute average percent error for the developed models were recorded.

Any PVT property which can be described as a curve can easily be estimated using the outlined approach in this work. Therefore, this work will hopefully be a very fast and low cost method for predicting PVT properties for optimizing the oil production operation.



## خلاصة الرسالة

**الاسم:** منير الدين أجادي أولوسو  
**عنوان البحث:** التنبؤ بخصائص الضغط والحجم والحرارة للنفط الخام باستخدام التقنيات البرمجية  
**التخصص:** هندسة النظم  
**تاريخ الرسالة:** يونيو 2009

إن توصيف خصائص الضغط والحجم والحرارة للنفط الخام أمر مهم جدا للعديد من الحسابات النفطية كتحديد خصائص الهيدروكربونات المائية ورفع الغاز وتصميم الأنابيب وحسابات استرداد المخازن بالاستنفاد الطبيعي وتقنيات الانتعاش. ومن هذه الخواص الهامة خاصيتنا اللزوجة ونسبة الغاز إلى النفط. ويتم تحديد هذه الخواص عن طريق التحاليل التجريبية لمختلف قيم الضغط، مما يترتب عليه إنفاق للوقت والمال. وللاستعاضة عن هذه التحاليل المخبرية التجريبية المضنية، والمستنفزة للمصادر والمال والزمن، فقد تم اشتقاق معادلات الأحوال EOS تجريبياً للتنبؤ بخواص المخازن.

ولقد استمر العمل على هاتين الطريقتين ردحا من الزمن حتى بزغت الطرق البرمجية والمعتمدة على الذكاء الصناعي، وخاصة الخلايا العصبية، لتحسين أداء المتنبئات. لكن هذه الأساليب كلها حتى وقتنا الحاضر لا تزال تعتمد على التنبؤ بالنقطة الواحدة أو بالنقط العديدة، حتى وإن احتيج لمنحنيات كاملة لتمثيل بعض الخواص كالضغط والحجم والحرارة.

قمنا في هذه الدراسة بتطوير طريقة جديدة للتنبؤ بخصائص الضغط والحجم والحرارة التي تحتاج في تمثيلها للمنحنيات التي تنتمي قيم المدخلات فيها لمدى معين، مع تبيان ذلك بمثالي اللزوجة والنسبة بين الغاز والنفط. وقد تمت في البداية دراسة عميقة لخصائص المخازن المستهدفة بالاستعانة بنتائج التحاليل المخبرية للضغط والحجم والحرارة. كما أخضعت البيانات لبعض التحاليل الإحصائية لتحديد المتطرفات. ثم تم استكشاف إمكانات التقنيات البرمجية المختلفة للتنبؤ بهذه الخواص. فتم تطوير العديد من نماذج التنبؤ باستخدام انحدار المتجهات الداعمة (SVR) والشبكات الوظيفية والأنظمة العصبية الضبابية المتكيفة والشبكات العصبية الصناعية بالإضافة لنظامين هجينين يعتمدان على خوارزمية التطور التفاضلية مع الشبكات العصبية، والخوارزميات الجينية مع الأنظمة العصبية الضبابية المتكيفة. وكانت النتيجة الحصول على نسبة خطأ صغيرة بمعياري جذر معدل المربعات ومعدل القيم المطلقة.

هذا، ونشير إلى أن الطريقة المتبعة في هذا البحث تصلح لأي خاصية من خصائص الضغط والحجم والحرارة ذات القيم المتراوحة على مدى معين من الضغط. وعليه، فإننا نرجو أن يمثل هذا العمل طريقة سريعة ورخيصة للتنبؤ بخصائص الضغط والحجم والحرارة للوصول لأكبر انتاجية نفطية ممكنة.

# CHAPTER ONE

## INTRODUCTION

### 1.1. Overview

In petroleum engineering, characterization of reservoir fluids plays an important role in developing strategies for operating and managing existing reservoirs and development of new ones. These reservoir fluid properties are important for petroleum engineering computations to determine: the amount of oil or gas present in reservoirs; the amount that can be recovered (reserve); the flow rate of oil or gas; the forecast of future production and the design parameters for production facilities.

Traditionally, these properties are determined from laboratory studies on samples collected from the bottom of the wellbore or after recombining the liquid and vapour samples collected from the separator at the surface. Such experimental data are, however, not always available or very expensive to obtain. Also, the experimental analysis that is used to determine these properties takes a lot of time and a very high expertise is required for it.

There are basically two laboratory methods for determining Pressure-Volume-Temperature (PVT) properties; flash test and differential liberation test, Dake [16]. In the former test, the pressure in the Pressure-Volume (PV) cell is initially raised to a value far in excess of the bubble point. The pressure is subsequently reduced in stages, and on each occasion the total volume of the cell contents is recorded. As soon as the bubble point pressure is reached, gas is liberated from the oil and the overall compressibility of the system increases significantly. Thereafter, small changes in pressure will result in large changes in the total fluid volume contained in the PV cell. In this manner, the flash

expansion experiment can be used to "feel" the bubble point. Since the cell used is usually opaque, the separate volumes of oil and gas below bubble point pressure cannot be measured in the experiment and therefore, only total fluid volumes are recorded. In the laboratory analysis, the basic unit of volume against which all others are compared, is the volume of saturated oil at the bubble point, irrespective of its magnitude. This test is essential for determination of gas/oil ratio.

The second test, differential liberation test is used to generate viscosity curve. Prediction of These two important curves is the focus of this work. In differential liberation test, the associated and free gases are removed at each stage of separation as the pressure on the oil is reduced. The liberated gas is composed mainly of lighter components. When the gas is separated in this manner, a large amount of heavy and intermediate components will remain in the liquid and there will be minimal oil shrinkage in the stock tank, hence, resulting in greater oil recovery. To recover the gas fractions produced in the separators operating at medium and low pressure, it is necessary to re-compress them to the pressure of the high-pressure separator. In this case, all liquids collected in compressor suction tanks are recycled to the production unit. However, re-compression is sometimes considered to be too costly, hence, the usual gas flaring.

To solve the problem of going through these rigorous laboratory experimentations which consume valuable production resources: time and money, empirically derived correlations and Equations of States (EOS) have been used to predict these reservoir fluid properties. These two methods were used for a long period of time until Artificial Intelligence (AI)/Soft Computing (SC) Techniques were implemented to improve the prediction performances. Of all the existing SC techniques, the most widely used in

Petroleum Engineering is Artificial Neural Networks (ANN), Mohagheh [52].

According to Zadeh, “Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, and partial truth”, Nikravesh et al [54]. No doubt these techniques provide the opportunity to achieve robust, tractable solution whilst, at the same time, offering low cost. Now, soft computing like evolutionary algorithms, machine reasoning, fuzzy logic, neural systems, etc., crowd the computational landscape and new techniques are being developed every day. Though these techniques have not been widely-utilised in Petroleum Engineering compared with some other fields of lives, they have been applied successfully in some Petroleum Engineering problems with exceptional and acceptable performance.

In this research work, we will utilise four of these soft computing techniques, namely: Feedforward Neural Network (FFNN), Adaptive Neuro-Fuzzy Inference Systems (ANFIS), Functional Networks (FN) and Support Vector Regression (SVR), and two hybrid frameworks: Differential Evolution Algorithm with ANN (DE+ANN) and Genetic Algorithm with ANFIS (GA+ANFIS), in a heuristic approach to predict viscosity and gas/oil ratio curves.

## **1.2. Problem Statement**

There is necessity to have accurate prediction of reservoir fluid and rock properties in petroleum engineering. At every stage of the petroleum exploration and production business, *a priori* knowledge of how the fluids will behave under a wide range of pressure and temperature conditions, particularly in terms of their volumetric and thermo physical properties, is always required. Hence, the need for the prediction of

PVT properties that is done through the use of equations of states (EOS). After EOS, empirical correlations which are now widely used were introduced. However, the EOS are derived for pure substances and hence, there is a need to always add correction factor(s) when used on practical data. Likewise, most of the existing correlations were developed using regional crude oils and their performance on crude oils from other regions are usually unacceptable.

To improve the accuracy of the predictions, some AI techniques have been applied. However, some of these properties that are in form of curves are predicted through single or multi- data points. Meanwhile, the usual single or multi-data point predictions could comprise the original shape of the curves. Hence, there is a need to predict the entire measurements that are described by curves for some of these properties. Two examples of such PVT properties are oil viscosity and gas/oil ratio. These two properties vary with pressure and there is always the need to have their values over a certain range of pressures.

### **1.3. Thesis Objectives**

This study introduces a new direction in prediction of PVT properties. In this study, instead of predicting single or multi-data points for a PVT property curve, the entire curve is estimated. Specifically, the following tasks were performed.

- A new approach has been formulated for predicting PVT properties that are generated as curves.
- All relevant physical laws have been taken into consideration and most importantly, the shapes of the curves are preserved in the formulation

- Computer models for predicting viscosity and gas-to-oil curves were developed using soft computing techniques for almost hundred samples from different crude oil wells.
- The soft computing techniques that have been exploited are: Artificial Neural Networks (ANN), Adaptive Neuro-Fuzzy Inference Systems (ANFIS), Support Vector Regression (SVR) and Functional Networks (FN), and two hybrid frameworks, DE-Optimized ANN (DE+ANN) and GA-Optimized ANFIS (GA+ANFIS).
- Graphical and Numerical comparisons vis-à-vis the accuracy of the four techniques in the prediction have also been presented.

#### **1.4. Scope of the Thesis**

In this study, a new approach to predict some PVT properties that are in curve form is proposed. We have limited ourselves to two of the properties based on the tasks involved and availability of data. For all required predictions in this work, we have implemented four independent soft computing techniques: ANN, FN, SVR, ANFIS and two hybrids frameworks: DE+ANN and GA+ANFIS.

#### **1.5. Thesis Organization**

The rest of the thesis is organized as follows. Chapter two gives a detailed literature survey in the area of PVT properties prediction. In chapter three, adequate information about the implemented independent soft computing techniques including the hybrids is presented. In chapter four, information about the data sets used for this study,

problem formulation approach and implementation of the SC techniques is given. Subsequently, we present the simulation results and a comparative study is carried out in chapter five. Lastly, we draw conclusions in chapter six and state the contributions that have been achieved in this study.

## CHAPTER TWO

### LITERATURE REVIEW

#### 2.1. Introduction

From the onset, relationships between pressure, volume and temperature of a fluid are expressed through the use of equations of state. An Equation of State (EOS) is used to define the state of the system and to determine the properties of the system at that state. It is a functional relationship between state variables — usually a complete set of such variables. Most EOS are written to express functional relationships between P, T and V. The most fundamental EOS, equation 2.1, is the combination of Boyle's and Charles' Laws to represent the PVT behaviour of an ideal gas [48]

$$PV = nRT \quad (2.1)$$

However, no gas behaves ideally in reality. Therefore, the ideal EOS is not useful for practical applications, although it is important as the basis for understanding of gas behaviour. These gases that do not behave ideally are referred to as real gases. In this regard, a correction factor is introduced to account for the discrepancies between experimental observations of real gases and predictions from the ideal model. Hence we have equation 2.2, representing PVT behaviour of a real gas.

$$PV = ZnRT \quad (2.2)$$

Where  $Z$  is the correction factor/ compressibility factor and  $Z=1$  for ideal gases.

These two equations are the foundations for all other modern EOS. The main setback of equations of state is that, they are developed for pure substances and their application to mixtures requires an additional variable. To improve on the estimation of PVT properties through EOS, correlations and soft computing techniques have been applied.



The most common empirical methods and the published artificial intelligence techniques that have been used in predicting PVT properties are reviewed in this chapter. In addition, the most common limitations of the popular techniques for predicting PVT properties especially the widely used empirical methods are discussed.

## **2.2. Empirical Models for Predicting PVT Properties and Their Evaluations**

Realizing the need and significance of predicting PVT properties, researchers have developed several empirical models to estimate these properties in the last six decades. Regression analysis is the widely used approach in developing these correlations. In this regard, several correlations have been developed especially for regional crude oils, and this is one of the problems associated with this conventional long time approach. A correlation that is developed for predicting a crude oil property of a particular region often fails to give a satisfactory performance when used on crude oils from another region. The main reason is the difference in crude oil compositions.

Katz [40] presented graphical methods for predicting the reservoir oil shrinkage and it is called the Katz empirical correlation. Oil shrinkage is the inverse of oil formation volume factor. The complexity in the use of Katz correlation lies in the need to combine graphical interpretations with calculations. Standing [63, 64, and 65] presented correlations for bubble point pressure ( $P_b$ ) and oil formation volume factor ( $B_o$ ). Standing's correlations were based on reports from laboratory experiments that were carried out on 105 oil samples from 22 different crude oils in California, U.S.A. These two correlations developed by Standing are functions of: solution gas/oil ratio ( $R_s$ ),

reservoir temperature, oil gravity and gas gravity. He was the first researcher to correlate ( $P_b$ ) and ( $B_o$ ) with those four parameters.

Vazquez and Beggs [70] presented correlations for  $R_s$ ,  $B_o$  for saturated and undersaturated oils (i.e. below and above bubble point pressures respectively), and also for oil viscosity above bubble point pressure (undersaturated oil viscosity).  $R_s$  was correlated as a function of oil gravity, gas gravity and temperature.  $B_o$  for saturated oil was correlated with  $R_s$ , temperature ( $T$ ), oil gravity and gas gravity. On the other hand,  $B_o$  for undersaturated oil was given as a function of oil compressibility, reservoir pressure ( $P$ ) and  $P_b$ . The oil compressibility was correlated as an intermediate result with  $R_s$ ,  $T$ , oil gravity, gas gravity and  $P$ . Lastly, the undersaturated viscosity was correlated with oil viscosity at bubble point ( $\mu_{ob}$ ), pressure  $P$  and  $P_b$ . To use Vazquez and Beggs correlations, gas gravity must be normalized to separator conditions of 100 psig.

Glaso [24] developed his own correlations for estimating bubble point pressure ( $P_b$ , saturation pressure), oil formation volume factor ( $B_o$ ) at  $P_b$  and total formation volume factor below  $P_b$ . These correlations were based on 45 oil samples from North Sea hydrocarbon mixtures and they were developed as a function of reservoir temperature, total surface gravity, producing  $R_s$  and stock-tank oil gravity. Glaso also developed a correlation for dead oil viscosity based on 26 crude oil samples. Given the usual variations of crude compositions, Glaso presented correction factors for the effect of non-hydrocarbons that could be present in crude oils. In essence, he aimed to solve the usual problem that is faced in using a correlation for crude oils that might have not been used in developing the correlation.

Al-Marhoun [1] published his correlations for estimating  $P_b$  and oil formation volume factor at  $P_b$  ( $B_{ob}$ ) for Middle East crude oils. Data sets that were used for his study were from PVT analyses of 69 bottomhole fluid samples from 69 Middle Eastern reservoirs. He used 160 experimentally obtained data points to develop each of the correlations. The correlations were developed as functions of  $R_s$ , total surface gas relative density, stock-tank oil relative density and reservoir temperature. He compared the accuracy of his correlations with those of Standing [63] and Glaso [24]. He reported lower average percentage absolute error (AAPRE) of 3.66% for  $P_b$  against 12.08% for Standing [63] and 25.22% for Glaso [24]. For  $B_{ob}$ , he reported AAPRE of 0.88% against 2.32% for Glaso [24]. The comparison was meant to establish that PVT correlations are actually regional sensitive since the properties of the crude oils vary. Labedi [45] presented new correlations for oil formation volume factor for African crude oils. He used 97, 28 and 4 data sets from Libya, Nigeria, and Angola respectively to develop his correlations.

Dokla and Osman [18] developed correlations for estimating bubble point pressure and  $B_{ob}$  for UAE crude oil using 51 data sets. In developing their correlations, they calculated new coefficients for Al-Marhoun's correlations [1]. The AAPREs of Dokla and Osman's correlations for  $P_b$  and  $B_{ob}$  were 7.61% and 1.225% respectively. They also performed the same comparison as in [1] by applying the correlations in [64], [24] and [1] on the UAE crude oil. Their correlations, as expected, gave the best results. However, Al-Yousef and Al-Marhoun [8] pointed out that Dokla and Osman's correlation [18] contradicts physical laws; the  $P_b$  decreases with temperature and it is

insensitive to changes in oil-gravity. Also, Al-Marhoun [2] presented new correlations for oil formation volume factor at bubble point pressure using 4012 experimentally obtained data points. The data set represented samples from all over the world, but mostly from Middle East and North America. The AAPRE for this Al-Marhoun's new correlation was 0.28%.

Petrosky and Farshad [60] developed correlations for predicting  $P_b$ ,  $B_{ob}$ ,  $R_s$  and undersaturated isothermal oil compressibility for Mexico crude oils. Their correlations for  $P_b$ ,  $B_{ob}$  and  $R_s$  were derived from Standing's correlations [63], while their undersaturated oil compressibility correlation is similar to that of Vazquez and Beggs [70]. A total of 81 laboratory PVT analyses were used to develop the correlations. Comparisons were made between the performance of their correlations and those in [64], [70], [24] and [1] when used for Mexico crude oils. The AAPREs for Petrosky and Farshad's correlations were 3.28%, 3.8%, 0.64% and 6.66% for bubble point pressure, solution gas/oil ratio, bubble point oil formation volume factor and undersaturated isothermal oil compressibility respectively.

Almehaideb [5] presented new correlations for UAE crude oils to estimate the formation volume factor at bubble point pressure, oil compressibility, bubble point pressure and bubble point oil viscosity and undersaturated oil viscosity. He used data sets from more than 15 reservoirs in UAE. He showed the need for regional correlations by comparing his results with those in [64], [70], [24], [1] and [18] among others. The AAPREs for Almehaideb's correlations were 1.35%, 9.88%, 4.997%, 13% and 2.885% for bubble point oil formation volume factor, oil compressibility, bubble point pressure, bubble point oil viscosity and undersaturated oil viscosity respectively. Hemmati and

Kharrat [33] presented correlations for estimating bubble point pressure, solution gas/oil ratio ( $R_s$ ), and oil formation volume factor at  $P_b$  for Iranian crude oils. The data that were used to develop the correlations covered a wide range of reservoirs with oil gravity of 18.8 to 48.34 API. They also demonstrated that their newly developed correlations performed better than any previous one in predicting PVT properties of Iranian crude oils. The AAPREs for Hemmati and Kharrat's correlations are 3.67%, 1.08% and 4.07% for  $P_b$ ,  $B_{ob}$  and  $R_s$  respectively.

Farshad et al [23] presented correlations to estimate bubble point pressure, solution gas oil ratio, oil formation volume factor and isothermal compressibility for Colombian crude oils. Separator conditions were taken into consideration in developing the correlations and a total of 98 reservoir fluid samples were used. A correlation was developed for each PVT property under consideration for different separator stages, and the results were evaluated using average percent relative error and error standard deviation. Velarde et al [71] also presented correlations for black oils. The correlations were developed for oil formation volume factors, bubble point pressure and gas/oil ratio with AAPREs of 1.74%, 11.5% and 4.73% respectively. They reported to have taken into consideration the material balance between their inputs and the reservoir oil density.

Petrosky and Farshad [59] presented new empirical correlations for dead oil, saturated and undersaturated oil viscosities with AAPREs of 12.4%, 14.5% and 2.9% respectively. They compared the performance of the correlations with some of the previous ones. Khan et al [43] presented three different correlations for viscosities (below, at and above) bubble point pressure with AAPREs of 5.157%, 12.148% and 1.915% respectively. For viscosity at bubble point pressure, the independent variables

were gas relative density, solution gas/oil ratio, relative temperature and relative density at the reference point. For viscosity above and below bubble point, the correlating variables were bubble point oil viscosity, pressure and bubble point pressure. Also,

Omar and Todd [55] developed correlations for bubble point pressure and bubble point oil formation volume factor with AAPREs of 7.17% and 1.44% respectively. A total of 93 PVT data sets from Malaysian crude oils were used. These new black oil correlations were based on those of Standing.

Sidqi and Al-Marhoun [62] developed a new PVT correlation for finding viscosity at bubble point pressure,  $\mu_b$  for Canadian and Middle Eastern crude oils. The correlation was developed as a function of  $B_{ob}$ ,  $\gamma_g$ ,  $R_s$  and oil relative density at bubble point pressure,  $\gamma_o$ . An AAPRE of 4.91% was reported with correlation of 0.997. Meanwhile, several authors have evaluated the accuracy and reliability of the existing correlations. What they do basically is to apply some commonly used correlations for predicting PVT Properties of their regional crude oils, and also improve on the correlations with the ultimate aim of improving the prediction accuracy.

Sunday et al [66] evaluated existing correlations for Niger Delta crude oils. They used a total of 237 PVT reports for their study. The analysis was done for a quite number of PVT properties using many of the existing correlations. Also, Ghetto et al [27], Elsharkawy et al [21], Mahmood and Al-Marhoun [47], Hemmati and Kharrat [34], Al-Marhoun [3], McCain et al [49], Hanafy [31] and Al-Shammasi [7], all evaluated some already developed correlations on their regional data to indicate the best one to characterise PVT for the used data. Normally, the correlations coefficients are re-calculated for the best representation to improve the prediction accuracy during an

evaluation study. Also, Ayoub et al [9] evaluated existing PVT correlations of viscosity below bubble point pressure for Pakistani crude oil.

### **2.3 Artificial Intelligence/Soft Computing Techniques for Predicting PVT Properties**

The first AI technique that was applied in prediction of PVT properties is Artificial Neural Networks (ANNs). ANNs are parallel-distributed information processing models that can recognize highly complex patterns within available data. In recent years, neural networks have gained popularity in petroleum applications. Many authors discussed the applications of neural networks in petroleum engineering [6, 44, 52 and 53]. Recently, it was shown in both machine learning and data mining communities that artificial neural networks have the capacity to learn complex linear/nonlinear relationships amongst input and output data. There are many different types of neural networks. The most widely used neural network in the literature is the feedforward neural networks with back propagation training algorithm.

This type of neural networks is a good computational intelligence modeling scheme in both prediction and classification tasks, though with some drawbacks that researchers have emphasized. Relatively, a quite number of studies have been carried out in the petroleum industry to model PVT properties using neural networks. Though it used not to be quite popular within the petroleum industry, it has gained awareness and opened ways for other AI techniques within the petroleum industry.

Gharbi and Elsharkawy [25] and Osman et al [56] carried out comparative studies between the performances of feedforward neural networks and the four empirical correlations in [2, 24, 60 and 64]. In [25], the authors published neural network models

for estimating bubble point pressure and oil formation volume factor for Middle East crude oils. They used a neural system with log sigmoid activation function and backpropagation with momentum for training. Two neural networks were trained separately to estimate the bubble point pressure ( $P_b$ ) and oil formation volume factor ( $B_o$ ), respectively. The input data were solution gas/oil ratio, reservoir temperature, oil gravity, and gas relative density. They used two hidden layers (2HL) neural networks: the first neural network, (4-8-4-1) to predict the bubble point pressure and the second neural network, (4-6-6-1) to predict the oil formation volume factor. Both neural networks were built using a data set of size 520 observations from Middle East region. The input data set was divided into a training set of 498 data points and a testing set of 22 data points. The AAPREs for  $P_b$  and  $B_o$  during ANN testing were 6.89% and 2.79% respectively and the ANN testing correlations were 0.962 and 0.979 respectively.

Osman et al [56] used the feedforward ANN to estimate the formation volume factor at the bubble point pressure. The neural network model was developed using 803 data which were gathered from Malaysia, Middle East, Gulf of Mexico, and Colombia. They designed one hidden layer (1HL) feedforward neural network (4-5-1) with the back propagation learning algorithm. The input layer has four neurons covering the input data of gas/oil ratio, API oil gravity, relative gas density, and reservoir temperature, one hidden layer with five neurons and single neuron for the formation volume factor in the output layer. The results of the developed neural network model outperformed most common empirical correlations techniques with testing absolute average error of 1.789%, and correlation coefficient of 0.988.



Al-Shammasi [7] presented neural network models for predicting PVT properties and compared the performances of the developed ANN models with those of empirical correlations. He concluded that statistical and trend performance analysis showed that some of the correlations violate the physical behaviours of hydrocarbon fluid properties. In addition, he pointed out that the published neural network models missed major model parameters to be reproduced. He used 2 hidden layers (2HL) neural networks, (4-5-3-1) structure for predicting bubble point pressure and oil formation volume factor. He evaluated published correlations and neural-network models for bubble point pressure,  $P_b$ , and oil formation volume factor ( $B_o$ ) for their accuracy and flexibility in representing hydrocarbon mixtures from different locations worldwide. The study presented a new and improved correlation for  $P_b$  based on global data. For his ANN models, the testing AAPREs were 19.86% and 11.68% for  $P_b$  and  $B_o$  respectively. On the other hand, his newly developed correlations gave APPREs of 17.8% and 1.806% for  $P_b$  and  $B_o$  respectively.

Varotsis et al [69] introduced a novel approach for predicting the complete PVT behavior of reservoir oils and gas condensates using two-hidden-layer neural networks. This network was trained by a PVT database of over 650 reservoir fluids originating from all parts of the world. It was reported that during testing of the ANN models, most of the PVT properties were estimated with a very low mean relative error of 0.5-2.5% and no one was in excess of 5%.

Al-Marhoun and Osman [4] developed two new ANN models to predict the bubble point pressure, and the oil formation volume factor at the bubble-point pressure for Saudi crude oils. The developed ANN architectures were 4-7-1 and 4-8-1 for  $P_b$  and

$B_{ob}$  respectively. The two networks were trained using backpropagation with sigmoid function. The models were developed using 283 data sets collected from different Saudi oil fields. Out of the 283 data sets, 142 were used to train the  $B_{ob}$  and  $P_b$  ANN models, 71 to cross-validate the relationships established during the training process and adjust the calculated weights, and the remaining 70 to test the model to evaluate its accuracy. The results showed that the developed  $B_{ob}$  model provides better predictions and higher accuracy than the published empirical correlations. The neural networks model predicted  $B_{ob}$  with an absolute average percent error of 0.5116%, standard deviation of 0.6626 and correlation coefficient of 0.9989. In addition, the developed ANN model for  $P_b$  outperformed the published empirical correlations. Prediction of  $P_b$  gave an absolute average percent error of 5.8915%, standard deviation of 8.6781 and correlation coefficient of 0.9965.

Goda et al [28] developed feedforward neural networks to estimate both bubble point pressure ( $P_b$ ) and oil formation volume factor ( $B_o$ ) through two linked feedforward neural networks from 180 data sets. For  $P_b$ , the inputs were: gas/oil ratio, API oil gravity, relative gas density, and reservoir temperature. These four inputs into the  $P_b$  ANN model along with the predicted  $P_b$  were used as inputs for  $B_o$  model. For  $P_b$ , a two hidden layers (2HL) neural network (4-10-10-1) was used and also, a two hidden layers ANN was used for  $B_o$  prediction with structure (5-8-8-1). The training algorithm that was adopted by these authors was the commonly used backpropagation with log sigmoid function. The testing correlations of the ANN models for  $P_b$  and  $B_o$  are 0.9981 and 0.9936, and the average absolute errors are 0.030704 and 0.00368 respectively.

Osman and Al-Marhoun [58] developed two new ANN models to predict different brine properties. The first model, using Radial Basis Function of architecture (3-38-3), predicted brine density, formation volume factor, and isothermal compressibility as a function of pressure, temperature and salinity. The second model, using backpropagation with network structure (2-2-1), was developed to predict brine viscosity as a function of temperature and salinity only. The models were developed using 1040 data sets. These data were divided into three groups: training, cross-validation and testing. Trend analysis was performed to ensure that the developed model followed the physical laws. The AAPREs of the developed correlations were 0.0981%, 1.0643%, 0.1305% and 1.908% for brine formation volume factor, compressibility, density and viscosity respectively.

Gharbi et al [26] presented an ANN model for predicting bubble point pressure and oil formation volume factor based on 5200 training and 234 testing data sets. The architecture of the ANN model was (4-5-1) using backpropagation with momentum training. The inputs into the network were gas/oil ratio, gas specific gravity, oil specific gravity and the reservoir temperature. For ANN testing data, the AAPREs for bubble point pressure and oil formation volume factor were 6.48% and 1.97%, and the correlations were 0.9891 and 0.9875 respectively.

Elsharkawy [22] presented Radial Basis Function Neural Network (RBFN) models to predict PVT properties of crude oil and natural gas. Two models were developed to predict solution gas/oil ratio, oil formation volume factor, oil viscosity, oil density, undersaturated oil compressibility and gas gravity with AAPREs of 4.53%, 0.53%, 8.72%, 0.4%, 5.98% and 3.03% respectively. The first RBF (4-100-100-4) model

was used to predict the first four listed PVT properties while the second model (with 4-100-100-1 structure) was used to predict the last two properties, undersaturated oil compressibility and gas gravity. He used differential PVT data of ninety samples for training and another ten samples for testing the model. As stated by the author, input data to the RBFN models included reservoir pressure, temperature, stock tank oil gravity, and separator gas gravity.

Osman and Abdel-Aal [57] introduced the Abductive Network as an alternative modeling tool to predict both bubble point pressure ( $P_b$ ) and bubble point oil formation volume factor ( $B_{ob}$ ) for Saudi crude oils using 283 PVT observations. Out of 283 data sets, 198 data points were used for training and 85 for testing. Unlike neural network, the abductive network uses various types of more powerful polynomial functional elements based on prediction performance, which is based on the self-organizing group method of data handling (GMDH). The correlation and AAPRE for  $P_b$  prediction were 0.9898 and 5.62% respectively, and for  $B_{ob}$ , they were 0.9959 and 0.86% respectively. Unlike what could be found in some other research works, the authors predicted  $B_{ob}$  as a function of only reservoir temperature and gas/oil.

Ayoub et al [9] constructed a neural network model to predict the viscosity below bubble point pressure for Pakistani crude oil in addition to the evaluation of the existing correlations on the crude oil. The correlating parameters were: pressure, reservoir temperature, bubble point pressure, oil FVF, solution GOR, gas specific gravity and API gravity. The neural network correlation outperformed all the evaluated empirical correlations with testing correlation coefficient of 99.3% and AAPRE of 1.171%. An ANN of (7-8-8-1) structure with backpropagation training was used by the authors.

El-Sebakhy et al [20] developed a Support Vector Machines framework to predict bubble point pressure ( $P_b$ ) and bubble point oil formation volume factor ( $B_{ob}$ ), using a collection of 782 data sets from some previous researchers. They reported to have carried out quality to remove redundant data and unuseful observations. They compared their results with those of ANN model and correlations in [1, 24 and 64] and SVM had the best performance. Lastly, Hajizadeh [30] predicted viscosity for Iranian crude oil using Genetic Algorithms. The input parameters are: pressure, temperature, gas/oil ratio and oil density. He reported a testing correlation coefficient of 99.74%.

In all these predictions, data points are being predicted even for properties like gas oil ratio and viscosity that are usually represented by curves over required reservoir pressure. However, if we predict the data points and want to plot the regular curves, we might rather have a scattered plot; hence, compromising the consistency in the behaviour of the fluid property and such a resulting curve will have no practical use. This is what this work aims to address; prediction of the entire curve for some PVT properties over the required reservoir pressure.

## **CHAPTER THREE**

### **SOFT COMPUTING TECHNIQUES**

#### **3.1. Artificial Neural Networks (ANN)**

Neural Networks or Artificial Neural Networks to be more precise, represent an emerged technology rooted in many disciplines. They are endowed with some unique attributes (just like other soft computing (SC)/Artificial Intelligence (AI)) techniques: universal approximation (input-output mapping), the ability to learn from data and adapt to their environment and the ability to invoke weak assumptions about the underlying physical phenomena responsible for the generation of the input data. A neural Network is a massively parallel distributed processor that has a natural propensity for storing experimental knowledge and making it available for use. It resembles the brain in two respects:

1. Knowledge is acquired by the network through a learning process.
2. Interneuron connection strengths known as synaptic weights are used to store knowledge.

The procedure to perform the learning process is called a learning algorithm. The synaptic weights of the network are modified in an orderly fashion so as to attain a desired design objective [39].

##### **3.1.1. Benefits of Neural Networks**

The use of neural networks offers the following useful properties and capabilities:

1. Neurobiological Analogy: The design of a neural network is motivated by analogy with brain, which is a living proof that fault-tolerant parallel processing is not only physically possible but also fast and powerful.

2. **Nonlinearity:** A neuron is basically a nonlinear device. Consequently, a neural network, made up of an interconnection of neurons, is itself nonlinear. Moreover, the nonlinearity is of a special kind in the sense that it is distributed throughout the network. Nonlinearity is a very important property, particularly if the underlying physical mechanism responsible for the generation of an input signal (e.g., speech signal) is inherently nonlinear.
3. **Input-Output Mapping:** A popular paradigm of learning called supervised learning involves the modification of the synaptic weights of a neural network by applying a set of labeled samples or task examples. The network is presented an example picked at random from the set, and the synaptic weights of the network are modified so as to minimize the difference between the desired response and the actual response of the network produced by the input signal in accordance with an appropriate statistical criterion. Thus the network learns from the examples by constructing an input-output mapping for the problem at hand.
4. **Adaptivity:** Neural Networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions. The natural architecture of a neural network for pattern classification, signal processing, and control applications, coupled with the adaptive capability of the network, make it an ideal tool for use in adaptive pattern classification, adaptive signal processing, and adaptive control.

5. **Evidential Response:** In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also to reject ambiguous patterns, should they arise, and thereby improve the classification performance of the network. This is the same for prediction.
6. **Contextual Tolerance:** Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network. Consequently, contextual information is dealt with naturally by a neural network.
7. **Fault Tolerance:** A neural network, implemented in hardware form, has the potential to be inherently fault tolerant in the sense that its performance is degraded gracefully under adverse operating condition.
8. **VLSI Implementability:** The massively parallel nature of a neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network ideally suited for implementation using very-large-scale-integrated (VLSI) technology. The particular virtue of VLSI is that it provides a means of capturing truly complex behaviour in a highly hierarchical fashion, Mead and Conway [50]. This makes it possible to use a neural network as a tool for real-time applications involving pattern recognition, signal processing, and control.
9. **Uniformity of Analysis and Design:** Basically, neural network enjoy universality as information processors.



### 3.1.2. Model of a Neuron

A neuron is an information-processing unit that is fundamental to the operation of a neural network. In essence, it composes of signal processing elements called neurons. The model of a neuron is shown in Figure 3.1. The three basic elements of the neuron are described thus.

1. A set of *synapses or connecting links*, each of which is characterised by a weight or strength of its own. Specifically, a signal  $x_j$  at the input of synapse  $j$  connected to neuron  $k$  is multiplied by the synaptic weight  $w_{kj}$ .
  2. An *adder* for summing the input signals, weighted by the respective synapses of the neuron.
  3. An *activation function* for limiting the amplitude of the output of a neuron.
- The activation function is also referred to as *squashing function*. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval  $[0 \ 1]$  or alternatively  $[-1 \ 1]$ .

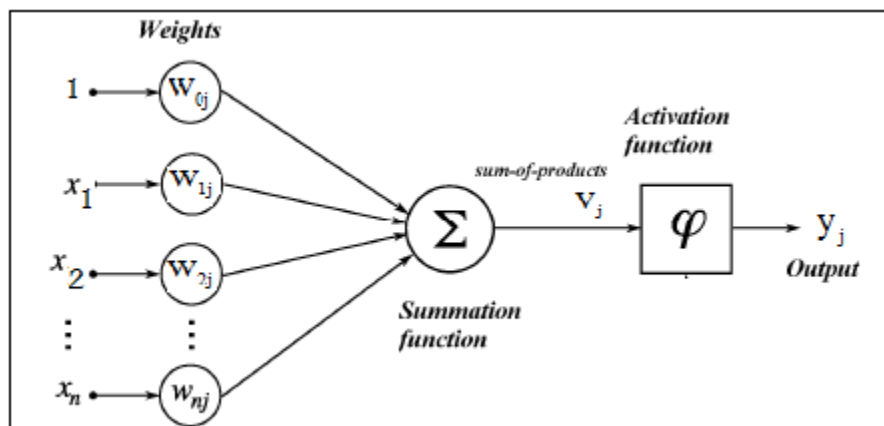


Figure 3.1: Nonlinear Model of a Neuron

Mathematically,

$$V_j = \sum_{k=1}^n W_{kj} \cdot X_k + W_{oj} \quad (3.1)$$

$$Y_j = \varphi(V_j) \quad (3.2)$$

Where  $X_1, X_2, \dots, X_n$  are the input signals.  $W_{1j}, W_{2j}, \dots, W_{nj}$  are the synaptic weights of neuron j,  $W_{oj}$  is the bias,  $V_j$  is the linear combiner output,  $f(\cdot)$  is an activation function and  $Y_j$  is the output signal of the neuron.

Some commonly used activation functions are:

1. *Threshold Function*: For this type of activation function, we have

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases} \quad (3.3)$$

2. *Piecewise-Linear Function*: This activation function is also referred to as saturating linear function. It can be a linear combiner if the region of operation is maintained without running into saturation. Also, it can reduce to a threshold function if the amplification factor of the linear region is made infinitely large.

$$\varphi(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ v & \text{if } +\frac{1}{2} > v > -\frac{1}{2} \\ 0 & \text{if } v \leq -\frac{1}{2} \end{cases} \quad (3.4)$$

3. *Sigmoid Function*: The sigmoid function is by far the most common form of activation function used in the construction of ANN's. It is defined as a strict activation function that exhibits smoothness and asymptotic properties. An example of the sigmoid is the logistic function, defined by;

$$\varphi(v) = \frac{1}{1 + e^{-av}} \quad (3.5)$$

where “a” is the slope parameter of the sigmoid function. When we vary the parameter “a”, we obtain sigmoid functions of different slopes.

4. *Tangent Hyperbolic Function*: This transfer function is often used in place of sigmoid function. It is described by the following mathematical form

$$\varphi(v) = \tanh(v) = \frac{e^{av} - e^{-av}}{e^{av} + e^{-av}} \quad (3.6)$$

### 3.1.3. ANN Architectures

The manner in which the neurons of a neural network are structured is intimately linked with the learning algorithm used to train the network. Therefore, people speak of learning algorithms (rules) used in the design of neural networks as being structured. In general, there are four different classes of network architecture.

#### 1. Single-Layer Feedforward Network

A layered neural network is a network of neurons organized in the form of layers. In the simplest form of a layered network, we just have an input layer of source nodes that project unto an output layer of neurons (computation nodes), but not vice versa. In other words, this network is strictly of a *feedforward* type. It is illustrated in Figure 3.2 for the case of four nodes in both input and output layers. Such a network is called a *single-layer* network, Haykin [32].

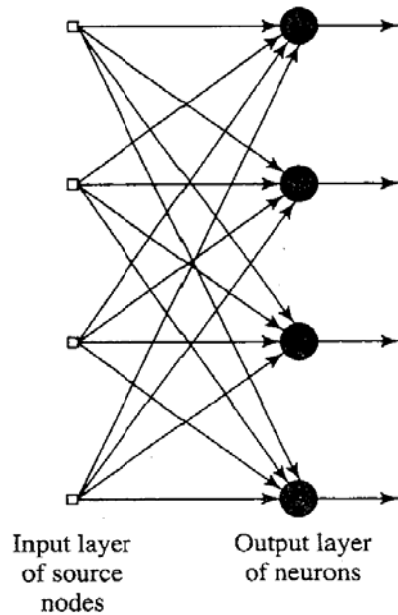


Figure 3.2: Single Layer Feedforward Network

## 2. Multilayer Feedforward Networks/Multilayer Perceptron (MLP)

The second architecture of neural networks is the mostly used in many applications including reservoir characterisation. This class of feedforward neural network, *multilayer feedforward network*, distinguishes itself by the presence of one or more *hidden layers*, whose computation nodes are correspondingly called *hidden neurons* or *hidden units*. The function of the hidden neurons is to intervene between the external input(s) and the network output(s). The presence of one or more hidden layers, whose neurons are correspondingly called hidden neurons enabled the network to extract higher order statistics, i.e. the network acquires a global perspective despite its local connectivity by virtue of the extra set of synaptic connections and extra dimension of neural interactions. Such a network is referred to as “multi-layer feedforward” network.

Multilayer networks can either be *fully-connected* or *partially connected*. The network is said to be fully connected when every node in each layer of the network is

connected to every other node in the adjacent forward layer. If, however, some of the communication links are missing from the network, we say that the network is partially connected. A form of partially connected feedforward network of particular interest is a locally connected network. A typical fully connected multilayer feedforward network is shown in Figure 3.3.

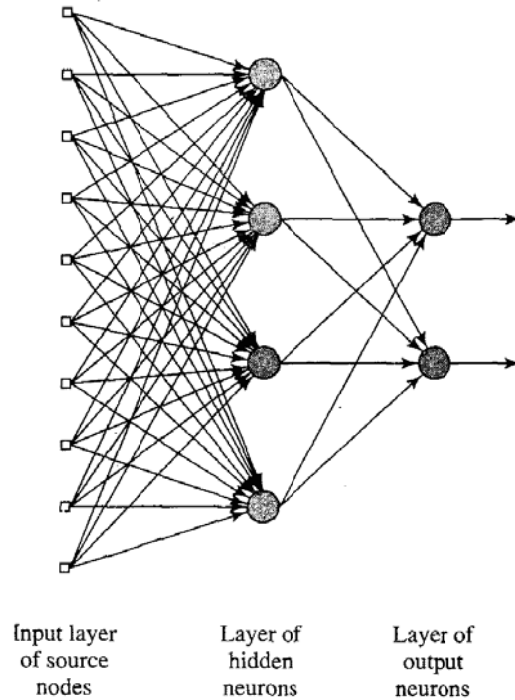


Figure 3.3: Fully Connected Multilayer Feedforward Network

### 3. Recurrent Networks

A recurrent neural network distinguishes itself from a feedforward neural network in that it has at least one feedback loop. For instance, a recurrent network may consist of a single layer of neurons with each neuron feeding its output signal back to the inputs of all other neurons. Two possible recurrent neural network architectures are depicted in Figures 3.4 and 3.5. The network in Figure 3.4 has feedback but no self-feedback. Self feedback refers to a situation where the output of a neuron is fed back to its own input.

The presence of feedback in both networks has a profound impact on their learning capability as well as its performance. Moreover, the feedback loops involve the use of particular branches composed of unit-delay elements. Hence, this finds its application in non-linear dynamic systems.

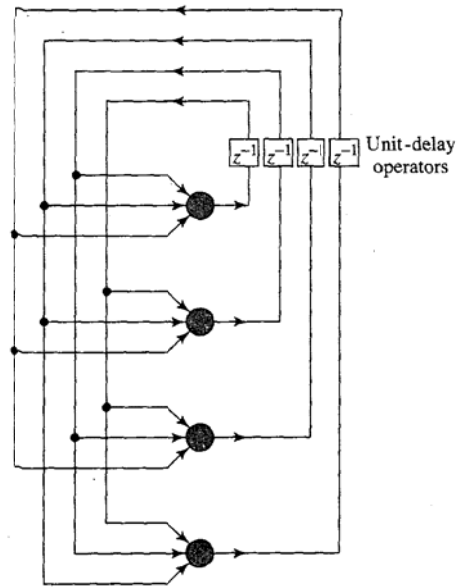


Figure 3.4: Recurrent Network with no Self-feedback Loops and no Hidden Neurons

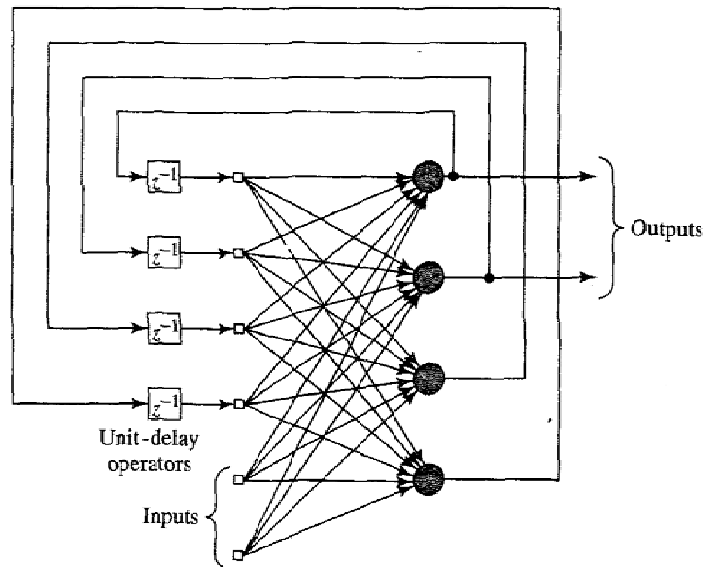


Figure 3.5: Recurrent Network with Hidden Neurons

### 3.1.4. Learning Algorithm for Neural Networks

We consider briefly the common learning algorithm for the mostly used ANN, MLP. The basic idea in the learning procedure is to provide the network with a training set of patterns having inputs and outputs. Real valued  $m$ -dimensional input feature vectors  $\mathbf{x}$  are presented to each of the first hidden layer units through weight vector  $\mathbf{w}$ . A hidden layer unit  $k$  receives input  $j$  through the synaptic weight,  $w_{kj}$ ,  $k = 1, 2, \dots, n$ , and  $j = 1, 2, \dots, m$ . The unit  $k$  computes a function of the input signal  $x$  and the weights  $w_{kj}$  and passes its output to all of the units in the next successive layer. Like the first hidden layer, the units of the second hidden layer are fully connected to the previous layer through the synaptic weights. These units also compute a function of their inputs and their synaptic weights and pass their outputs on to the next layer. The output of one layer becomes the input to the subsequent layer. Then at the output unit, an error is calculated between the target value and the computed value of the pattern. This process is repeated until the final computation is produced by the output unit when some criteria are met. The learning algorithm for this type of network is called the backpropagation (BP) algorithm which was published in the mid 1980s for multilayer perceptrons. Hornik et al [36] suggested that if a sufficient number of hidden units are available then an MLP with one hidden layer having a sigmoid transfer function in the hidden layer and a linear transfer function in the output layer can approximate any function to any degree of accuracy.

Backpropagation is a systematic method for training multilayer neural networks due to its strong mathematical foundation. The steps to implement the backpropagation algorithm are given as follows:

- The error signal at the output of neuron  $j$  at iteration  $n$  (i.e. presentation of the  $n$ th training pattern) is defined by

$$e_j(n) = d_j(n) - y_j(n) \quad (3.7)$$

where  $d_j(n)$  refers to the desired response for the neuron  $j$  and  $y_j(n)$  is the function signal appearing at the output of neuron  $j$  and  $e_j(n)$  refers to the error signal at the output of neuron  $j$ . The instantaneous value of the sum of squared errors is obtained by summing square error over all neurons in the output layer; which is written as:

$$\xi(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.8)$$

- The net internal activity level  $v_j(n)$  produced at the input of the nonlinearity associated with neuron  $j$  is therefore

$$v_j(n) = \sum_{i=0}^p w_{ji}(n) y_i(n) \quad (3.5)$$

where  $p$  is the total number of inputs (excluding the threshold) applied to neuron  $j$  and  $w_{ji}(n)$  denote the synaptic weight connecting the output of neuron  $i$  to the input of neuron  $j$  at iteration  $n$ . Hence the output of neuron  $j$  at iteration  $n$  is

$$y_j(n) = \varphi_j(v_j(n)) \quad (3.10)$$

- The instantaneous gradient which is proportional to the weight correction term is given as:

$$\begin{aligned} \frac{\partial \xi(n)}{\partial w_{ji}(n)} &= \frac{\partial \xi(n)}{\partial e_j(n)} \frac{\partial e_j(n)}{\partial y_j(n)} \frac{\partial y_j(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \\ \frac{\partial \xi(n)}{\partial w_{ji}(n)} &= -e_j(n) \varphi_j'(v_j(n)) y_j(n) \end{aligned} \quad (3.11)$$



- The correction  $\Delta w_{ji}(n)$  applied to  $w_{ji}(n)$  is defined by the delta rule

$$\Delta w_{ji}(n) = \eta \frac{\partial \xi(n)}{\partial w_{ji}(n)}$$

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

$$\delta_j(n) = e_j(n) \phi_j'(v_j(n)) \quad (3.12)$$

- When neuron  $j$  is located in a hidden layer of the network, the local gradient is redefined as

$$\delta_j(n) = -\frac{\partial \xi(n)}{\partial y_j(n)} \phi_j'(v_j(n)) \quad (3.13)$$

$$\delta_j(n) = \phi_j'(v_j(n)) \sum_k \delta_k w_{kj}(n) \quad (3.14)$$

where the  $\delta_k$  requires the knowledge of the error signals  $e_k$  for all those neurons that lie in the layer to the immediate right of hidden neuron  $j$ . The  $w_{kj}(n)$ , consists of the synaptic weights associated with these connections. We are now ready to put forward the weight correction update for the back-propagation algorithm, which is defined by the delta rule:

$$\Delta w_{ji}(n) = \eta \delta_j y_j \quad (3.15)$$

It is important to note that weight correction term depends on whether neuron  $j$  is an output node or a hidden node:

- If neuron  $j$  is an output node, equation (3.14) is used for the computation of the local gradient.
- If neuron  $j$  is a hidden node, equation (3.15) is used for the computation of local gradients.

- The network performance is checked by monitoring the average squared error. The average square error is obtained by summing  $\xi(n)$  over all  $n$  and then normalizing with respect to  $N$  (number of training patterns)

$$\xi_{av} = \frac{1}{N} \sum_{n=1}^N \xi(n) \quad (3.16)$$

Both the instantaneous and average squared errors are functions of free parameters (synaptic weights and biases). The process is repeated a number of times for each pattern in the training set until the total output squared error converges to a minimum or until some limit is reached in the number of training iterations. One of the major problems with BP algorithm is the long training time due to the steepest descent method, as it is a simple but slow minimization method. The learning rate is sensitive to the weight changes. The smaller the learning rate the smaller will be the changes to the synaptic weights from one iteration to the next, and the smoother will be the trajectory in the weight space.

On the other hand, if the learning rate is chosen to be too large in order to speed up the rate of learning, the resulting large changes in the synaptic weights make the network unstable. In order to speed up the convergence of BP algorithm along with improved stability, a momentum term is added to the weight update of the BP algorithm. A momentum term is simple to implement and this significantly increases the speed of convergence. The inclusion of momentum term represents a minor modification to the weight update. The inclusion of momentum may also have the benefit of preventing the learning process from terminating in shallow local minima on the error surface.

The second method of accelerating BP algorithm is by using Levenberg - Marquardt BP (LMBP) algorithm, Hagan et al. [29]. It is based on Newton's optimization method (Hagan et al. [29]) and differs from the usual BP algorithm in the manner in which the resulting derivatives are used to update the weights. The main drawback of the algorithm is the need for large memory and storage space of the free parameters in the computers. If the network has more than a few thousand parameters, the algorithm can take a long time to converge. In this study, the feed forward network architecture used for our comparison has been designed to have number of free parameters to be smaller than the number of training patterns in order for Levenberg-Marquardt BP (LMBP) algorithm to be adequate for training the network.

### **3.1.5. Drawbacks of Artificial Neural Network**

The main problem of ANN is its opacity or black-box nature. The associated lack of explanation capabilities is a handicap in some decision support applications such as medical diagnostics, where the user would usually like to know how the model came to a certain conclusion. Model parameters are buried in large weight matrices, making it difficult to gain insight into the modelled phenomenon or compare the model with available empirical or theoretical models. Information on the relative importance of the various inputs to the model is not readily available, which hampers efforts for model reduction by discarding less significant inputs. Additional processing techniques such as the principal component analysis may be required for this purpose.

### 3.2. Overview of Support Vector Machines (SVM)

Support Vector Machines are an attractive approach to data modelling. They combine generalisation control with a technique to address the curse of dimensionality. The formulation results in a global quadratic optimisation problem with box constraints, which is readily solved by interior point methods. The kernel mapping provides a unifying framework for most of the commonly employed model architectures, enabling comparisons to be performed. SVM which was primarily developed for classification problems has also been recently extended to regression problems. In classification problems, generalisation control is obtained by maximising the margin, which corresponds to minimising the weight vector in a canonical framework. The solution is obtained as a set of support vectors that can be sparse. These lie on the boundary and as such summarise the information required to separate the data. Figure 3.6 shows how a margin is created between two sets of data in a classification problem.

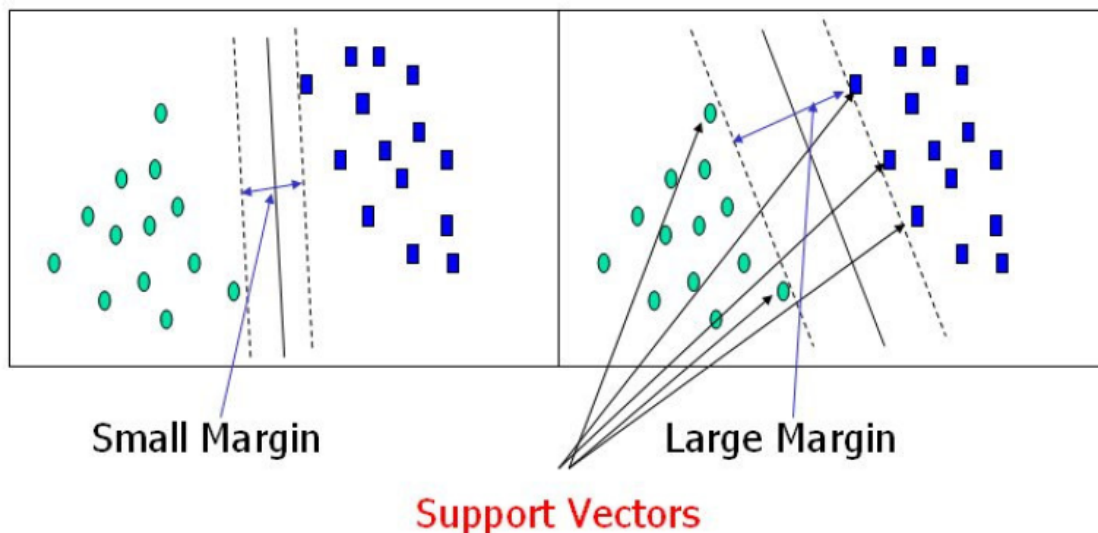


Figure 3.6: Creation of margins between two data sets by support vectors

### 3.2.1. Support Vector Regression (SVR)

Drucker et al. [19] introduced Support Vector Regression (SVR) as a regression version of Support Vector Machines (SVMs) which was primarily developed to solve pattern recognition and classification problems. Unlike classification problems where the outputs are either 1 and 0 or 1 and -1, the outputs in the regression problems are real numbers. This makes it a bit difficult to model this type of information which has infinite possibilities. In the case of regression, a margin of tolerance  $\epsilon$  is set in approximation to the SVM which would have already been inferred from the problem. As shown in Figure 3.7, SVMs map input vectors to a higher dimensional space where a maximal separating hyperplane is constructed, Littman [46], Cortes and Vapnik [14].

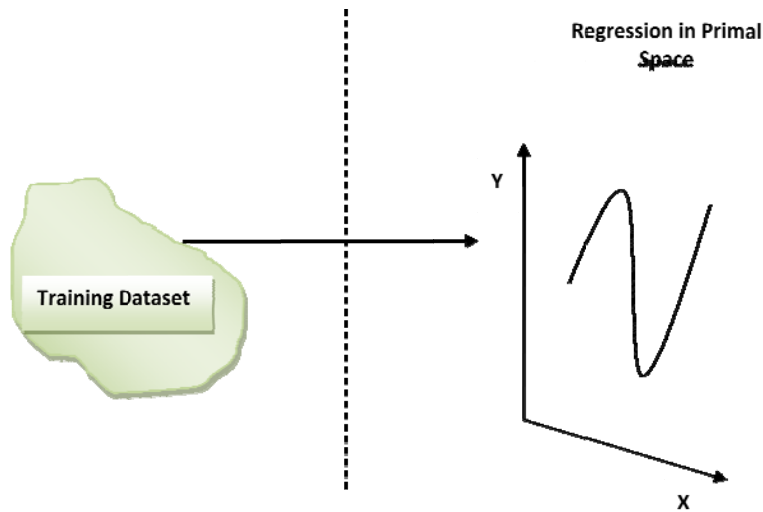


Figure 3.7: SVR maps input vectors to a higher dimensional space

Though, the main idea is always the same: for both SVR and SVM for classification, that is, to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. Error toleration is depicted in Figure 3.8.

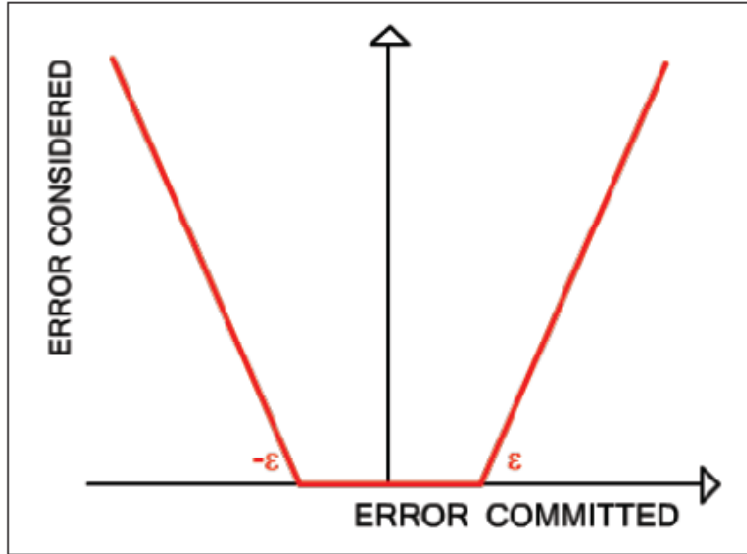


Figure 3.8: The error function

Figure 3.8 shows how the error of SVR is calculated. Up until the threshold is reached, the error is considered 0, after that it is calculated as “error-epsilon”. The solution to the problem is known as a “tube”, Figure 3.9.

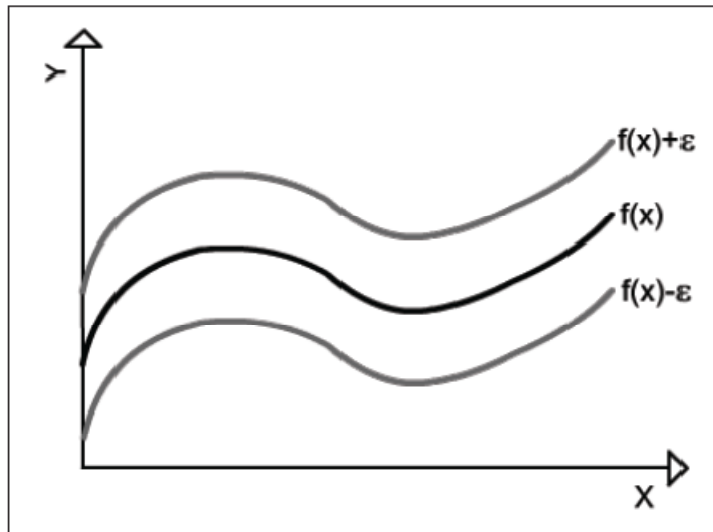


Figure 3.9: The Support Vector Regression Tube

Mathematically, since the main idea is to optimize the margin then the quadratic optimization problem becomes

$$\min_w \frac{1}{2} W^T W$$

$$s.t. \begin{cases} y_i - (W^T \phi(X) + b) \leq \varepsilon \\ (W^T \phi(X) + b) - y_i \leq \varepsilon \end{cases} \quad (3.17)$$

Where  $\phi(X)$  is the kernel function,  $W$  is the margin and the pair  $(x_i, y_i)$  is the training set. Then we add a bound in order to set the tolerance on errors number that can be committed:

$$\min_{W, b} \frac{1}{2} W^T W + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (3.18)$$

$$s.t. \begin{cases} y_i - (W^T \phi(X) + b) \leq \varepsilon + \xi_i \\ (W^T \phi(X) + b) - y_i \leq \varepsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0, i = 1, 2, \dots, l \end{cases} \quad (3.19)$$

This principle is similar to SVM for classification. Once it is trained, SVR will generate predictions using the following formula:

$$f(X) = \sum_{i=1}^l \theta_i \phi(X, X_i) + b \quad (3.20)$$

For the kernel, possible options are functions such as: Gaussian, polynomial, radial basis and wavelet. The kernel plays the most important role in determining the accuracy of SVR prediction. Of all the kernels that we tested in our simulations, Gaussian kernel gave the best performances in predictions. Next to it was the polynomial kernel.

### 3.2.2. Description of SVR Parameters

1. *Kernel Function*: The kernel function is responsible for transforming the data set into hyperplane. The variables of the kernel must be computed accurately since they determine the structure of high-dimensional feature space which governs the complexity of the final solution. The most commonly used kernel functions in the literature are:

- I. Linear:  $\phi(X_i, X_j) = (X^T X + \gamma)$
- II. Polynomial:  $\phi(X_i, X_j) = (\delta + \gamma X^T X)^d$
- III. Gaussian:  $\phi(X_i, X_j) = \exp\left(\frac{\|X_i - X_j\|^2}{-2\sigma^2}\right)$
- IV. Sigmoid:  $\phi(X_i, X_j) = \tan sh(\gamma(X_i X_j) - \delta)$
- V. Fourier series:  $\phi(X_i, X_j) = \frac{\sin(2n+1)(X_i - X_j)}{\sin\left(\frac{1}{2}(X_i - X_j)\right)}$

Where  $\gamma$  is the gain,  $\delta$  is the offset,  $d$  is the degree of the polynomial kernel and  $\sigma^2$  is the bandwidth of the Gaussian kernel.

2. *Regularization parameters (C)*: This determines the trade-off cost between minimizing the training error and minimizing the model's complexity.

3. The tube size of the  $\varepsilon$ -insensitive loss function ( $\varepsilon$ ): This is equivalent to the approximation accuracy placed on the training data.

4. Bandwidth of the kernel function  $\sigma^2$ : This represents the variance of the Gaussian kernel function.



### 3.3. Functional Networks

Functional networks were introduced as a powerful alternative to neural networks, Castillo [11] and Castillo et al. [13]. Unlike neural networks, functional networks have the advantage that they use domain knowledge in addition to data knowledge. The network initial topology can be derived based on the modeling of the properties of the real world. Once this topology is available, functional equations allow one to obtain a much simpler equivalent topology. Although functional networks also can deal with data only, the class of problems where functional networks are most convenient is the class where the two sources of knowledge about domain and data are available, Castillo et al. [12].

Functional networks as a new modelling scheme have been used in solving both prediction and classification problems. It is a general framework that is useful for solving a wide range of problems in engineering, statistics, and functions approximations.

#### 3.3.1. Background and Definition of Functional Networks

A functional network is defined as a pair,  $(X, \varphi)$  where  $X$  is a set of nodes and  $\varphi = \{(Y_j, f_j, Z_j), j = 1, 2, \dots, m\}$  is a set of neuron functions/functional units over  $X$ , such that, every node  $X_j \in X$  must be either an input or an output node of at least one neuron function in  $\varphi$ . For all  $j$ , a node  $X_j \in X$  is called a multiple node if it is an output of more than one neuron function. Otherwise, it is called a simple node. Other facts about functional networks are itemized as following.

- A functional unit (also called a neuron)  $\varphi$  over the set of nodes  $X$  is a triplet  $(X, f, Z)$ , where  $(Y, Z) \subset X$ ;  $Y \neq \phi, Z \neq \phi, Y \cap Z = \phi$  and  $f : Y \rightarrow Z$  is a given

function. We say that  $Y$ ,  $Z$ , and  $f$  are the set of input nodes, the set of output nodes, and the processing function of the functional unit  $\varphi$  respectively.

- An input node in a Functional Network  $(X, \varphi)$  is the input node of at least one functional unit in  $\varphi$  and is not the output of any functional unit in  $\varphi$ .
- An output node in a Functional Network  $(X, \varphi)$  is the output node of at least one functional unit in  $\varphi$  and is not the input of any functional unit in  $\varphi$ .
- An intermediate node in a Functional Network is the input node of at least one functional unit in  $\varphi$  and, at the same time, is the output node of at least one functional unit in  $\varphi$ .

### 3.3.2. Differences between Functional and Neural Networks

There is no doubt that functional networks are motivated from neural networks, however, their structures and the way they handle a problem are different. The characteristics and key features of functional networks, as compared with those of neural networks are shown in Figures 3.10 and 3.11.

1. In selecting the topology of functional networks, the required information can be derived from the data, from domain knowledge, or from different combinations of the two. In the case of standard neural networks, only the data are used. This implies that, in addition to the data information, other properties of the function being modelled by the functional networks can be used for selecting its topology (associativity, commutativity, invariance, etc.). This information is available in some practical cases.
2. Unlike standard neural networks, where the neuron functions are assumed to be fixed and known and only the weights are learned, in functional networks, the functions are

learned during the structural learning (which obtains simplified network and functional structures) and estimated during the parametric learning (which consists of obtaining the optimal neuron function from a given family).

3. Arbitrary neural functions can be assumed for each neuron, while in neural networks, they are fixed sigmoidal functions.

4. In functional networks, weights are not needed, since they can be incorporated into the neural functions.

5. The neural functions are allowed to be truly multiargument in functional networks[e.g., neural functions  $f_1$ ,  $f_2$  and  $f_3$  in Figure 3.11]. However, in many cases, they can be equivalently replaced by functions of single variables. Note that in standard neural networks the neural sigmoidal functions are of a single argument though this is a linear combination of all inputs (pseudo-multiargument functions).

6. In functional networks, intermediate or output units can be connected (linked) to several storing units, say  $m$  units, indicating that the associated values must be equal. Each of these common connections represents a functional constraint in the model and allows writing the value of these output units in different forms (one per different link). This leads to a system of  $m - 1$  functional equations. By solving this system, the initial neuron functions can be simplified, for example, by reducing the number of arguments. Intermediate layers of units are introduced in functional network architectures to allow several neuron outputs to be connected to the same units, which is not possible in neural networks.

7. Functional networks are extensions of neural networks. In other words, neural networks are special cases of functional networks. For example, in Figure 3.11, the neural

network and its equivalent functional network are shown. Note that weights are subsumed by the neural functions.

### 3.3.3. Methods of Selecting Functional Network Models

To learn (parametric) functional networks, we can choose different sets of linearly independent functions for the approximation of the neuron functions. At this same time, there is need for us to select the best model according to some criterion of optimality. Minimum Description Length Principle (MDLP) is one of the model selection principles we can use as discussed in Castillo et al. [13].

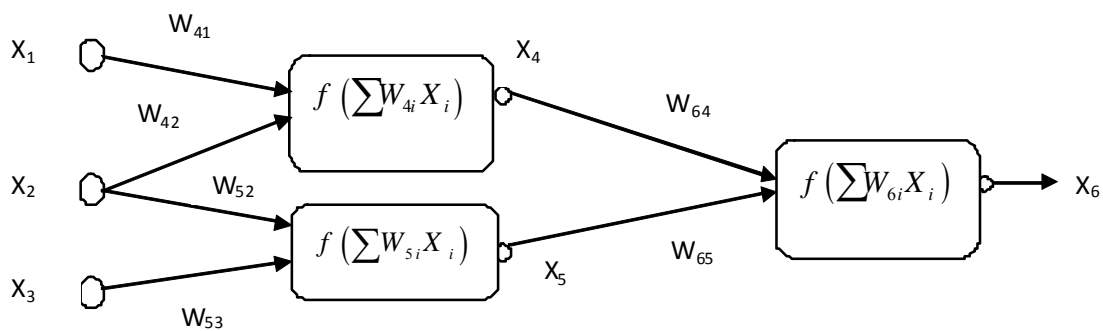


Figure 3.10: A Standard Neural Network

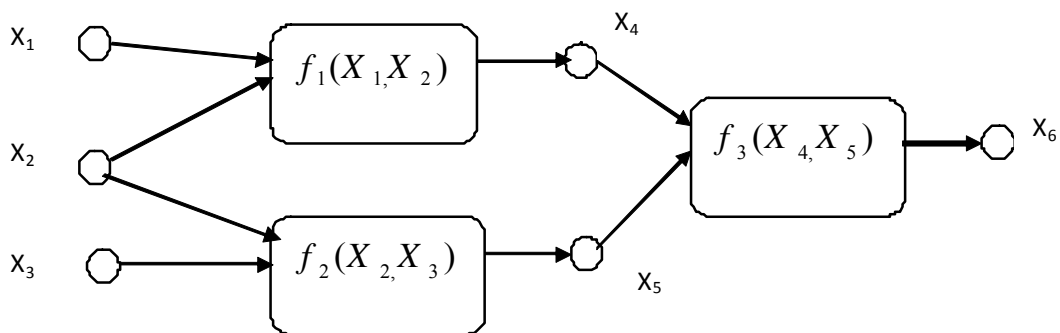


Figure 3.11: A Standard Functional Network

The idea behind the MDLP measure is to find the minimum information required to store the given training set using the functional network model. Therefore, we can say that the best functional network model for a given problem corresponds to that with the minimum description length value. The code length  $L(x)$  of  $x$  is defined as the amount of memory needed to store the information  $x$ . For example, assume we have a data set  $(x_{1j}, x_{2j}, x_{3j}, \dots, x_{13j}, y_j) | j \in J$  where  $(x_{1j}, x_{2j}, x_{3j}, \dots, x_{13j}) | j \in J$  are the inputs and  $(y_j | j \in J)$  is the output. (NOTE: This actually refers to our case). To store these data we have two options:

**Option 1: Store Raw Data:**

Store  $(x_{1j}, x_{2j}, x_{3j}, \dots, x_{13j}, y_j) | j \in J$ . In this case, the initial description length of the data set is

$$DL = \sum_{j=1}^n [L(x_{1j}) + L(x_{2j}) + \dots + L(x_{13j}) + L(y_j)] \quad (3.21)$$

**Option 2: Use a Model:**

By selecting a model, we try to reduce this length as much as possible. In this case, we can store the parameters of the model  $\beta_k, k \in K$  and then the residuals are

$$e_j = y_j - \hat{g}^{-1} \left[ \hat{g} \left( (x_{1j}) + \hat{g}(x_{2j}) + \dots + \hat{g}(x_{13j}) \right) \right], j \in J \quad (3.22)$$

where  $\hat{g}$  are the approximate neuron functions of the model. The description length becomes:

$$DL_{model} = \sum_{j=1}^J [L(x_{1j}) + L(x_{2j}) + \dots + L(x_{13j})] + \sum_{k=1}^K L(\beta_k) + \sum_j L(e_j | model) \quad (3.23)$$

Where  $L(\beta_k)$  is the code length of the estimated parameters  $\beta_k, k \in K$ .

Generally, the description length is a measure that allows comparing not only the quality of different approximations, but also different functional network models. The description length measure can be calculated for any model. In addition, it is used to compare models with different parameters, because it has penalty term for over-fitting.

Moreover, it is distribution independent. This makes the minimum description length a convenient method for solving the model selection problem. Accordingly, the best functional network model for a given problem corresponds to the one with the smallest description length value. To achieve this goal the following methods could be used:

*The Exhaustive Search:* This method computes the MDL measure for all possible models and choose the one leading to the smallest value of the error measure. The obvious shortcoming of this method is its computational complexity.

*The Forward Method:* This method starts with all models of a single parameter and selects the one leading to the smallest value of  $L(x)$ . Next, it incorporates one more parameter by selecting the new one leading to the smallest value of  $L(x)$ . The process continues until no improvement in  $L(x)$  is obtained.

*The Backward method:* This method starts with the model with all parameters and first removes the one leading to the smallest value of  $L(x)$ . Next, removes one more parameter by selecting the one leading to the smallest value of  $L(x)$ . The process continues until no improvement in  $L(x)$  is obtained.

*The Backward-Forward method:* The backward process starts with the complete model with all parameters and sequentially removes the one leading to the smallest value of the

MDL measure, repeating the process until no improvement in the measure. Next, the forward process is applied, but starting from the final model of the backward process, and sequentially adds the one variable that leads to the smallest value of MDL measure, repeating the process until no improvement in the measure. This process is repeated until no further improvement in MDL measure is obtained neither by removing nor by adding a single variable.

*The Forward-Backward method:* The forward process starts with all models of a single parameter and selects the one leading to the smallest value of  $L(x)$ . Next, it incorporates one more parameter with the same criterion and the process continues until no improvement in  $L(x)$  can be obtained by adding an extra parameter to the previous model. Then the backward process is applied, but starting from the final model of the forward process and sequentially remove the one variable that is leading to the smallest value of MDL measure, repeating the process until no improvement in  $L(x)$  is possible. The double process is repeated until no further improvement in  $L(x)$  is obtained neither by adding nor by removing a single variable.

### 3.3.4. Development of Functional Network Model

The functional network that was finally adopted is summarized as follows.

- **Objective**

Given a data set  $\{x_{ij}|y_i; i = 1,2, \dots, n \ \& \ j = 1,2, \dots, 13\}$  where  $x_{ij}$ 's are the predictors and  $y_i$  is the output. Mathematically, the relationship is given by

$$Y = f(X_1, X_2, \dots, X_{13}) \tag{3.24}$$

The aim is to get  $\hat{Y}$  which is an estimate of  $Y$  such that the square of the error is minimized. That is

$$\min \left\{ \frac{1}{n} \sum_{i=1}^n \left( Y_i - \hat{Y}_i \right)^2 \right\}$$

- **Functional Network Model**

A generalised model of functional network that learns from data is given as

$$y = \sum_{r_1}^{m_1} \dots \sum_{r_k}^{m_k} C_{r_1 r_2 \dots r_k} \varphi_{r_1}(x_1) \dots \varphi_{r_k}(x_k) \quad (3.25)$$

Where  $C_{r_1 r_2 \dots r_k}$  are the unknown parameters and  $\varphi_j (j = 1, 2, \dots, k)$  are linearly independent functions.

- **Uniqueness and Simplification of the Model**

To prove the uniqueness of the generalized model in equation (3.25), we assume two sets of parameters  $C_{r_1 r_2 \dots r_k}$  and  $C^*_{r_1 r_2 \dots r_k}$  such that:

$$\sum_{r_1}^{m_1} \dots \sum_{r_k}^{m_k} C_{r_1 \dots r_k} \varphi_{r_1}(x_1) \dots \varphi_{r_k}(x_k) = \sum_{r_1}^{m_1} \dots \sum_{r_k}^{m_k} C^*_{r_1 r_2 \dots r_k} \varphi_{r_1}(x_1) \dots \varphi_{r_k}(x_k) \quad (3.26)$$

This equation can be re-written as:

$$\sum_{r_1}^{m_1} \dots \sum_{r_k}^{m_k} (C_{r_1 \dots r_k} - C^*_{r_1 \dots r_k}) \varphi_{r_1}(x_1) \dots \varphi_{r_k}(x_k) = 0 \quad (3.27)$$

Since the set of functions  $\varphi_j (j = 1, 2, \dots, k)$  are linearly independent, then

$(C_{r_1 \dots r_k} - C^*_{r_1 \dots r_k}) = 0$  for all  $r_1 \dots r_k$ . This implies  $C_{r_1 \dots r_k} = C^*_{r_1 \dots r_k}$  and hence,

equation (3.26) is unique.



In the same vein, we can simplify the general functional form further by assuming that all coefficients of the cross-multiplication terms between the functions of different variables rather than one are equal to zero. Then equation (3.25) becomes

$$y = f_1(x_1) + f_2(x_2) + \dots + f_k(x_k) \quad (3.28)$$

That is;

$$y = \sum_{r_1}^{m_1} C_{r_1} \varphi_{r_1}(x_1) + \sum_{r_2}^{m_2} C_{r_2} \varphi_{r_2}(x_2) + \dots + \sum_{r_k}^{m_k} C_{r_k} \varphi_{r_k}(x_k) \quad (3.29)$$

where  $f(x)$  is the sum of the basis functions for each predictor.

Equation (3.29) can further be written as:

$$y = \sum_{k=1}^p \sum_{r_k}^{m_k} C_{r_k} \varphi_{r_k}(x_k) \quad (3.30)$$

- **Learning the Functional Network Model**

Given the data set  $\{x_{i1}, x_{i2}, \dots, x_{ik} | y_i; i = 1, 2, \dots, n\}$ , the general form of a functional network that learns from the data has earlier been reduced to

$$y_i = \sum_{k=1}^p \sum_{r_k}^{m_k} C_{r_k} \varphi_{r_k}(x_{ik}) \quad , \quad i = 1, 2, \dots, n \quad (3.31)$$

where  $n$  is the number of observations of response  $Y$  and  $k$  is the number of predictors.  $\varphi_{r_k}$  is the linear combinations of selected linearly independent functions and  $C_{r_k}$  are the coefficients of  $\varphi_{r_k}$ . Some of the commonly used linearly independent (basis) functions are:

- 1). Polynomial function:  $\varphi = \{1, x, x^2, \dots, x^m\}$
- 2). Exponential Function:  $\varphi = \{1, e^x, e^{-x}, e^{2x}, e^{-2x}, \dots, e^{mx}, e^{-mx}\}$

3). Fourier Function:  $\varphi = \{1, \sin(x), \cos(x), \sin(2x), \cos(2x), \dots, \sin(mx), \cos(mx)\}$

4). Logarithm Function:  $\varphi = \{1, \log(x + 2), \log(x + 3), \dots, \log(x + m)\}$

NOTE: It is possible to use different combinations of these basis functions.

If  $\hat{Y}$  is the estimate of  $Y$  from equation (3.31), the problem reduces to minimization of the error  $\varepsilon$  between  $Y$  and  $\hat{Y}$ . Hence, we have

$$\varepsilon = \min(Y - \hat{Y}) \quad (3.32)$$

From equation (3.31), we can write  $\hat{Y}$  as

$$\hat{Y} = WC^* \quad (3.33)$$

$$\varepsilon = \min(Y - WC^*) \quad (3.34)$$

Using least square optimization technique, equation 3.34 is solved for  $C^*$  and we have

$$C^* = (W^T W)^{-1} W^T Y \quad (3.35)$$

Equation 3.35 gives the unknown optimal coefficients,  $C^*$ .

### 3.4. Adaptive Network Fuzzy Inference System (ANFIS)

This is also known as *Adaptive Neuro-Fuzzy Inference System* or *Neuro-Fuzzy Systems* or *Fuzzy Neural Networks (FNN)*. The word “neuro” or “neural” is actually used because the inference system borrows learning methodology from ANN which has been well-established before it, Jang [37] and Mendel [51].

#### 3.4.1. Development of Fuzzy Inference System

The concept of uncertainty / vagueness that fuzzy logic was developed to handle has been with us for long. Vagueness and uncertainty has been discussed extensively by philosophers both past and present, Russell [61], Black [10], and Keefe and Smith [41]. Lotfi Zadeh was the first person to use the term fuzzy logic, Zadeh [74]

It is an established fact that the world is imprecise, uncertain and vague, and decision making systems usually emulate human expertise, hence the needs for fuzzy logic based reasoning to cater for natural uncertainties. Ever since its introduction, fuzzy logic has witnessed unprecedented successes particularly in the area of Applications in Control. Fuzzy systems have a number of attractions:

- They allow for linguistic description of a problem by an expert;
- They are often more robust than traditional mathematical approaches;
- The underlying reasoning process can be examined.

They do however have a major drawback in that they do not learn and therefore require significant human intervention from an expert. In particular the membership functions of the fuzzy sets have to be determined. This problem has been tackled by a number of researchers, Wang [72], Watanabe [73], and Turksen [68], but these approaches are often domain dependent and still require input from human expertise. An alternative approach is an adaptive fuzzy system, which offers the ability to learn from data, Cox [15] and Jang [38].

### **3.4.2 Learning Algorithm for ANFIS**

To illustrate how ANFIS works, we assume the fuzzy inference system under consideration has two inputs  $x$  and  $y$  and one output  $z$ . Suppose that the rule base contains two fuzzy if-then-rules of Takagi and Sugeno's type as follows [67].

Rule1: If  $x$  is  $A_1$  and  $y$  is  $B_1$  then  $f_1 = p_1x + q_1y + r_1$ ,

Rule2: If  $x$  is  $A_2$  and  $y$  is  $B_2$  then  $f_2 = p_2x + q_2y + r_2$ ,

Then the fuzzy reasoning and the corresponding equivalent ANFIS architecture are shown in Figure 3.12.

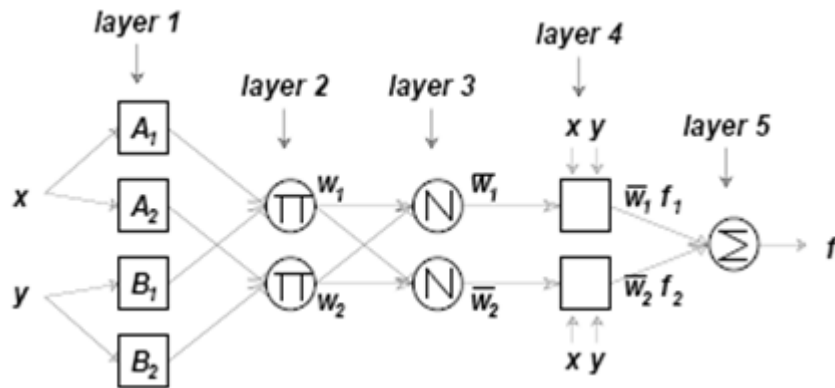
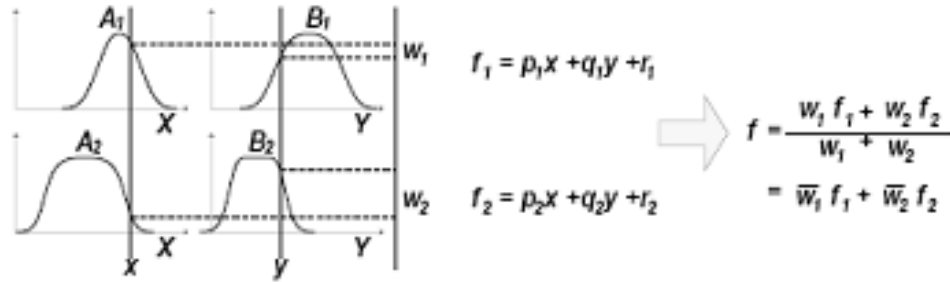


Figure 3.12: Fuzzy Reasoning and Its Equivalent ANFIS

The node functions in the same layer are of the same function family as described below:

**Layer 1:** Every node  $i$  in this layer is a square node with a node function:

$$O_i^1 = \mu A_i(x),$$

where  $x$  is the input to node  $i$ , and  $A_i$  is the linguistic label (small, large, etc.) associated with this node function. In other words,  $O_i^1$  is the membership function of  $A_i$  and it specifies the degree to which the given  $x$  satisfies the quantifier  $A_i$ . Usually we choose  $\mu A_i(x)$  to be bell-shaped with maximum and minimum equal to 1 and 0

respectively, such as:

$$\mu A_i(x) = \frac{1}{1 + \left[ \left( \frac{x - c_i}{a_i} \right)^2 \right]^{b_i}},$$

(3.36a)

Or 
$$\mu A_i(x) = \exp \left\{ - \left[ \left( \frac{x - c_i}{a_i} \right)^2 \right]^{b_i} \right\}, \quad (3.36b)$$

where  $\{a_i, b_i, c_i\}$  is the parameter set. As the values of these parameters change, the bell-shaped functions vary accordingly, thus exhibiting various forms of membership functions on linguistic label  $A_i$ . In fact, any continuous and piecewise differentiable functions are also qualified candidates for the node functions in this layer. Parameters in this layer are referred to as premise parameters.

**Layer 2:** Every node in this layer is a circle node labelled  $\Pi$  which multiplies the incoming signals and sends the product out. For instance,

$$w_i = \mu A_i(x) \times \mu B_i(y), \quad i = 1, 2. \quad (3.37)$$

Each node output represents the firing strength of a rule. In fact, it must be noted here that other firing T-norm operators that perform generalized AND can be used as the node function in this layer.

**Layer 3:** Every node in this layer is a circle node labelled  $N$ . The  $i$ -th node calculates the ratio of the  $i$ -th rule's firing strength to the sum of all rules' firing strengths:

$$\bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \quad (3.38)$$

For convenience, output of this layer is usually called normalized firing strengths.

**Layer 4:** Every node in this layer is a square node with a node function

$$O_i^4 = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad (3.39)$$

where  $\bar{w}_i$  is the output of layer 3, and  $\{p_i, q_i, r_i\}$  is the parameter set. Parameters in this layer will be referred to as consequent parameters.

**Layer 5:** The single node in this layer is a circle node labelled  $\Sigma$  that computes the overall output as the summation of all incoming signals, that is,

$$O_1^5 = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \quad (3.40)$$

The network described has a number of parameters to be learnt. Firstly, there are parameters for the membership grades of the antecedent type-1 fuzzy sets. Since the output of the network is numeric, this can be compared with the expected output from a supervisor (i.e. supervised learning) and back propagation (BP) and/or least mean square (LMS) algorithms can be used to feed the error back to adjust the parameters in the nodes.

### 3.5. Genetic Algorithm

Genetic algorithms (GA) are inspired by Darwin's theory about evolution. The solution to a problem solved by genetic algorithms is said to be evolved. The algorithm starts with a set of solutions (represented by **chromosomes**) called **initial population** [75]. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one.

#### 3.5.1. General Overview of Genetic Algorithm

In what follows, a brief overview of common terms in GA is presented.

**Chromosomes:** These contain information about the solution(s) which they represent. In ANN, the weights are the chromosomes while these are the subtractive clustering radii in

ANFIS in our models. These chromosomes are initiated in a population before genetic operations, basically crossover and mutation, are performed on them. These chromosomes must be encoded before genetic operations are performed on them. Some of the common encoding methods are: *Binary, Value, Permutation, Tree* and *Simulated Binary* encodings. An encoding method should be chosen based on the problem at hand [17].

**Crossover:** This is the process of selecting genes from the parent chromosomes and creating new offspring. This process can be rather complicated and it depends on the method used for encoding the chromosomes. Specific crossover made for a specific problem can improve the performance of GA.

**Mutation:** After a crossover is performed, mutation takes place. Ordinarily, it introduces random changes into the characteristics of chromosomes and prevents falling of all solutions (chromosomes) in a population into a local minimum of the solved problem. Basically, mutation changes randomly the new offspring. Mutation depends on the encoding methods as well as the crossover.

**Crossover Probability:** This determines how often crossover will be performed. If there is crossover, offspring are made from parts of parents' chromosomes. Crossover is performed with the hope that new chromosomes will have good part of old chromosomes and may be the new chromosomes will be better.

**Mutation Probability:** This determines the frequency of mutating parts of chromosomes. If there is no mutation, offspring are taken after crossover without any change, or else some of the chromosomes are crossovered while some are mutated.

**Population Size:** This is the number of chromosomes in a population. If there are too few

chromosomes, GA has a few possibilities to perform crossover and only a very few solution search space is explored. On the other hand, if there are too many chromosomes, GA slows down and hence takes a long time to complete.

**Methods for Chromosome Selection:** In selecting chromosomes from the population for genetic operation (crossover and mutation), several methods have been proposed. The selection process is often done based on the principle of “survival of the fittest”. Among them are: *roulette wheel selection*, *tournament selection*, *rank selection* and *elitism*. In *roulette wheel selection*, fitness is assigned to possible solutions (chromosomes) based on a pre-defined fitness function. The fitness level is then used to assign probability of selection to each individual chromosome. Because of probabilistic selection that is involved, *roulette wheel selection* allows some weak chromosomes to be selected for crossover.

In *tournament selection*, a "tournament" is run among a few individuals (chromosomes) chosen at random from the population and the winner (the one with the best fitness) is elected for crossover. This method can easily be adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

For the *rank selection* method, chromosomes are evaluated and ranked. The chromosome with the highest rank receives the highest fitness. A selection for crossover is then made based on the fitness for crossover. This method can lead to slower convergence when the best chromosomes are not so much different from others.

Another commonly used selection method of chromosomes for crossover is *elitism*. This method aims to solve the problem of losing the best chromosomes while



creating new population by genetic operators (crossover and mutation). In *elitism*, the best chromosomes or a few best chromosomes are first copied into the new population before performing any classical selection process.

### 3.5.2. Steps for Implementing a Genetic Algorithm

We outline the procedure for carrying out a simple genetic algorithm for an optimization task within a given solution space.

```
begin GA  
  
  g:=0 { generation counter }  
  
  state termination criterion/criteria (e.g. maximum no of generations allowed)  
  
  Initialize population P(g)  
  
  Evaluate population P(g) { i.e., compute fitness values }  
  
  while the termination criterion is not met, do  
  
    g:=g+1  
  
    Select P(g) from P(g-1)  
  
    Crossover P(g)  
  
    Mutate P(g)  
  
    Evaluate P(g)  
  
  end while  
  
end GA
```

Figure 3.13: Pseudo-code for a Simple Genetic Algorithm

In this study, two GA based hybrids have also been implemented for ANN and ANFIS. For the hybrid ANN, we used Differential Evolution (DE), which borrows its

concept from the traditional GA. In short, it is a genetic type of algorithm which was developed by Ken Price of Berkley University, USA [42]. Ilonen et al [37] first introduced a variant of DE for training neural networks. We used DE to train the feedforward neural network to form what is henceforth referred to as DE+ANN.

On the other hand, we built the hybrid ANFIS by developing and implementing the traditional concept of GA to search for the optimal radii of the data set in subtractive clustering during ANFIS implementation. This is also one of the contributions of this study.

## CHAPTER FOUR

### DATA ACQUISITION, APPROACH AND IMPLEMENTATION

#### 4.1. Introduction

The main contribution of this work is to develop a new approach in predicting viscosity and gas/oil ratio curves of PVT properties. Unlike the usual practice in prediction of PVT properties where a property that is generated as a curve is predicted through single or multi-data points, we present a simple way to predict any PVT properties that are generated as curves and vary over the entire required reservoir pressures. This has been demonstrated for two important PVT properties, viscosity and gas/oil ratio.

We implemented four independent Soft Computing techniques the predictions. These techniques are: Functional Network (FN), Support Vector Regression (SVR, may also referred to as SVM)), Adaptive Neuro-Fuzzy Inference Systems (ANFIS) and Artificial Neural Network (ANN). Two hybrid techniques are also introduced: Differential Evolution Algorithm with ANN (DE+ANN) and Genetic Algorithm with ANFIS (GA+ANFIS). That is, we have developed different models based on each of these overall six techniques for complete curve predictions of the considered PVT properties (viscosity and gas/oil ratio).

In our implementation, we optimized the MATLAB source codes for ANN and ANFIS for our models. Before implementation of ANFIS for prediction, we applied Subtractive Clustering to improve the performance of the predictions. Also for SVM and FN, we implemented the frameworks discussed in chapters four and five, using partially the source codes provided by the inventors of these techniques. For DE+ANN, we used

partially the source code for Differential Evolution developed by the authors in [37]. Lastly, we developed GA from scratch to implement GA+ANFIS. All simulations were carried out with MATLAB 7.5.0 (R2007b) on Pentium IV.

#### **4.2. PVT Data Acquisition and Processing**

There are three categories of data sets for the experimentations, namely, data sets A, B and C. Data set A consists of the hydrocarbon and non-hydrocarbon components of the crude oil and other relevant reservoir parameters. Data set B consists of viscosity-pressure measurements to generate viscosity curves for the corresponding wells in data set A, while data set C consists of gas/oil ratio-pressure measurements to generate gas/oil ratio curves for corresponding wells in A. These data were from Middle East crude oil.

Initially, there were 106 data points in set A. Statistical distribution of the data set A (which consists of the predictors) is shown in table 4.1. In preprocessing the data, we applied two different outlier-detection methods on it before utilizing it for prediction. The methods are: Cook's distance method and Chauvenet's criterion [35]. The former method was implemented using "STATISTICA" software while details on the latter method can be found in [35]. Only data points that were detected to be outliers by the two methods were declared as such and removed. Eventually, seven data points were declared as outliers. After removal of the outliers from data set A and the corresponding viscosity-pressure and gas/oil ratio-pressure measurements from data sets B and C respectively, data set A contains 99 points while data sets B and C have 1705 and 841 data points respectively.

### Explanation of Data

The following should be noted for the reported data in table 1.

- Mol\_n2, Mol\_co2 and Mol\_h2s: the mole fractions of gases  $N_2$ ,  $CO_2$ , and  $H_2S$  respectively.
- Sum\_mol\_C1-C3: summation of mole fractions of gases labelled C1, C2 and C3.
- Sum\_mol\_C4-C6: summation of mole fractions of hydrocarbons labelled iC4, nC4, iC5, nC5 and C6
- C7+: mole fraction of hydrocarbons C7+
- SumAppmol\_C1-C3: summation of apparent molecular weights of C1-C3
- SumAppmol\_C4-C6: summation of apparent molecular weights of C4-C6
- BPP( $P_b$ ): bubble point pressure
- API: oil specific gravity
- Res\_Temp: reservoir temperature
- $\mu_{od}$ : dead oil viscosity
- $\mu_{ob}$ : viscosity at bubble point pressure
- $R_{sb}$ : gas/oil ratio at bubble point pressure

Table 4.1: Description of Data Set A

Parameter	Max. Value	Min. Value
Mol_N <sub>2</sub>	2.43	0
Mol_CO <sub>2</sub>	8.66	0
Mol_H <sub>2</sub> S	12.6	0
Sum_mol_C1-C3	57.69	20.03
Sum_mol_C4-C6	18.72	8.8
Mol_C7	59.09	26.45
SumAppmol_C1-C3	33.28048	21.00485
SumAppmol_C4-C6	72.40335	66.67617
$P_b$	3202	381
API gravity	48	24.2
Res_Temp	240	130
$\mu_{od}$	5.99	0.85
$\mu_{ob}$	1.84	0.255
$R_{sb}$	1334	184

As recommended and usually done, we normalized the predictors between [0 1] using the formula in equation 4.1. This ensures that the predictors are independent of the measurement units.

$$x_i^{new} = \frac{(x_i^{old} - \min(x_i))}{(\max(x_i) - \min(x_i))}; \quad i = 1, 2, \dots, n \quad (4.1)$$

The data set (data set A) was then divided into training and testing sets. The training set consists of 70% (approx. 70 data points) while the testing set consists of 30% (approx. 29 points).

### 4.3. Approach and Problem Formulation

The approach used for formulating the problem of curve prediction before implementing the Soft Computing techniques for prediction is presented in what follows. This simple approach is being formulated for the first time. This formulation can be generalised for all PVT properties that can be generated as curves. The principle is pivoted on “*anchoring*” of the curve to a (or some) parameter that can easily be acquired.

#### 4.3.1. Viscosity Curve Prediction

A typical viscosity curve is shown in Figure 4.1 below. At point “a”, the viscosity is called dead oil viscosity ( $\mu_{od}$ ). Relatively, it is a parameter that can easily be acquired for a new well before the commencement of exploration. It is the viscosity of a fresh reservoir fluid. As the pressure increases, the fluid viscosity decreases until the saturation pressure (bubble point pressure) is reached at point “b”. This transition between “a” and “b” constitute the viscosity below bubble point ( $\mu_b$ ). As the pressure increases above this

saturation point, the fluid viscosity increases linearly as shown in Figure 4.1 between points “b” and “c”. The viscosity between “b” and “c” is called viscosity above bubble point ( $\mu_a$ ).

Meanwhile, the two transitions that exist between “a” and “b”, and between “b” and “c” (for  $\mu_b$  and  $\mu_a$  respectively) can easily be depicted perfectly with the following two equations respectively.

$$\frac{\mu_b - \mu_{od}}{\mu_{ob} - \mu_{od}} = \left( \frac{P - P_d}{P_b - P_d} \right)^\beta \quad (4.2)$$

$$\mu_a = \mu_{ob} + \alpha(P - P_b) \quad (4.3)$$

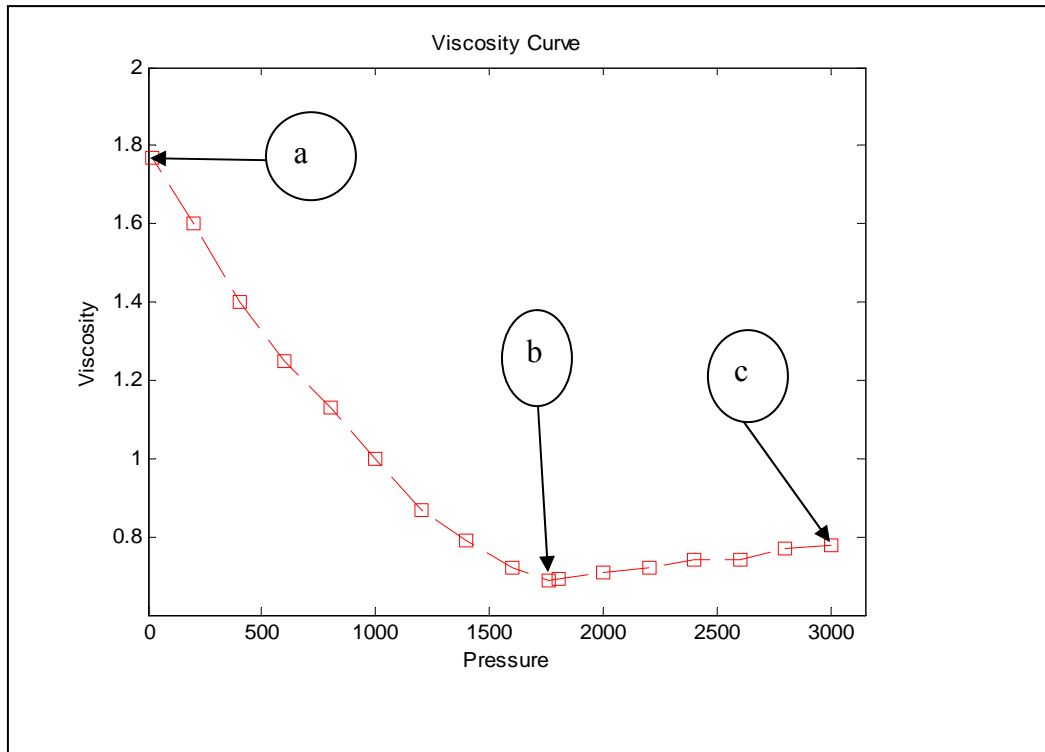


Figure 4.1: A Typical Viscosity Curve

Equation (4.2) is for the part “a” to “b” while equation (4.3) is for the linear part “b” to “c”. Equation (4.2) can be re-written as:

$$\mu_b = \mu_{od} + (\mu_{ob} - \mu_{od}) \left( \frac{P - P_d}{P_b - P_d} \right)^\beta \quad (4.4)$$

where  $\alpha$  and  $\beta$  are the fitting viscosity curve coefficients.

From these equations, there are three parameters that are needed to be predicted to generate a viscosity curve for a new oil well. These are  $\mu_{ob}$ ,  $\alpha$  and  $\beta$ . While we can easily predict  $\mu_{ob}$  from the first 12 variables shown in table 1, the problem arises on how to come about  $\alpha$  and  $\beta$  for a new well. The existing curves can be fitted using non-linear least square method to generate these two coefficients. A sample of a fitted viscosity curve is shown in Figure 4.2.

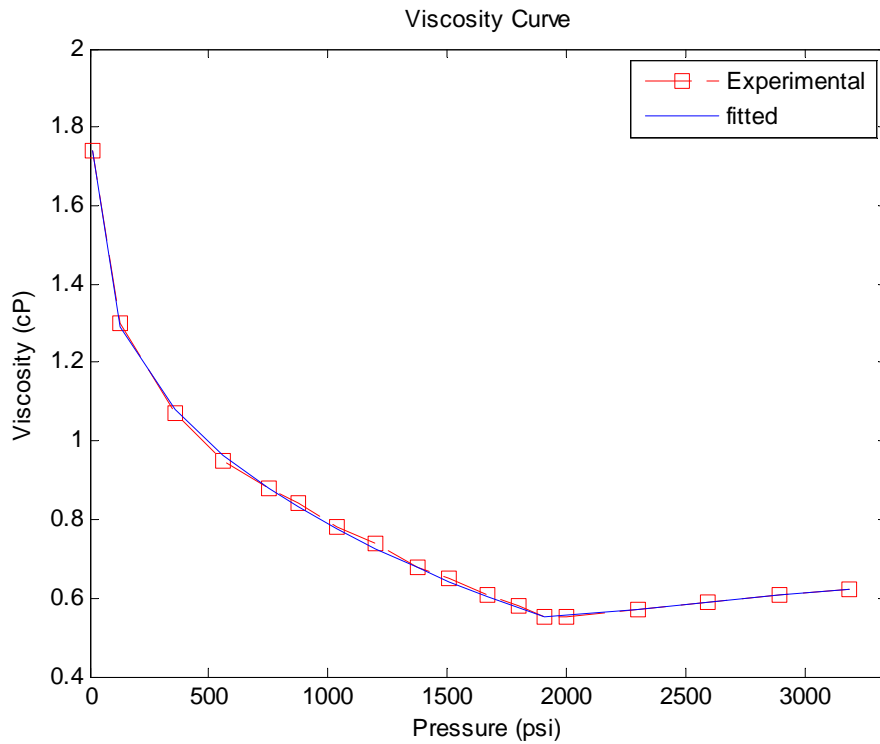


Figure 4.2: A fitted Viscosity Curve

The resulting  $\alpha$  and  $\beta$  which were generated from the non-linear least square curve fitting can then be trained using a SC technique. It should be noted that the



predicted viscosity curve is always anchored to the dead oil viscosity. All the six techniques mentioned in section 4.1 have been implemented for predicting each of the required variables ( $\alpha$ ,  $\beta$  and  $\mu_{ob}$ ). Seventy oil wells were used for training and 29 for testing. These simulation resources are aimed to be a useful tool that will save a lot of money and time that are always invested in the laboratory experimentations to generate viscosity and gas/oil ratio curves for the new wells.

### 4.3.2. Gas/Oil Ratio Curve Prediction

A typical gas/oil ratio curve is shown in Figure 4.3. The highest value of gas/oil ratio in the curve is at the bubble point pressure. Following the same trend as we did for viscosity, the curve is fitted and the only coefficient of the fitted curves as well as the gas/oil ratio at the bubble point pressures is predicted.

A perfect equation for fitting this curve is given in equation 4.5. This equation resembles that of oil viscosity between dead oil viscosity and viscosity at bubble point pressure. The main difference is that the gas/oil ratio which is equivalent to dead oil viscosity is zero. In a nutshell, the minimum gas/oil ratio is zero.

A sample of a fitted gas/oil ratio curve is shown in Figure 4.4.

$$R_s = R_{sb} \left( \frac{P - P_d}{P_b - P_d} \right)^\tau \quad (4.5)$$

where  $\tau$  is the fitting gas/oil ratio curve coefficient.

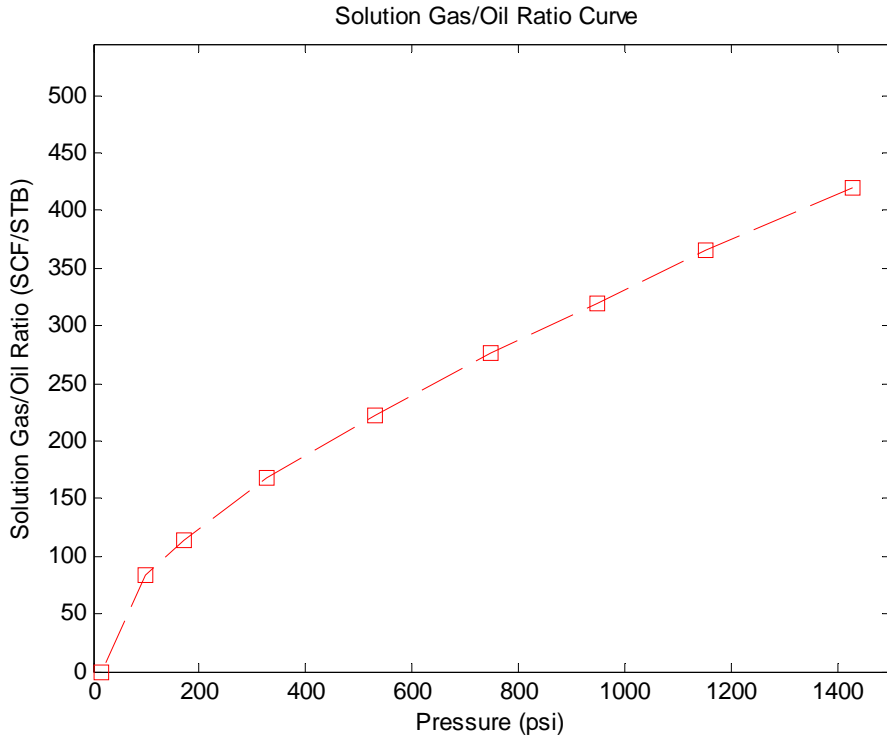


Figure 4.3: A Typical Gas/Oil Ratio Curve

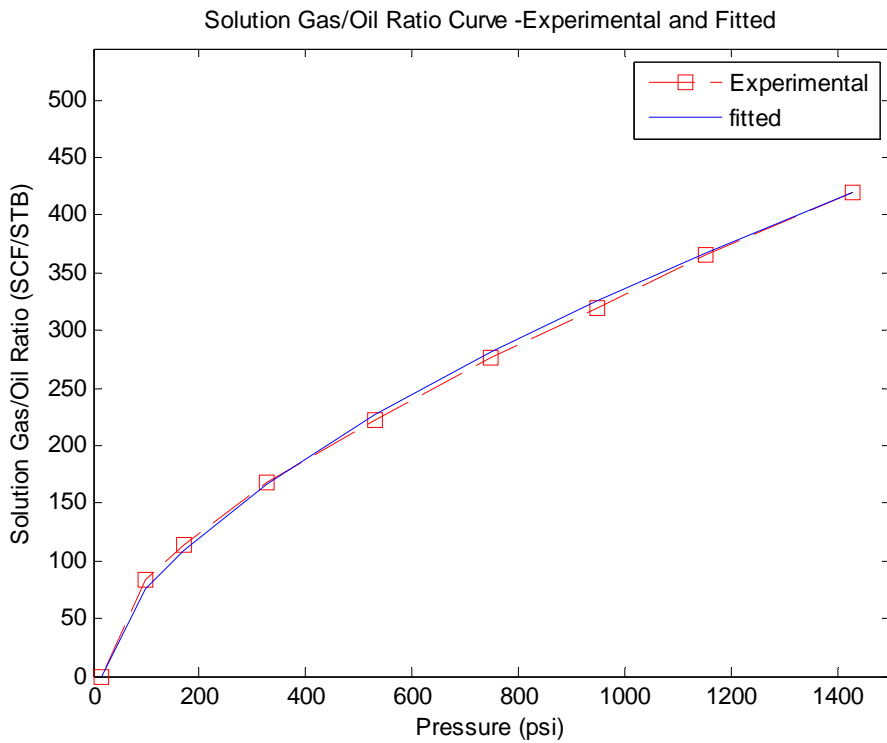


Figure 4.4: A fitted Gas/Oil Ratio Curve

The required parameters to be predicted are two: the fitting coefficient,  $\tau$ , and the gas/oil ratio at bubble point pressure ( $R_{sb}$ ). As we had under viscosity experimentation, there are two types of data sets, namely: data sets A and C. The data set C consists of gas/oil ratio-pressure measurements which are used to generate the curves and hence the fitting coefficients. Similar to viscosity curve prediction, data set A is used here also as predictors, with 70 wells for training and the remaining 29 for testing.

Table 4.2 Distribution of the Fitting Coefficients

Parameter	Max. Value	Min. Value
$\alpha$	1.90E-04	1.49E-05
$\beta$	0.924	0.1331
$\tau$	0.922647	0.43773

#### 4.4. Implementation of the Soft Computing Techniques

In what follows, we briefly describe the basis of all the designed Soft Computing Techniques: ANN, SVM, FN and ANFIS, and the two hybrid models: DE+ANN and GA+ANFIS. All these are put into three categories:

- i) ANN and DE+ANN
- ii) SVM and FN
- iii) ANFIS and GA+ANFIS

All in all, there are just five variables  $\alpha, \beta, \mu_{ob}, \tau$  and  $R_{sb}$  to be predicted in order to predict both viscosity and gas/oil ratio curves (see sections 4.3.1 and 4.3.2).

#### 4.4.1. ANN and DE+ANN Implementation

One of the commonly emphasised drawbacks of ANN is the unrestricted trial and error method used in selecting the number of hidden layers and their neurons that can correctly map and capture the non-linear relationship between the inputs and output(s). At the same time, Hornik et al [36] suggested that a sufficient number of neurons in a hidden layer can approximate any function to any degree of accuracy. However, such a system may not be stable. In developing the ANN (feedforward neural network, FFNN) models for the five aforementioned predicting variables, a number of trials were made viz: selecting the number of hidden layers, number of neurons in each hidden layer and the training algorithm. For  $\mu_{ob}$  and  $R_{sb}$ , we eventually used two hidden layers with thirteen and six neurons respectively. Hence, we have 12-12-6-1 ANN structure, (12 input neurons, 12 neurons in the first hidden layer, 6 neurons in the second hidden layer and 1 output neuron), in each case. For the three fitting variables, we used 12-12-5-1 ANN (FFNN) architecture. In all cases, tangent sigmoid transfer function and Levenberg-Marquardt training optimization were eventually used, and the best network out of 1000 runs in each case was taken. The architectures of the ANN above were retained during the implementation of DE+ANN. As earlier said, a variant of Differential Evolution (DE) called “traindiffevol”, (see [37]), was used for building the DE+ANN models.

## 4.4.2. SVR and FN Implementation

### A. Support Vector Regression (Support Vector Machines Framework)

Support Vector Regression (SVR), a variant of SVM for regression, is a robust machine learning algorithm which is unique, optimal and unlikely to generate local minima. Cortes and Vapnik [14] first introduced it as an advancement to neural network. We have highlighted the conventional implementation of SVR in sub-section 3.2.1.:

For the five predicting variables, the selected optimal relevant variables are stated as follows.

1.  $\alpha$ : C =10000; lambda = 1e-7; epsilon = 0.09; kerneloption =0.9; kernel= 'poly';  
verbose=1.

2.  $\beta$ : C =60; lambda = 1e-7; epsilon = 0.08; kerneloption =0.8; kernel= 'poly';  
verbose=1.

3.  $\mu_{ob}$ : C= 40000; lambda = 1e-7; epsilon = 0.001; kerneloption =0.994;  
kernel= 'gaussian' ; verbose=1.

4.  $\tau$ : C=100000; lambda = 1e-7;epsilon = 0.001; kerneloption =2.8;  
kernel='poly'.

5.  $R_{sb}$ : C= 500000; lambda = 1e-7;epsilon = 0.001; kerneloption =0.12;  
kernel='gaussian'.

After selection of a kernel, the other highly influential parameters in any SVR model based on observation are “C” and “kerneloption”. For ‘poly’ kernel, kerneloption denotes the degree of the kernel polynomial while it denotes kernel bandwidth for ‘gaussian’. “C” is the trade-off between achieving minimal training error and complexity of the model.

## B. Functional Networks

In functional networks, neural functions are to be learned instead of weights. To learn these neural functions, a set of linearly independent functions have to be used. These are called basis functions. Possible basis functions are: polynomial, exponential, Fourier and logarithm functions or their combinations. Selection of the basis function along with the possible learning method is essential in developing FN model .Section 3.3 gives detailed explanation on this. Details about the basis function that gave best results in each of the five cases are given in what follows. As explained in section 3.3, Minimum Description Length Principle (MDLP) was used to optimize the network and select the best model. The output is given by:

$$y = f_1(x_1) + f_2(x_2) + f_3(x_3) + \dots + f_{12}(x_{12}) \quad (4.6)$$

A. For  $\alpha$ , polynomial family of degree 3 was used and  $f_1 \dots f_{12}$  are

$$f_1(x_1) = -0.78629 - 0.00026x_1 + 0.00012x_1^2 - 4.5 \times 10^{-5}x_1^3;$$

$$f_2(x_2) = -0.0002x_2; \quad f_3(x_3) = -0.00019x_3 + 7.6 \times 10^{-7}x_3^2;$$

$$f_4(x_4) = -0.00021x_4 + 2.29 \times 10^{-7}x_4^2;$$

$$f_5(x_5) = -1.7 \times 10^{-5}x_5^2 + 4.93 \times 10^{-7}x_5^3;$$

$$f_6(x_6) = -4.7 \times 10^{-6}x_6^2 + 3.72 \times 10^{-8}x_6^3;$$

$$f_7(x_7) = 2.58 \times 10^{-4}x_7 - 5.3 \times 10^{-6}x_7^2$$

$$f_8(x_8) = 0.03466x_8 - 0.0005x_8^2 + 2.4 \times 10^{-6}x_8^2;$$

$$f_9(x_9) = -1.7 \times 10^{-7}x_9 + 4.38 \times 10^{-11}x_9^2;$$

$$f_{10}(x_{10}) = 5.44 \times 10^{-5}x_{10} - 8.7 \times 10^{-7}x_{10}^2$$

$$f_{11}(x_{11}) = -4.2 \times 10^{-5}x_{11} + 2.25 \times 10^{-7}x_{11}^2 - 4 \times 10^{-10}x_{11}^3$$

$$f_{12}(x_{12}) = -0.00011x_{12} + 3.3 \times 10^{-5}x_{12}^2 - 2.3 \times 10^{-6}x_{12}^3;$$

B. For  $\beta$ , polynomial family of degree 3 gave the best result and  $f_1 \dots f_{12}$  are

$$\begin{aligned}
 f_1(x_1) &= -119.266 + 1.37266x_1^2 - 0.4728x_1^3; f_2(x_2) = 0.80142x_2 \\
 f_3(x_3) &= 0.752408x_3 + 0.02248x_3 - 0.00108x_3^2 \\
 f_4(x_4) &= 0.80746x_4 + 0.000544x_4^2 \\
 f_5(x_5) &= 1.83579x_5 - 0.09053x_5^2 + 0.002752x_5^3 \\
 f_6(x_6) &= 1.72097x_6 - 0.02352x_6^2 + 0.000207x_6^3 \\
 f_7(x_7) &= 0.01923x_7^2 - 0.00052x_7^3; f_8(x_8) = 0 \\
 f_9(x_9) &= 0.000295x_9 - 1.6 \times 10^{-6}x_9^2 + 2.67 \times 10^{-10}x_9^3 \\
 f_{10}(x_{10}) &= 3.2953x_{10} - 0.09306x_{10}^2 + 0.000853x_{10}^3 \\
 f_{11}(x_{11}) &= -0.33375x_{11} + 0.00175x_{11}^2 - 3 \times 10^{-6}x_{11}^3 \\
 f_{12}(x_{12}) &= -0.5895x_{12} + 0.070413x_{12}^2
 \end{aligned}$$

C. For  $\mu_{ob}$  polynomial family of degree 3 gave the best result and  $f_1 \dots f_{12}$  are

$$\begin{aligned}
 f_1(x_1) &= -25.5869x_1 \\
 f_2(x_2) &= 0.162028x_2 - 0.105347x_2^2 + 0.004261x_2^3 \\
 f_3(x_3) &= 0.032553x_3 - 0.00362x_3^2; f_4(x_4) = 0; \\
 f_5(x_5) &= 0; f_6(x_6) = 0; \\
 f_7(x_7) &= 1.9566x_7 - 0.07975x_7^2 + 0.001032x_7^3 \\
 f_8(x_8) &= 0; f_9(x_9) = -3.9 \times 10^{-7}x_9^2 + 8.62x_9^3; \\
 f_{10}(x_{10}) &= 1.70833x_{10} - 0.04799x_{10}^2 + 0.000443x_{10}^3 \\
 f_{11}(x_{11}) &= -0.13176x_{11} + 0.000663x_{11}^2 - 1.1 \times 10^{-6}x_{11}^3 \\
 f_{12}(x_{12}) &= 0.152996x_{12} + 0.01277x_{12}^2
 \end{aligned}$$

D. For  $\tau$  logarithm family gave the best result and  $f_1 \dots f_{12}$  are

$$\begin{aligned}
 f_1(x_1) &= -14589.5 - 9.80634 \log(x_1 + 2) + 15.5367 \log(x_1 + 3) \\
 f_2(x_2) &= 107.4192 \log(x_2 + 3) - 331.197 \log(x_2 + 4) + 239.503 \log(x_2 + 5) \\
 f_3(x_3) &= 77.258 \log(x_3 + 3) - 251.037 \log(x_3 + 4) + 188.3491 \log(x_3 + 5) \\
 f_4(x_4) &= -903.557 \log(x_4 + 3) + 949.7861 \log(x_4 + 4) \\
 f_5(x_5) &= 1.94 \times 10^5 \log(x_5 + 2) - 7.1 \times 10^5 \log(x_5 + 3) \\
 &\quad + 8.7 \times 10^5 \log(x_5 + 4) - 349768 \log(x_5 + 5) \\
 f_6(x_6) &= 3105700 \log(x_6 + 2) - 10000000 \log(x_6 + 3) \\
 &\quad + 10858916 \log(x_6 + 4) - 3895946 \log(x_6 + 5) \\
 f_7(x_7) &= -2.3 \times 10^7 \log(x_7 + 2) + 7.59 \times 10^7 \log(x_7 + 3) \\
 &\quad - 8.5 \times 10^7 \log(x_7 + 4) + 3.13 \times 10^7 \log(x_7 + 5) \\
 f_8(x_8) &= 1.2 \times 10^4 \log(x_8 + 2) - 12235.2 \log(x_8 + 4) \\
 f_9(x_9) &= 1.05 \times 10^9 \log(x_9 + 2) - 3.2 \times 10^9 \log(x_9 + 3) \\
 &\quad + 3.16 \times 10^9 \log(x_9 + 4) - 1.1 \times 10^9 \log(x_9 + 5) \\
 f_{10}(x_{10}) &= -1.2502 \log(x_{10} + 4); \quad f_{11}(x) = 0 \\
 f_{12}(x_{12}) &= -12.0186 \log(x_{12} + 2) + 14.2783 \log(x_{12} + 3)
 \end{aligned}$$

E. For  $R_{sb}$  logarithm family gave the best result and  $f_1 \dots f_{12}$  are

$$\begin{aligned}
 f_1(x_1) &= 251353 + 3409.847 \log(x_1 + 2) - 6184.59 \log(x_1 + 3); \\
 f_2(x_2) &= -17732.5 \log(x_2 + 2) + 66166.75 \log(x_2 + 3) - 54857.3 \log(x_2 + 4) \\
 f_3(x_3) &= 10315.9 \log(x_3 + 3) - 14498.5 \log(x_3 + 4); \\
 f_4(x_4) &= -1.3 \times 10^7 \log(x_4 + 2) + 2.68 \times 10^7 \log(x_4 + 3) - 1.4 \times 10^3 \log(x_4 + 4); \\
 f_5(x_5) &= 2.63 \times 10^6 \log(x_5 + 2) - 5.9 \times 10^6 \log(x_5 + 3) + 3.3144 \times 10^6 \log(x_5 + 4); \\
 f_6(x_6) &= -2.2 \times 10^7 \log(x_6 + 2) + 4.7 \times 10^7 \log(x_6 + 3) - 2.5 \times 10^7 \log(x_6 + 4); \\
 f_7(x_7) &= 1.65 \times 10^6 \log(x_7 + 3) - 1.71 \times 10^6 \log(x_7 + 4); \\
 f_8(x_8) &= 2056.883 \log(x_8 + 4);
 \end{aligned}$$



$$f_9(x_9) = 1.47 \times 10^9 \log(x_9 + 2) - 2.9 \times 10^9 \log(x_9 + 3) + 1.48 \times 10^9 \log(x_9 + 4);$$

$$f_{10}(x_{10}) = -5 \times 10^5 \log(x_{10} + 3) + 5.119 \times 10^5 \log(x_{10} + 4); \quad f_{11}(x_{11}) = 0;$$

$$f_{12}(x_{12}) = 84618.57 \log(x_{12} + 2) - 246973 \log(x_{12} + 3) + 169459 \log(x_{12} + 4)$$

It is noteworthy that in some cases some functions are zero, this means that the corresponding input to that node does not really affect the predicting output at that instance.

#### 4.4.3. ANFIS and GA+ANFIS Implementation

Rather than the traditional fuzzy system, ANFIS offers an adaptive fuzzy system where rules are learnt from the data. However, its implementation could result in generating a large set of rules. To reduce the number of rules that are generated during ANFIS implementation, different clustering methods have been proposed, e.g. subtractive clustering and fuzzy-C mean. Subtractive clustering is the most widely used clustering method in ANFIS implementation. Two important variables in the subtractive clustering are the clustering “radii”, (a vector whose length is equal to the number of columns in the data set and each radius has a value between 0 and 1), and “options” (with variables: quashFactor, acceptRatio, rejectRatio and Verbose). “Verbose” in the “options” is totally insignificant as it only relates to choice of information display during ANFIS execution. In our case, the dimension of radii is 1x13 since there are 12 predictors and one output. It should also be noted that each optimal radius is always in the neighbourhood of the centre of the radius, i.e. 0.5.

## A. ANFIS Modelling

The following are the values of “radii” and “options” chosen by trial and error method for predicting the five required variables:  $\alpha$ ,  $\beta$ ,  $\mu_b$ ,  $\tau$  and  $\gamma_b$

1.  $\alpha$ : radii=[0.5 0.4 0.5 0.3 0.3 0.5 0.3 0.4 0.3 0.3 0.3 0.3 0.25];  
options=[1.2 0.4 0.2 0]

2.  $\beta$ : radii= [0.5 0.38 0.3 0.3 0.3 0.423 0.3 0.388 0.3 0.3 0.3 0.3 0.25];  
options=[1.2 0.4 0.2 0]

3.  $\mu_{ob}$ : radii= [0.5 0.38 0.3 0.5 0.3 0.403 0.38 0.388 0.3 0.3 0.3 0.3 0.25];  
options=[1.0 0.44 0.125 0]

4.  $\tau$ : radii=[0.5 0.4 0.4 0.5 0.4 0.4 0.38 0.388 0.4 0.4 0.3 0.45 0.45]  
options=[1.0 0.4 0.12 0]

5.  $R_{sb}$  : radii=[0.5 0.4 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.25]  
options=[1.2 0.4 0.2 0]

Based on our observation, “radii” and “quashfactor” are the most influential out of all parameters to be selected for ANFIS with subtractive modelling. Appropriate selection of these two parameters easily prevents overfitting and underfitting.

## B. GA+ANFIS Modelling

To improve the performance of ANFIS, a hybrid ANFIS model called GA+ANFIS has been built and implemented for predicting all the five variables. It is obvious that trial and error searching for 13(or generally n) parameters in the radii will not give optimal results. In this regard, we introduced intelligent search using GA for finding the optimal radii in the solution space. The step-by-step implementation of GA+ANFIS is given as follows.

- Initialize the population size,  $N$ , and the termination criterion

NOTE: The termination is either maximum number of generation or the error goal.

- Generate randomly  $N$  number of *radii*,  $radii=[r_1, r_2, \dots, r_n]$  in the solution space

NOTE: Here radii's are synonymous with chromosomes. Each population now has  $(N \times n)$  chromosomes.

- Evaluate the whole population in the generation through ANFIS implementation, using root mean squared error of the testing data as the criterion.
- Select the suitable chromosomes from the population for genetic operations (Tournament selection)
- Perform crossover and mutation of selected chromosomes to generate child population approximately equal to the parent population size and evaluate ANFIS testing output using root mean squared error of the testing phase.
- Select approximately  $N$  best individuals from the parent and child populations (elitism)
- These new  $N$  individuals form the parent population for the next generation
- Repeat the process until the termination criterion is met.

The results for GA+ANFIS show a significant improvement for both viscosity and Gas/Oil ratio predictions than when trial-and-error selection method for the *radii* was used.

## CHAPTER FIVE

### EXPERIMENTAL RESULTS AND DISCUSSION

#### 5.1. Introduction

In this study, models based on four independent soft computing techniques: ANN, SVR and FN, and two hybrid frameworks, DE+ANN and GA+ANFIS, have been developed. In what follows, we present and compare the results based on categorization during the implementation in section 4.4. The criteria for prediction performance evaluation are:

- (i) The plots of the experimental and the predicted curves. Only few plots are shown here for comparison.
- (ii) The root means square errors (RMSE) of the training and testing wells.
- (iii) The average absolute percent relative error (AAPRE) of the training and testing wells.

The formulae for the two statistical measures mentioned above are given as follows:

#### 1. Root mean Square Error

$$RMSE = \sqrt{\frac{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \dots + (x_n - y_n)^2}{n}} \quad (5.1)$$

#### 2. Average absolute percent relative error

$$E_i = \left( \frac{Y_i - X_i}{Y_i} \right) \times 100 ; \quad i = 1, 2, 3, \dots, n$$

$$AAPRE = \frac{1}{n} \sum_i^n |E_i| \quad (5.2)$$

Where  $x$ 's are the predicted values and  $y$ 's are the actual values and “ $n$ ” is the total number of data points in all the training wells (70) or testing wells (29). A good model should have low RMSE and AAPRE values.

## 5.2. Results and Discussion for Viscosity Curve Prediction

Statistical description of the data sets used for training our models is shown in Table 4.1. The first twelve variables are the predictors while the thirteenth variable (viscosity at bubble point pressure) is the predicted. Also, other variables ( $\alpha$  and  $\beta$ ) from the fitting curves were also predicted using the first twelve variables. We used 70% of the data for training and 30% for testing. Once again, each data point corresponds to the properties of an individual oil well.

### 5.2.1. ANN and DE+ANN for Viscosity Curve Prediction

The models for ANN and DE+ANN as described in section 4.4.1 were implemented. Sample predicted plots for training and testing wells are shown in Figures 5.1 through 5.6. The predicted viscosity curves from DE+ANN have better matching with the experimental curves than those of ANN. Also, from Table 5.1, the RMSE and APPRE for DE+ANN predictions are 0.07681 and 8.982% respectively for testing wells, while the corresponding RMSE and AAPRE for ANN predictions are 0.08712 and 10.2457% respectively. Since DE+ANN predictions give lower RMSE and AAPRE than ANN, its overall performance is better than ANN in predicting viscosity curves. Though the performance in the training phase is not as important as the testing phase, DE+ANN overall performance is also better than that of ANN during training because of its lower RMSE and AAPRE. This shows consistency in the results of DE+ANN. The predicted curves from the two frameworks show good matching with the experimental curves for training wells with some deviations for some testing wells. Likewise, Table 5.2 shows

the predicted parameters of the viscosity curves in Figures 5.1 through 5.6 based on ANN and DE+ANN models.

Table 5.1: Statistical Performance Measures of ANN and DE+ANN Models for Viscosity Curve Prediction

MODEL	ANN		DE+ANN	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	0.08664	8.33945	0.07409	6.77273
TESTING	0.08712	10.24569	0.07681	8.98202

Table 5.2: Sample Predicted Viscosity Curve Parameters by ANN and DE+ANN Models

	ACTUAL	ANN	DE+ANN
TRAINING: $\alpha$	7.19E-05	5.7914E-05	5.79E-05
	1.09E-4	1.08E-4	1.08E-4
	8.59E-05	5.6E-05	5.6E-05
$\beta$	0.6688	0.50256	0.50256
	0.6201	0.5881	0.5981
	0.923938	0.6266	0.7265
$\mu_{ob}$	0.69	0.6628	0.7125
	0.9	0.8643	0.8986
	0.77	0.6356	0.6811
TESTING: $\alpha$	3.87E-05	4.32E-05	4.32E-05
	3.02E-05	4.85E-05	4.85E-05
	5.81E-05	5.11E-05	5.11E-05
$\beta$	0.3388	0.4889	0.4889
	0.3352	0.4594	0.4594
	0.4121	0.4785	0.4785
$\mu_{ob}$	0.58	0.5228	0.5434
	0.54	0.5131	0.5629
	0.719	0.5753	0.6755

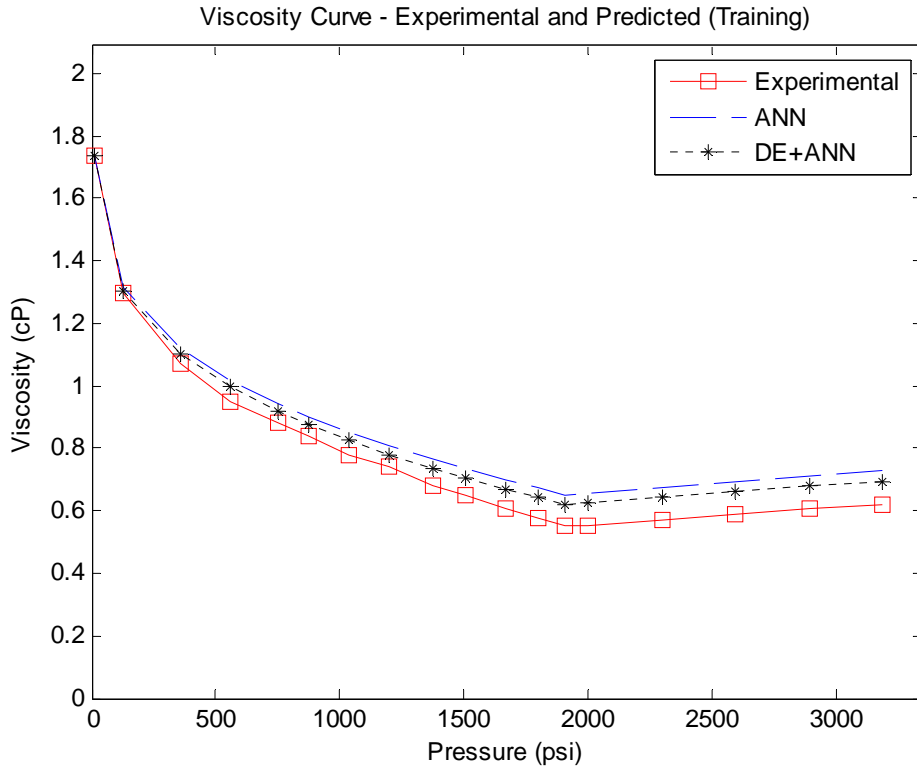


Figure 5.1: Viscosity vs Pressure Plot for Sample Well TR1

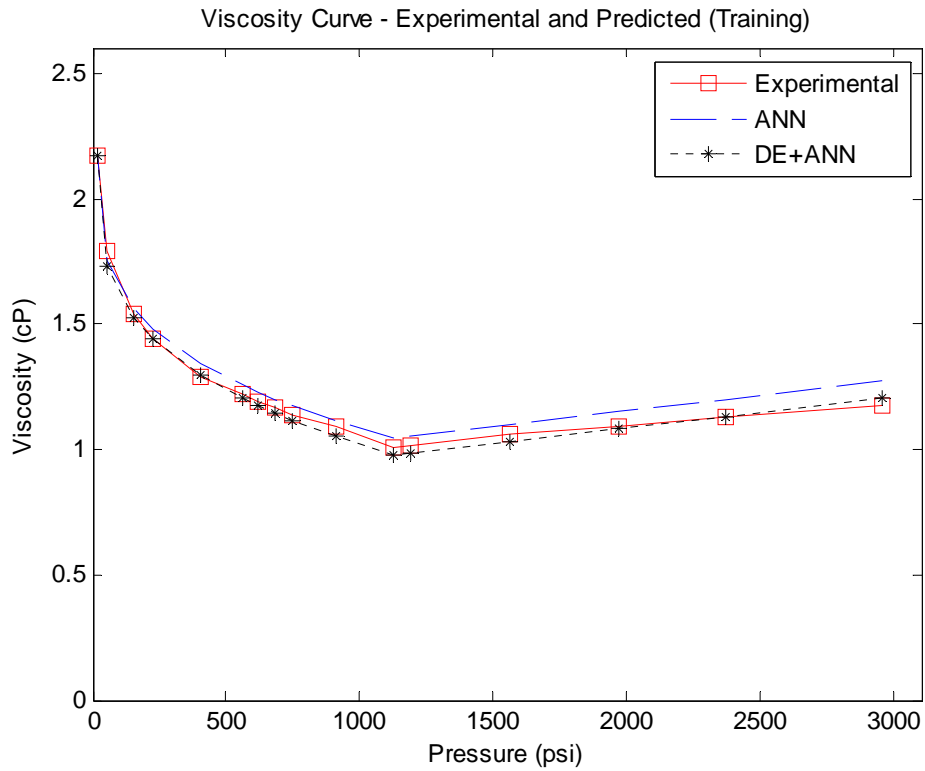


Figure 5.2: Viscosity vs Pressure Plot for Sample Well TR2

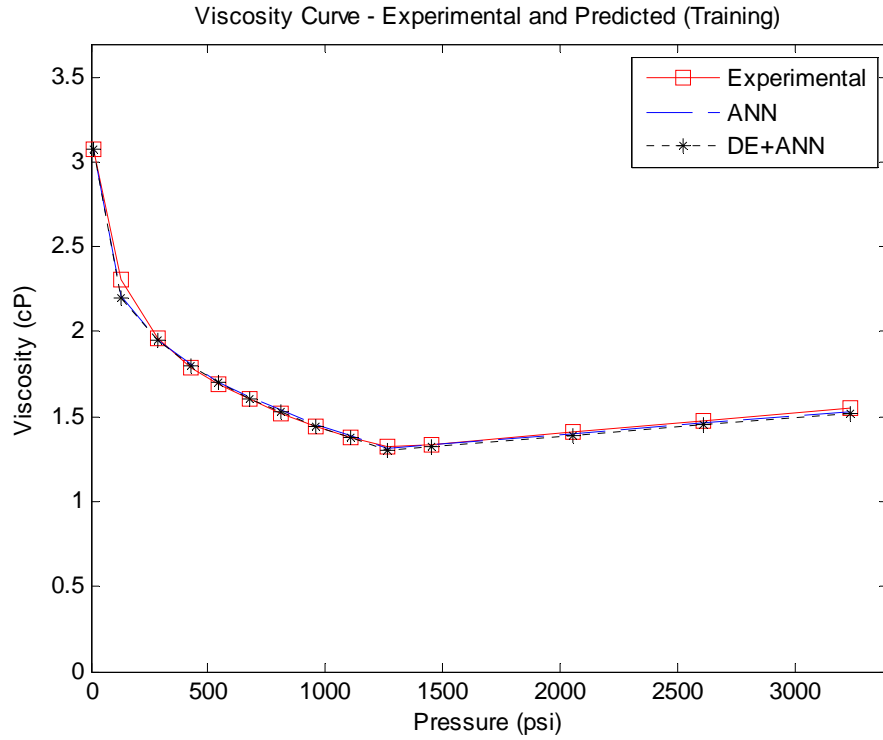


Figure 5.3: Viscosity vs Pressure Plot for Sample Well TR3

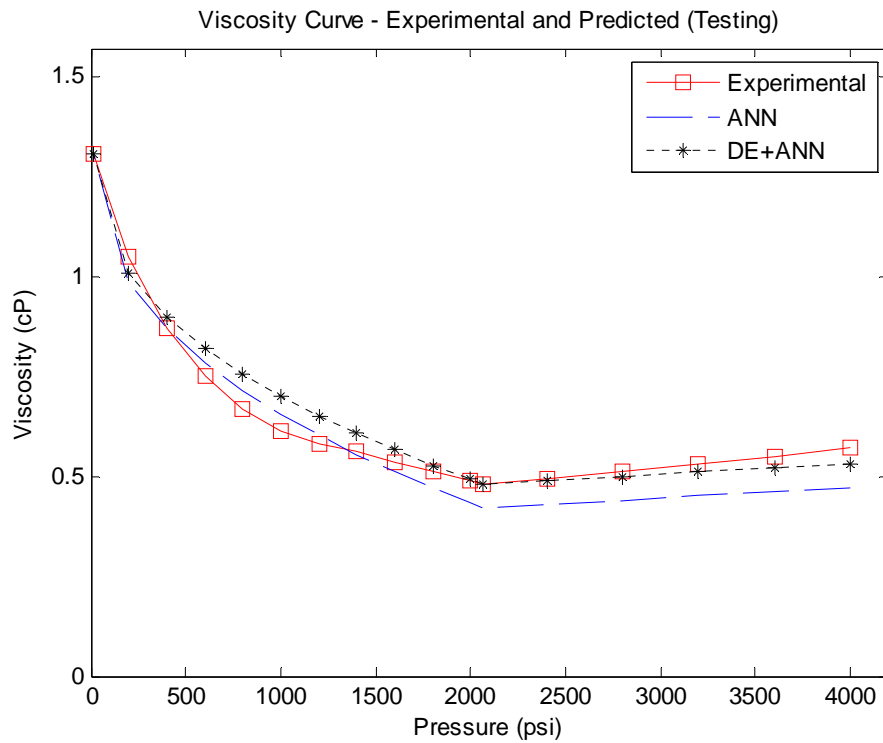


Figure 5.4: Viscosity vs Pressure Plot for Sample Well TS1



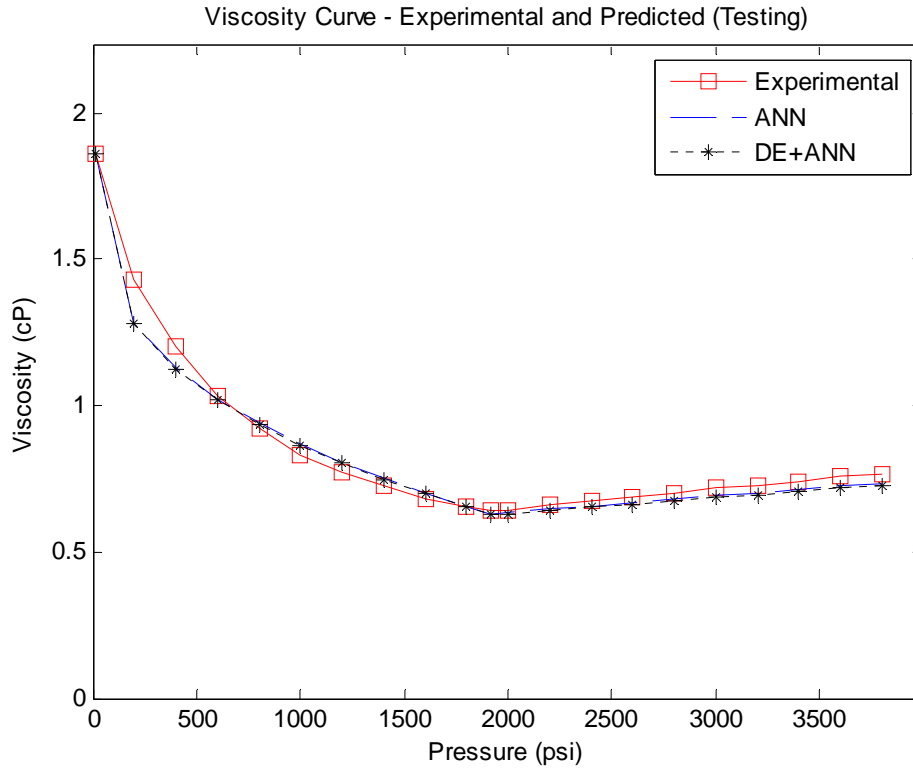


Figure 5.5: Viscosity vs Pressure Plot for Sample Well TS2

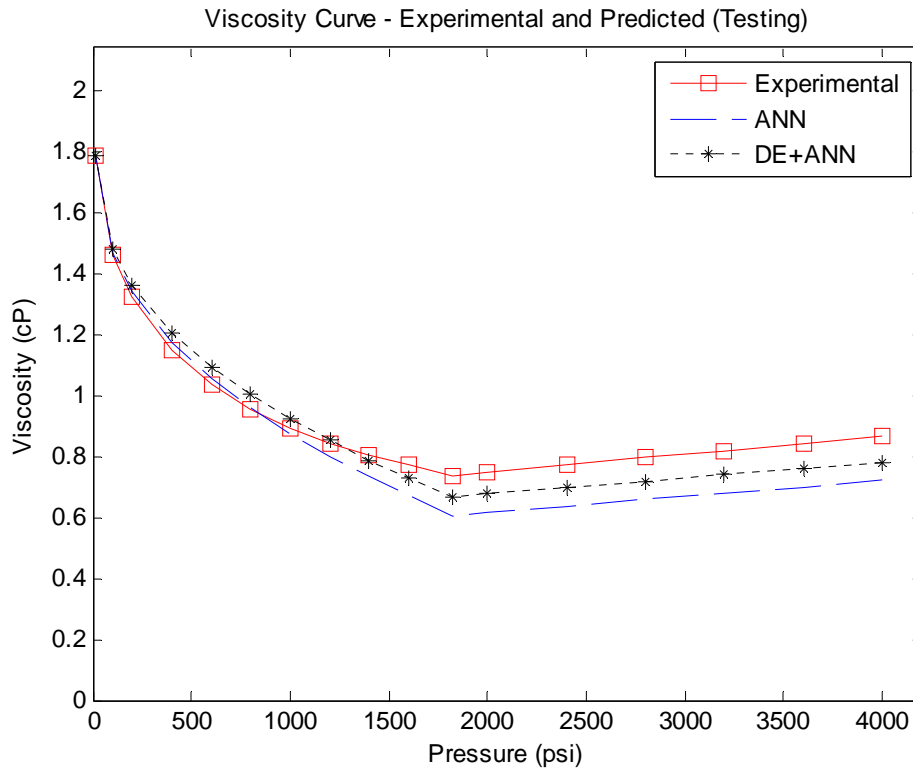


Figure 5.6: Viscosity vs Pressure Plot for Sample Well TS3

### 5.2.2. SVR and FN for Viscosity Curve Prediction

We implemented SVR and FN as described in section 4.4.2. Here also, sample predicted plots of training and testing wells from the two frameworks are shown in Figures 5.7 through 5.12. The statistical performance measures for the two frameworks are shown in Table 5.3. For this pair of techniques, the performance of both frameworks, SVR and FN, are very competitive. While FN performance is better than that of SVR in the training phase with lower RMSE and AAPRE, which are 0.06765 and 5.4% respectively, against those of SVR which are 0.07495 and 6.3953% respectively, SVR performance is very competitive with that of FN for the testing wells. For the testing phase, FN has lower AAPRE of 8.5514%, against that of SVR which is 8.5969%, while SVR has lower RMSE, 0.0765, against that of FN which is 0.07941. In essence, the results for these two frameworks are very competitive for viscosity curve prediction. The predicted curves from the two SC techniques show good matching with the experimental curves for both training and testing wells with little deviation in some testing wells. Likewise, Table 5.4 shows the predicted parameters of the viscosity curves in Figures 5.7 through 5.12 based on SVR and FN models

Table 5.3: Statistical Performance Measures of SVR and FN Models for Viscosity Curve Prediction

MODEL	SVR		FN	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	0.07495	6.3953	0.067648	5.400661
TESTING	0.07659	8.5969	0.079412	8.551437

Table 5.4: Sample Predicted Viscosity Curve Parameters by SVR and FN Models

	ACTUAL	SVR	FN
TRAINING: $\alpha$	7.19E-05	5.79E-05	5.89E-05
	1.09E-4	7.27E-05	8.07E-05
	8.59E-05	5.86E-05	5.71E-05
$\beta$	0.6688	0.626296	0.625281
	0.6201	0.67805	0.593434
	0.923938	0.833985	0.89791
$\mu_{ob}$	0.69	0.686544	0.71764
	0.9	0.782032	0.877111
	0.77	0.652283	0.658631
TESTING: $\alpha$	3.87E-05	4.88E-05	5.38E-05
	3.02E-05	4.57E-05	4.48E-05
	5.81E-05	5.10E-05	5.56E-05
$\beta$	0.3388	0.3386	0.3190
	0.3352	0.3426	0.3712
	0.4121	0.3781	0.4204
$\mu_{ob}$	0.58	0.5736	0.5549
	0.54	0.5378	0.5690
	0.719	0.6835	0.6767

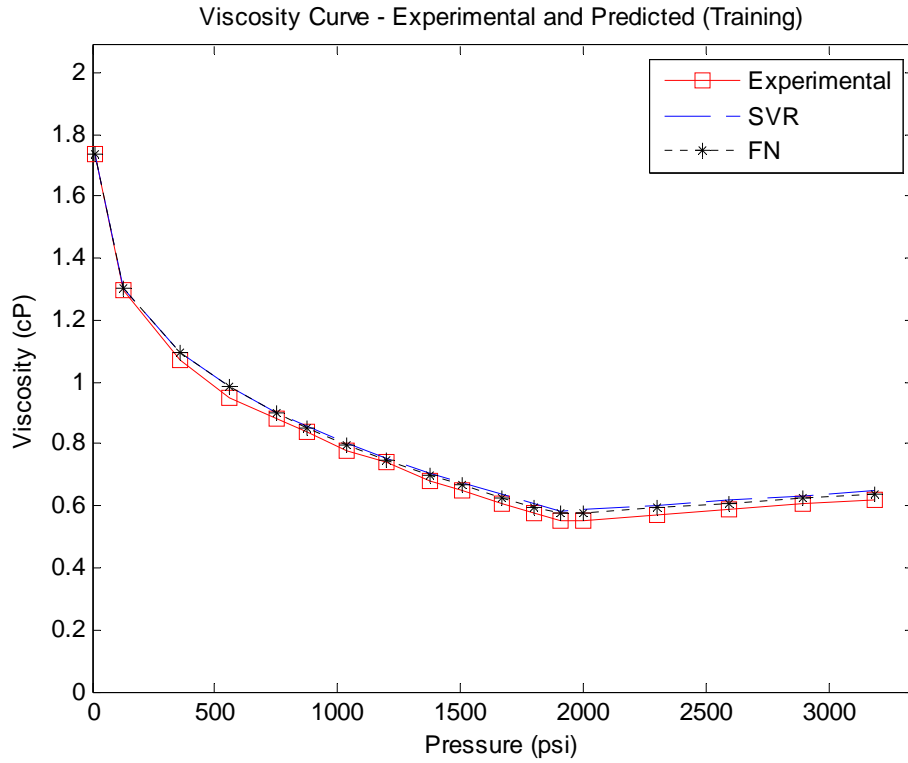


Figure 5.7: Viscosity vs Pressure Plot for Sample Well TR1

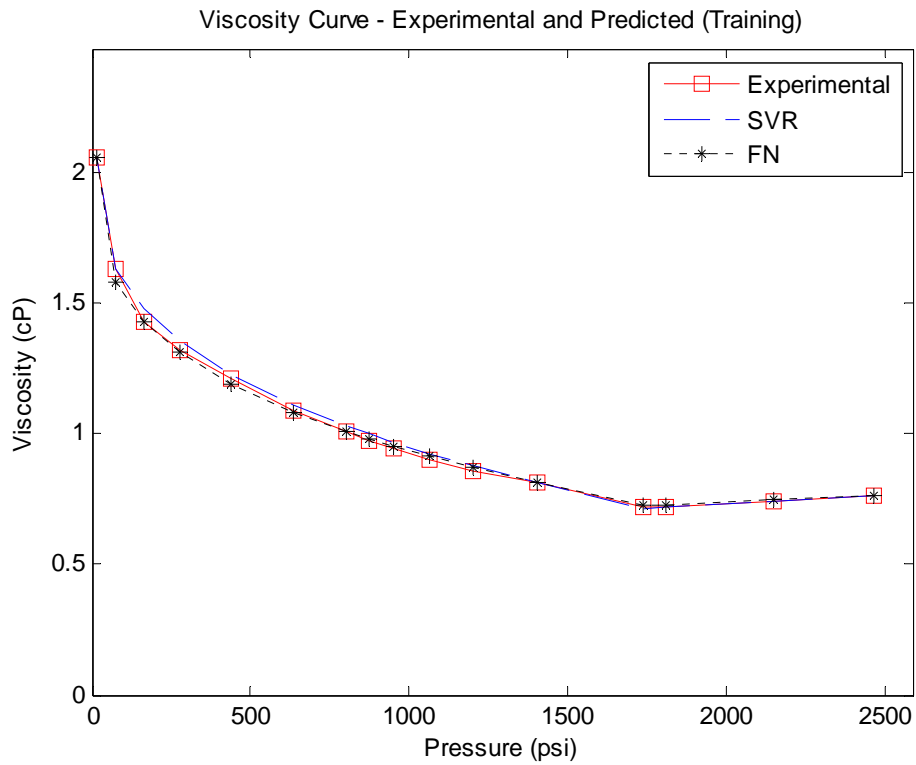


Figure 5.8: Viscosity vs Pressure Plot for Sample Well TR2

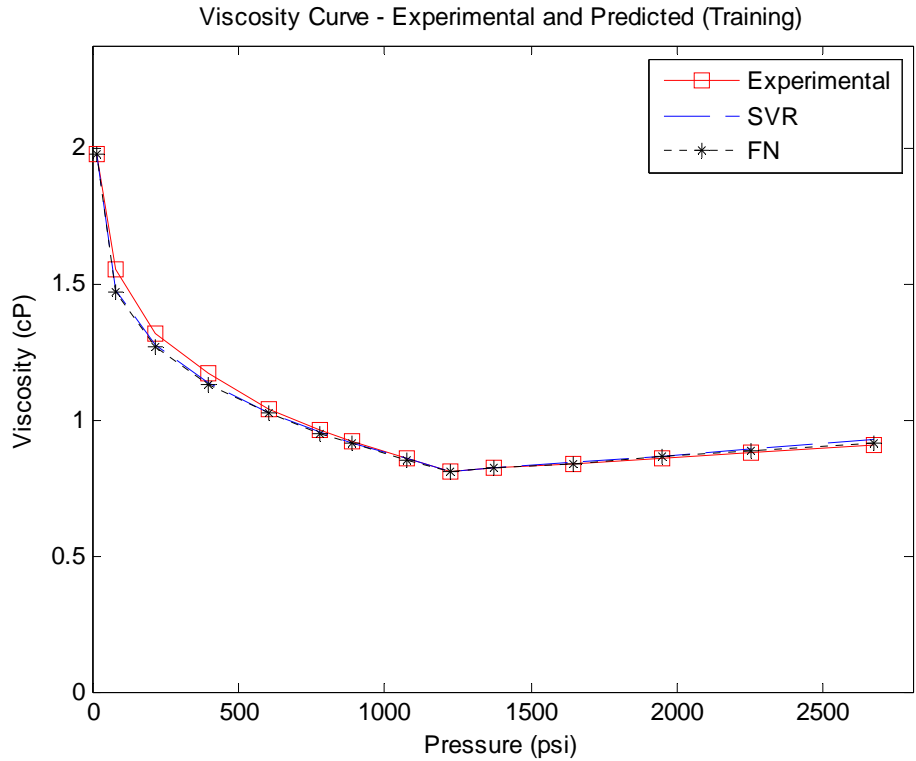


Figure 5.9: Viscosity vs Pressure Plot for Sample Well TR3

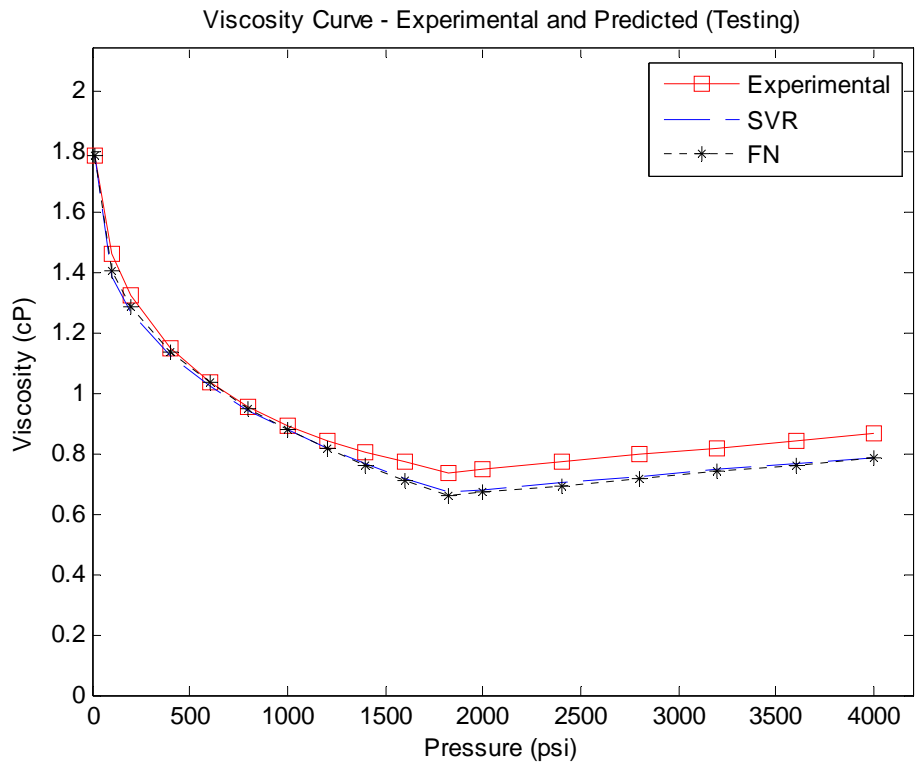


Figure 5.10: Viscosity vs Pressure Plot for Sample Well TS1

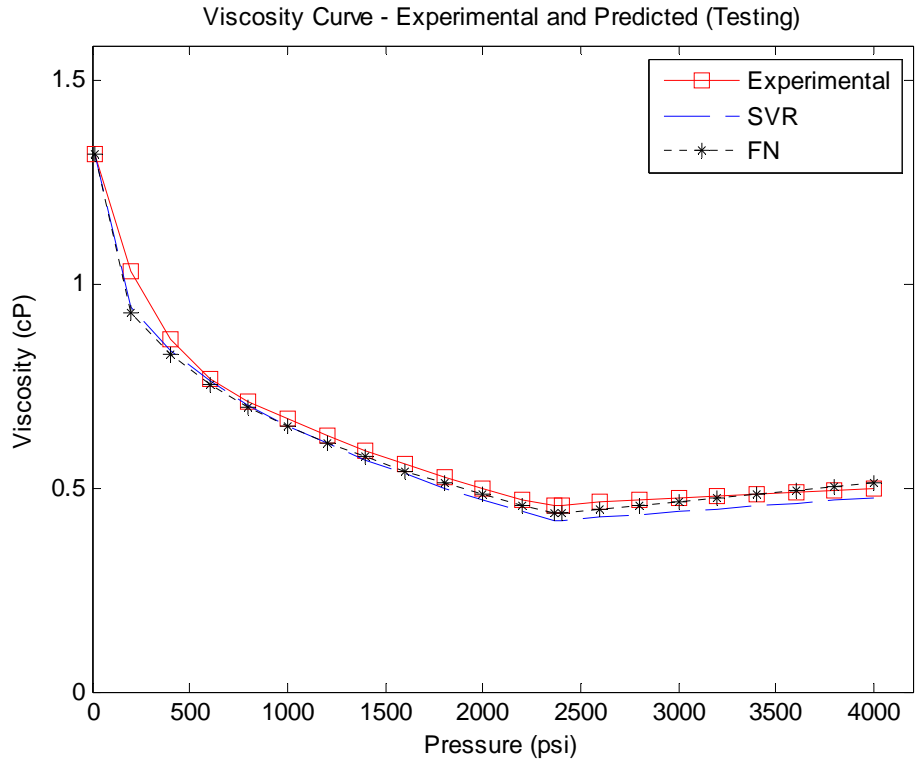


Figure 5.11: Viscosity vs Pressure Plot for Sample Well TS2

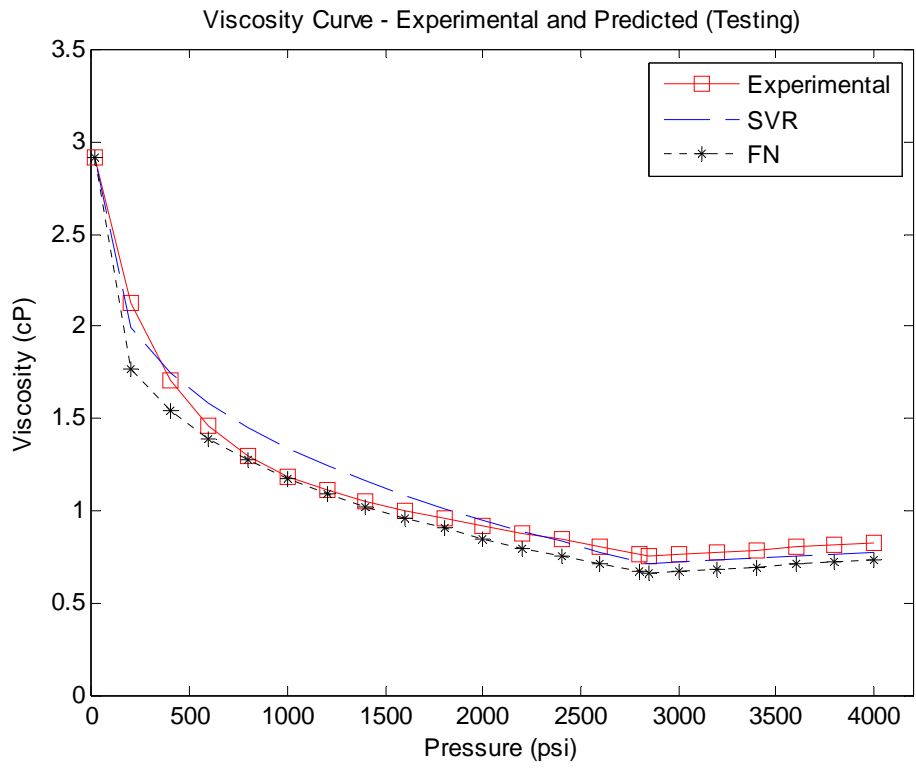


Figure 5.12: Viscosity vs Pressure Plot for Sample Well TS3

### 5.2.3. ANFIS and GA+ANFIS for Viscosity Curve Prediction

The implementation of these two frameworks have been highlighted in section 4.4.3. The aim of the developed GA+ANFIS framework is to improve the performance of independent ANFIS prediction which relies heavily on the right choice for subtractive clustering *radii* for the data set. Wrong choice of *radii* leads easily to over-fitting, giving good performance for training but very poor for testing, or even under-fitting. Following the same trend of the previous sections, sample predicted plots of the training and testing wells from the two frameworks are shown in Figures 5.13 through 5.18. The RMSE and AAPRE for the training and testing wells for the two SC techniques are given in Table 5.5.

Based on the three evaluation criteria, GA+ANFIS gives better performance than ANFIS. The RMSE and AAPRE for GA+ANFIS framework amount to 0.0686 and 7.7602% respectively for the testing wells, and the corresponding values for independent ANFIS are 0.08516 and 10.18525% respectively. These values of the RMSE and AAPRE for GA+ANFIS hybrid are far lower than those of ANFIS. Hence, GA+ANFIS hybrid gives better performance than ANFIS alone. This affirms the problem of possible over-fitting associated with the use of subtractive clustering in ANFIS implementation. In the case of GA+ ANFIS, GA has been used to search within the solution space and the implementation has avoided *radii*'s that could lead to local optimal. Though implementing GA with ANFIS, as we did in this case, to search for optimal *radii* is very slow, it enhances appropriate selection of *radii* and hence, improvement of the results. Even for the training wells, GA+ANFIS shows better performance than the ordinarily implemented ANFIS where trial-and-error methods have been used to choose the optimal

*radii*. Likewise, Table 5.7 shows the predicted parameters of the viscosity curves in Figures 5.13 through 5.18 based on ANFIS and GA+ANFIS models.

Table 5.5: Statistical Performance Measures of ANFIS and GA+ANFIS Models for Viscosity Curve Prediction

MODEL	ANFIS		GA+ANFIS	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	0.06177	5.221732	0.05241	4.551326
TESTING	0.08516	10.18525	0.068587	7.760236

Table 5.6: Sample Predicted Viscosity Curve Parameters by ANFIS and GA+ANFIS Models

	ACTUAL	ANFIS	GA+ANFIS
TRAINING: $\alpha$	7.19E-05	5.78E-05	5.97E-05
	1.09E-4	1.01E-4	1.04E-4
	8.59E-05	5.89E-05	5.48E-05
$\beta$	0.6688	0.6112	0.6716
	0.6201	0.6109	0.6124
	0.923938	0.6751	0.8963
$\mu_{ob}$	0.69	0.6923	0.7261
	0.9	0.9059	0.8843
	0.77	0.6472	0.7156
TESTING: $\alpha$	3.87E-05	4.66E-05	4.91E-05
	3.02E-05	4.3E-05	4.13E-05
	5.81E-05	4.47E-05	4.76E-05
$\beta$	0.3388	0.3081	0.3289
	0.3352	0.3442	0.3378
	0.4121	0.5201	0.3204
$\mu_{ob}$	0.58	0.5899	0.5835
	0.54	0.5613	0.5590
	0.719	0.6345	0.6823



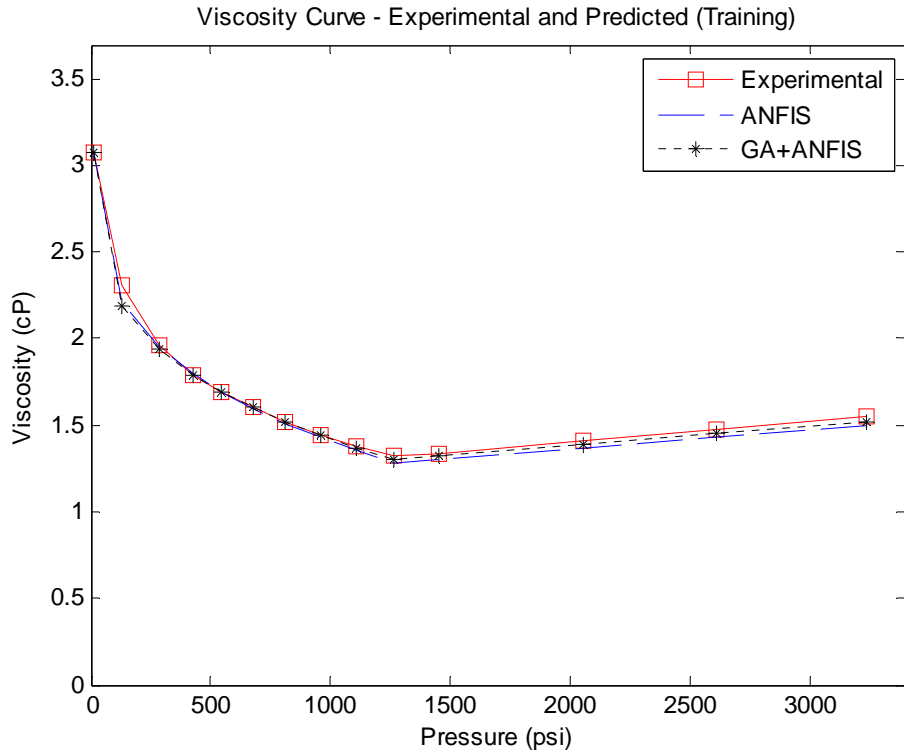


Figure 5.13: Viscosity vs Pressure Plot for Sample Well TR1

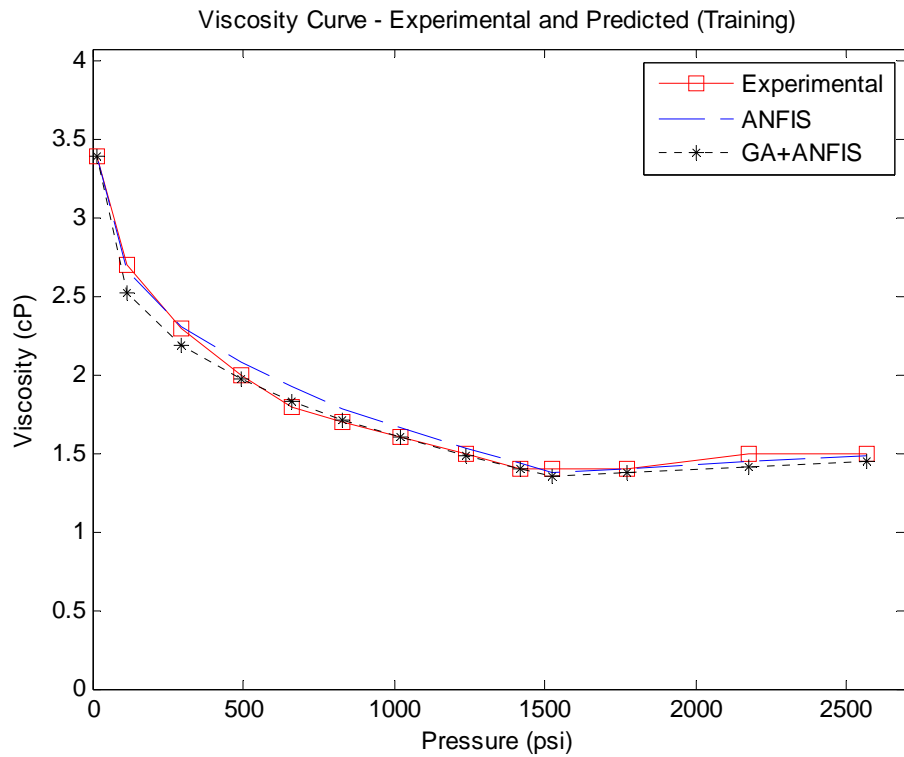


Figure 5.14: Viscosity vs Pressure Plot for Sample Well TR2

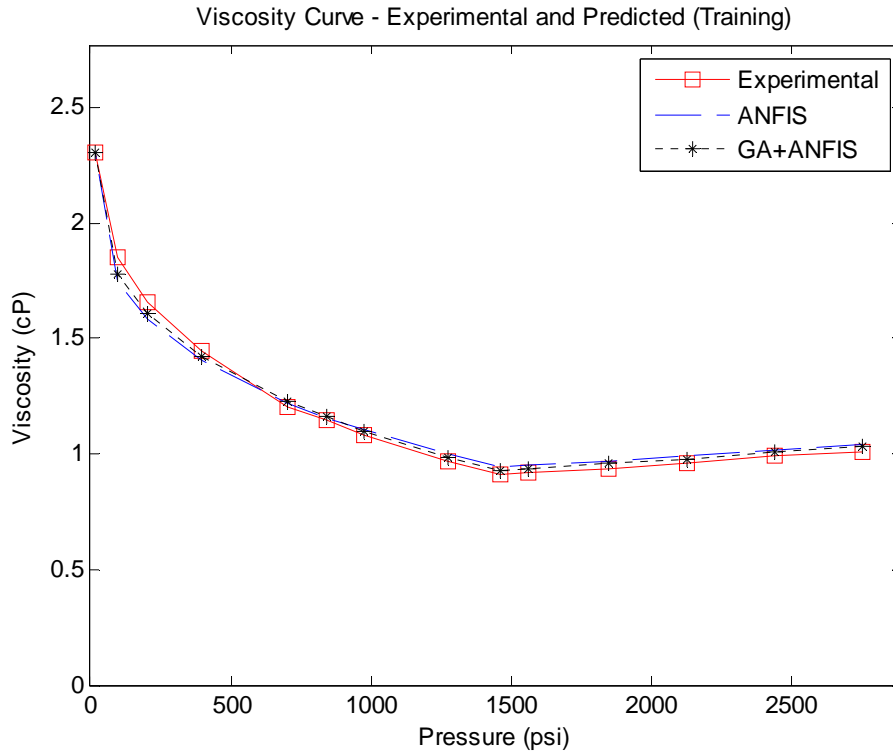


Figure 5.15: Viscosity vs Pressure Plot for Sample Well TR3

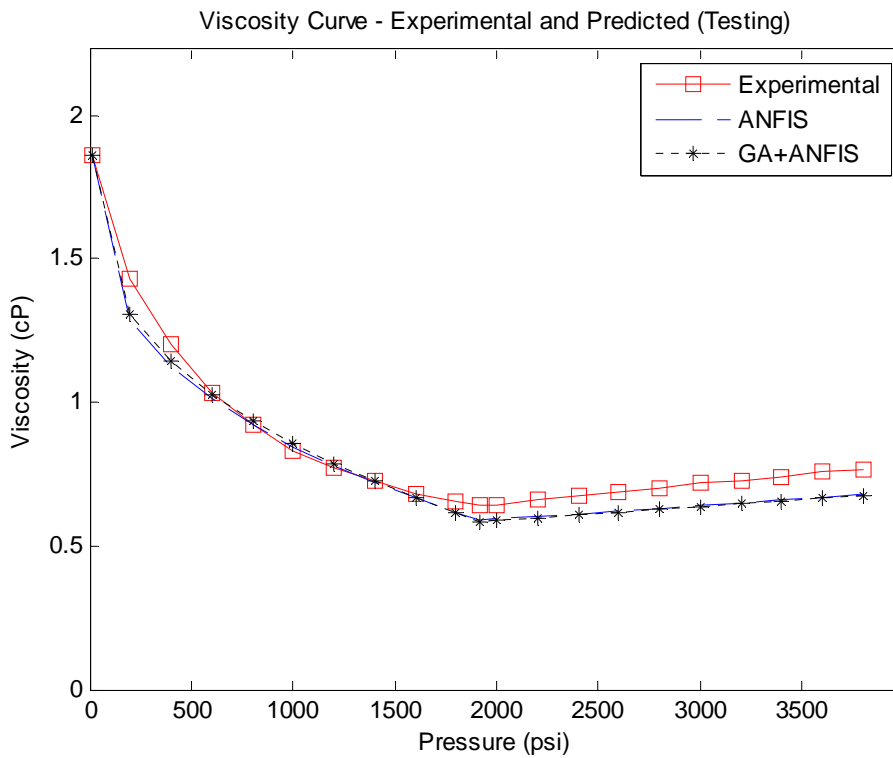


Figure 5.16: Viscosity vs Pressure Plot for Sample Well TS1

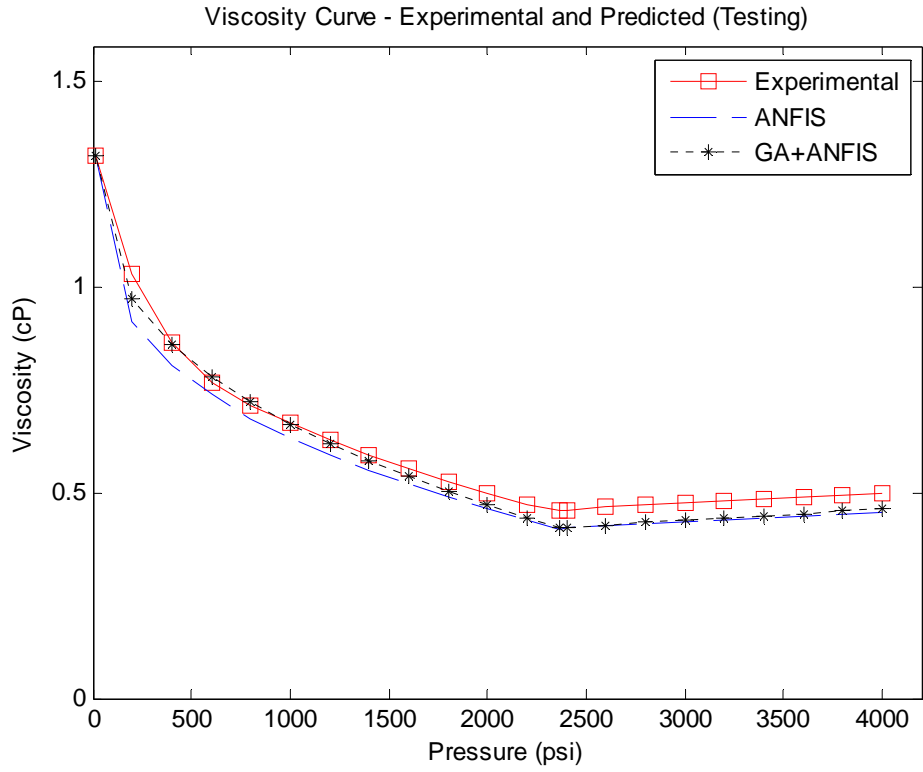


Figure 5.17: Viscosity vs Pressure Plot for Sample Well TS2

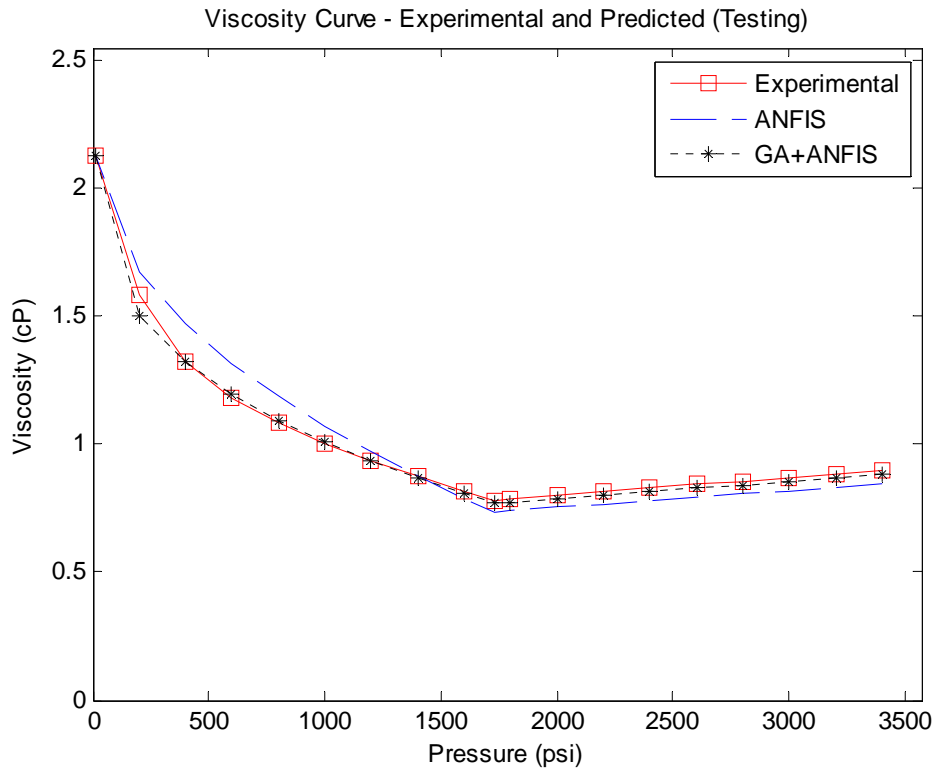


Figure 5.18: Viscosity vs Pressure Plot for Sample Well TS3

#### 5.2.4. Performances of All the Techniques for Viscosity Curve Prediction

In this section, we compare the results of all the implemented soft computing techniques for viscosity curve prediction based on statistical measures. Figures 5.19 through 5.22 show the comparisons between the statistical performances of all the implemented models for viscosity curve prediction.

From Figures 5.19 and 5.20, it is clearly seen that GA+ANFIS has the least RMSE for both training and testing. This means that GA+ANFIS gives the least overall RMSE for all the viscosity points in all the curves. Considering the testing phase for all other models (since testing is the main performance measure), FN performance is next to that of GA+ANFIS. Clearly, as it can be seen in Figure 5.20, ANN has the least performance with highest RMSE.

In the same vein, from Figures 5.21 and 5.22, GA+ANFIS has the least APPRE for both training and testing. Considering the testing case for all other models, SVR has the next least APPRE after GA+ANFIS.

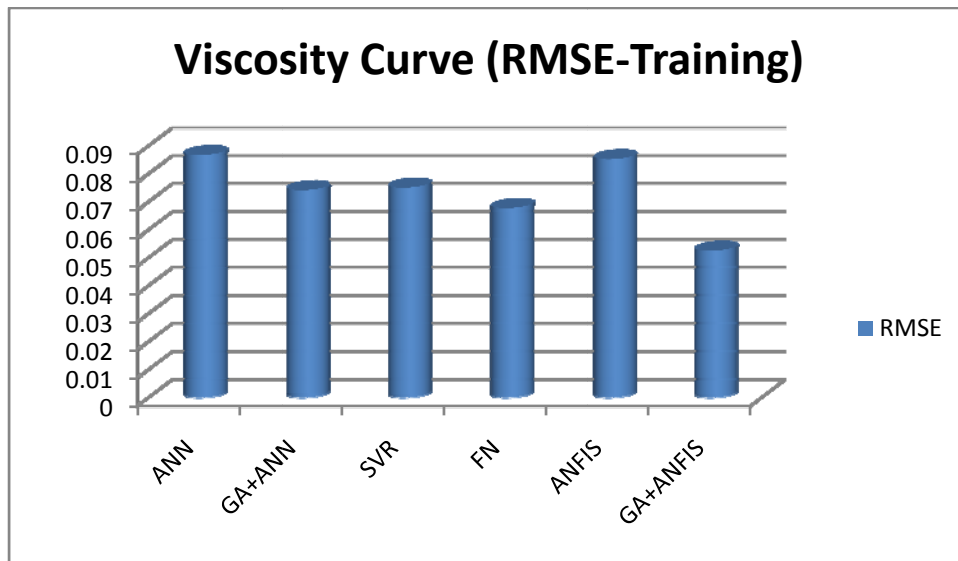


Figure 5.19: The Root Mean Square Error of all the Models for Viscosity Curve Prediction (Training)

Considering both error measures, we can conclude that GA+ANFIS has the best performance. This is followed by both FN and SVR.

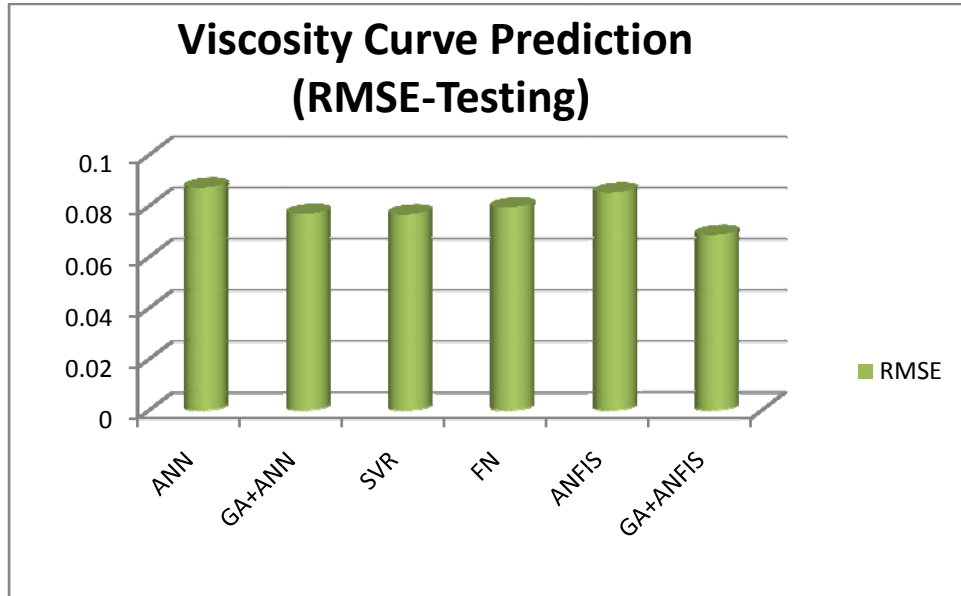


Figure 5.20: The Root Mean Square Error of all the Models for Viscosity Curve Prediction (Testing)

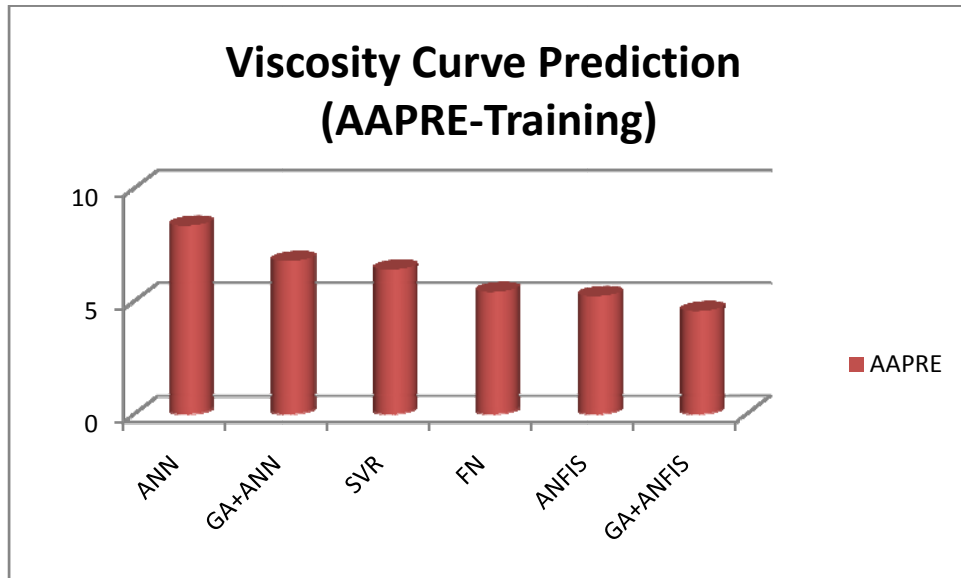


Figure 5.21: The Average Absolute Percent Relative Error of all the Models for Viscosity Curve Prediction (Training)

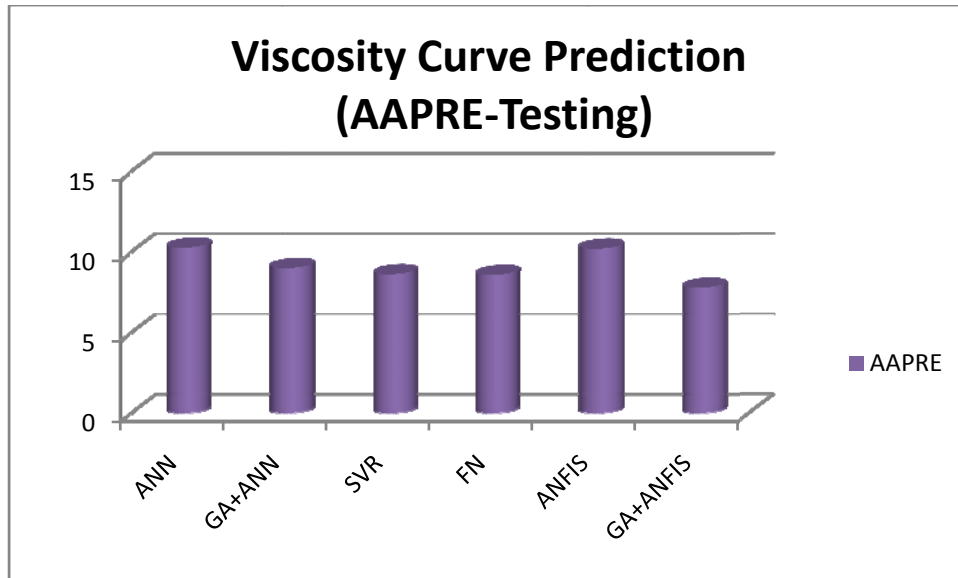


Figure 5.22: The Average Absolute Percent Relative Error of all the Models for Viscosity Curve Prediction (Testing)

Also, comparison based on time to complete development of each model is shown in Table 5.7. The training time may not be necessary or could be traded off, since after development of the model, only the testing phase will be utilised.

Table 5.7: Time Complexity of All Models for Viscosity Curve Prediction

Time Complexity for Viscosity Curve Prediction		
MODEL	CPU TIME (seconds)	
	TRAINING	TESTING
ANN	11.466	0.0468
DE+ANN	~86400	0.0512
SVR	2.6208	0.00103
FN	2.318	0.0936
ANFIS	12.3444	0.1234
GA+ANFIS	~118800	0.1256

### 5.3. Results and Discussion for Gas/Oil Ratio Curve Prediction

The same input data that were used for all the three variables under viscosity curve prediction are also used here for training our models. For gas/oil ratio curve prediction as explained in section 4.3.2 two variables,  $\tau$  and  $R_{sb}$  have to be predicted.

Similar to the case of viscosity curve prediction, 70% of the data set, equivalent to the number of oil wells, was used for training and 30% for testing.

### **5.3.1. ANN and DE+ANN for Gas/Oil Ratio Curve Prediction**

We implemented the models of ANN and DE+ANN as described in section 4.4.1. Sample plots of training and testing wells for the predicted gas/oil ratio curves are shown in Figures 5.23 through 5.28. The statistical measures for evaluating the performance of the two techniques in predicting gas/oil ratio curves are also shown in Table 5.8. In this case, ANN really displays one of its drawbacks of getting stuck at “local optimal”. In the training phase, ANN has higher performance with lower RMSE (23.5217) and AAPRE (8.804342%) than DE+ANN hybrid which has RMSE of 23.5843 and AAPRE of 9.0266%. However, for the testing wells, the predicted curves from DE+ANN hybrid framework have better matching with the experimental curves than those of ANN. Also from Table 5.8, the RMSE and AAPRE for DE+ANN predictions for the testing wells are 37.17693 and 12.1026% respectively, while the RMSE and AAPRE for ANN predictions are 39.0161 and 12.701% respectively. Since DE+ANN prediction gives lower RMSE and AAPRE than ANN for testing wells, its overall performance is better than ANN in predicting gas/oil ratio curves. Likewise, Table 5.9 shows the predicted parameters of the gas/oil ratio curves in Figures 5.23 through 5.28 based on ANN and DE+ANN models.

Table 5.8: Statistical Performance Measures of ANN and DE+ANN Models for Gas/Oil Ratio Curve Prediction

MODEL	ANN		DE+ANN	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	23.5217	8.804342	23.5843	9.0266
TESTING	39.0161	12.701	37.1769	12.1026

Table 5.9: Sample Predicted Gas/Oil Ratio Curve Parameters by ANN and DE+ANN Models

	ACTUAL	ANN	DE+ANN
TRAINING: $\tau$	0.7662	0.6558	0.6499
	0.6081	0.5788	0.5887
	0.7114	0.6346	0.6821
$R_{sb}$	548	559.3255	561.1312
	400	414.8949	385.8212
	571	567.7372	562.8003
TESTING: $\tau$	0.6200	0.6468	0.6030
	0.7014	0.6797	0.6864
	0.6502	0.6827	0.7153
$R_{sb}$	688	692.3316	690.6501
	702	695.099	685.3666
	633	585.6081	562.4248



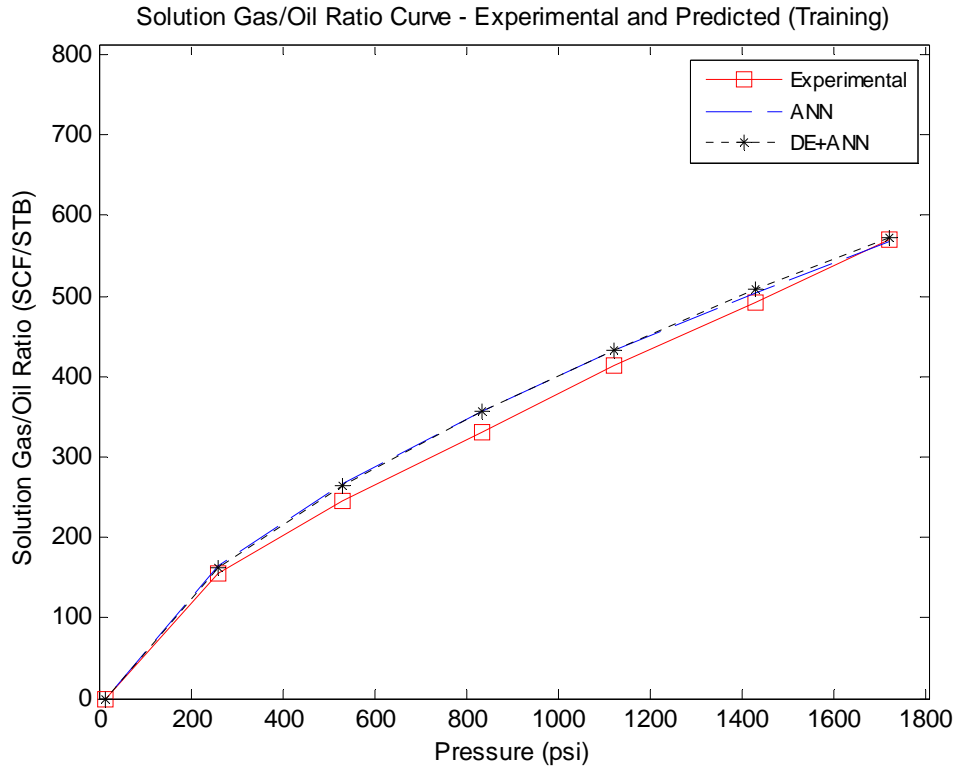


Figure 5.23: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1

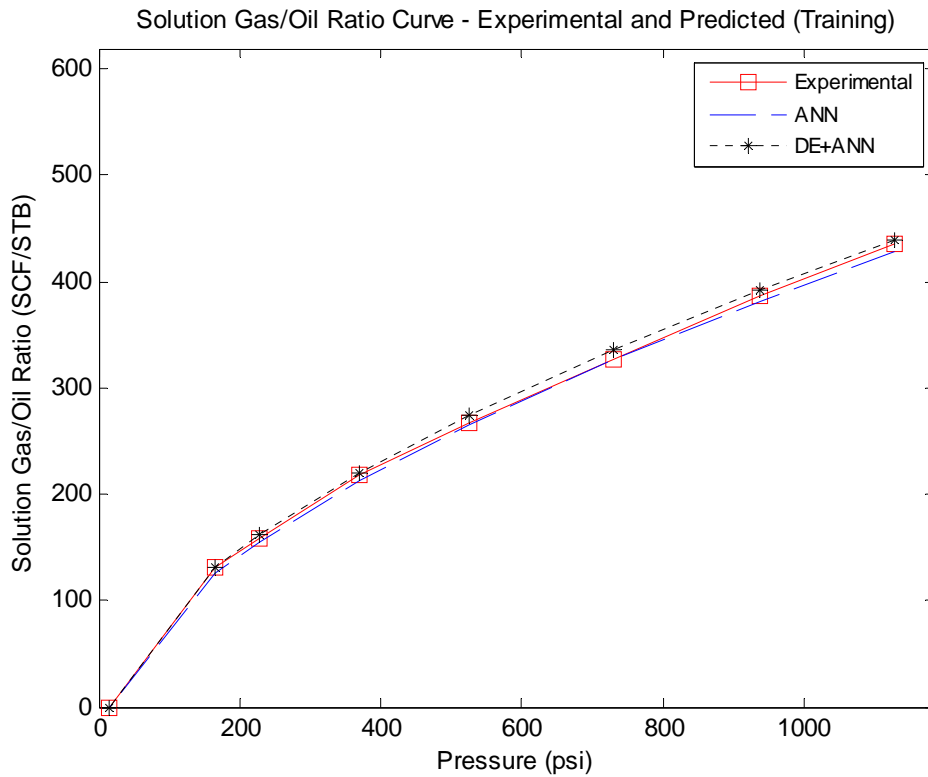


Figure 5.24: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2

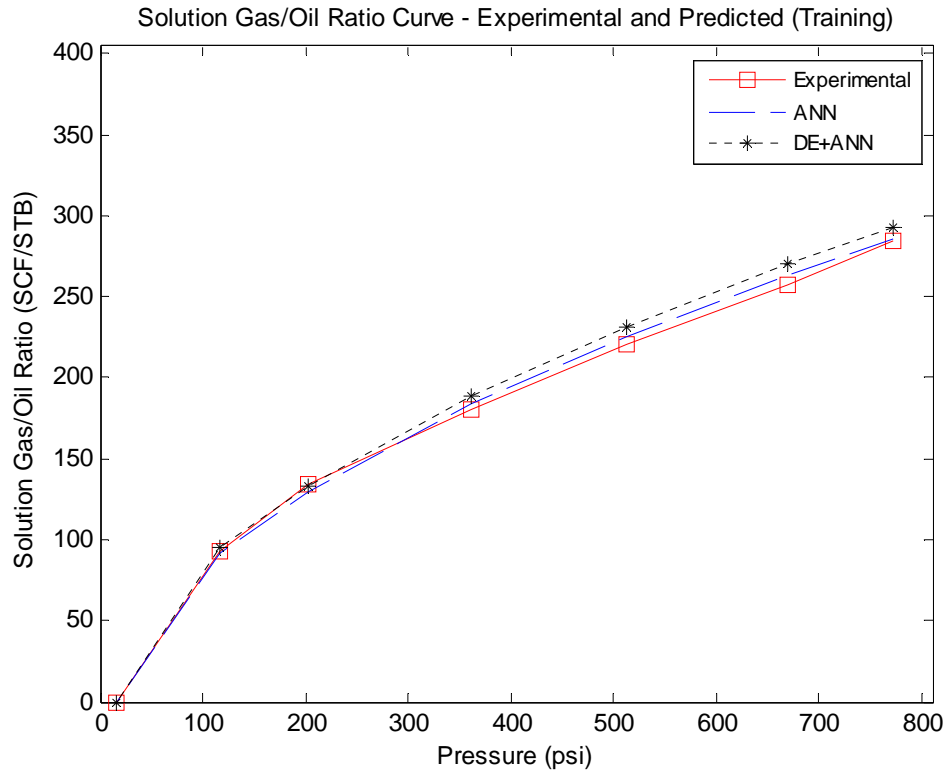


Figure 5.25: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3

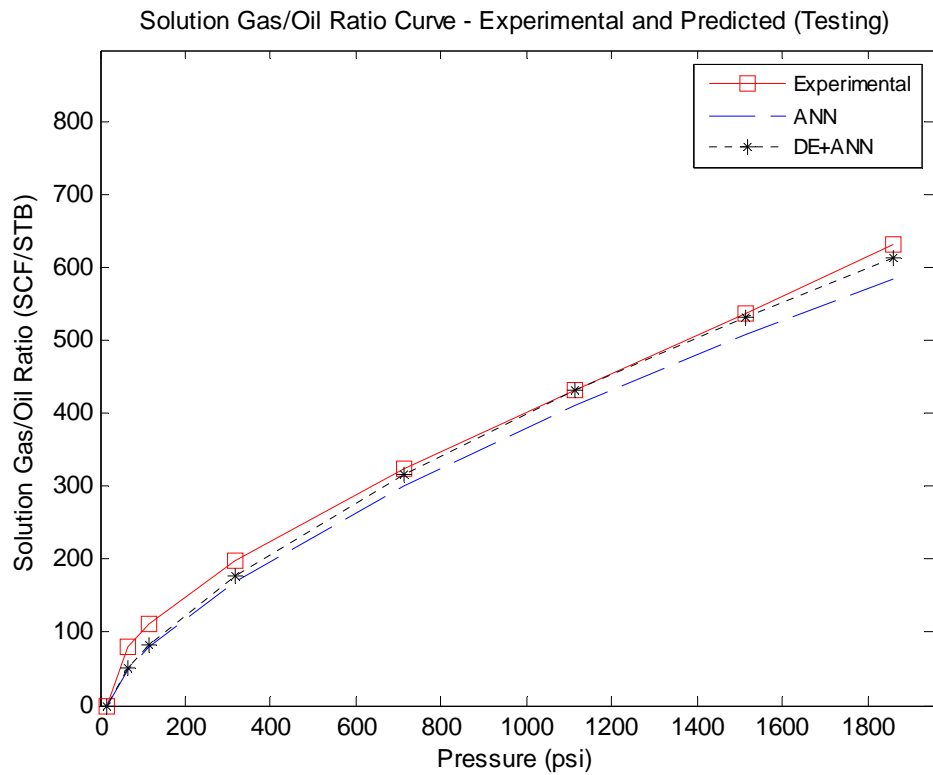


Figure 5.26: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1

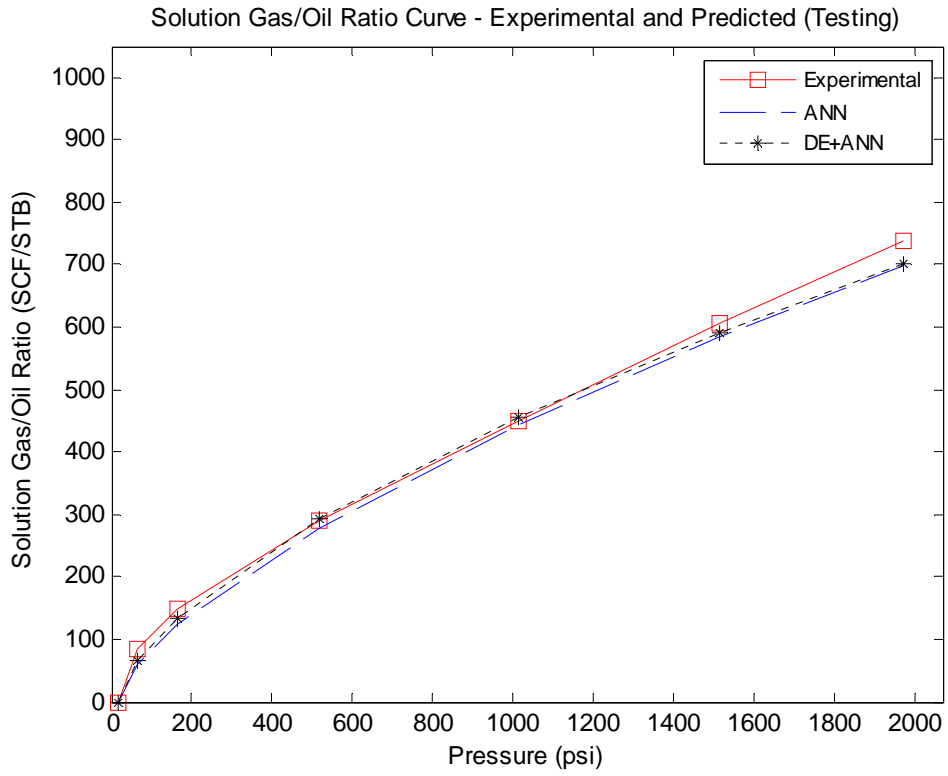


Figure 5.27: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2

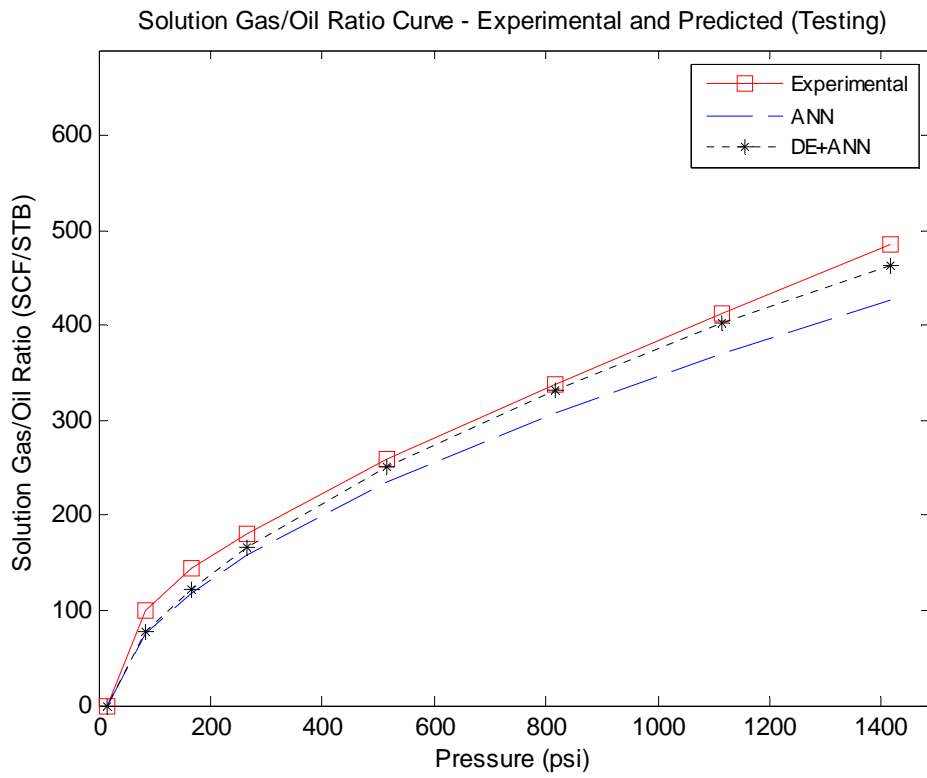


Figure 5.28: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3

### 5.3.2. SVR and FN for Gas/Oil Ratio Curve Prediction

The SVR and FN frameworks described in section 4.4.2 were implemented to predict the required variables,  $\tau$  and  $R_{sb}$ . Similar to other previous cases, only sample training and testing plots of the predicted gas/oil ratio curves are shown in Figures 5.29 through 34. Table 5.10 shows the statistical measures for evaluating the performance of SVR and FN techniques in predicting gas/oil ratio curves. The predicted curves from these two techniques show good matching with the experimental curves for training and testing wells. In this case, unlike the viscosity curve prediction where performances of both SVR and FN are very competitive, SVR has better average performance than FN in both training and testing phases, based on the statistical measures used for evaluation. For the training, SVR has RMSE 19.0043 and AAPRE of 7.5279%, while FN has RMSE of 21.6942 and AAPRE of 8.4167%. For testing, SVR has RMSE of 30.0170 and AAPRE of 9.0757%, while FN has RMSE of 32.8196 and AAPRE of 10.2012%. Based on the preceding analysis, though performance of FN is also good, SVR framework gives better performance in predicting gas/oil ratio than FN. This is also evident from the sample predicted curves. Likewise, Table 5.11 shows the predicted parameters of the gas/oil ratio curves in Figures 5.29 through 5.34 based on SVR and FN models.

Table 5.10: Statistical Performance Measures of SVR and FN Models for Gas/Oil Ratio Curve Prediction

MODEL	SVR		FN	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	19.0043	7.5279	21.6942	8.4167
TESTING	30.0170	9.0757	32.8196	10.2012

Table 5.11: Sample Predicted Gas/Oil Ratio Curve Parameters by SVR and FN Models

	ACTUAL	SVR	FN
TRAINING: $\tau$	0.7662	0.6655	0.6813
	0.6081	0.6099	0.6259
	0.7114	0.6445	0.6762
$R_{sb}$	548	584.0497	593.5556
	400	400.0016	405.2821
	571	571.0214	562.5857
TESTING: $\tau$	0.6200	0.6397	0.5983
	0.7014	0.6773	0.6497
	0.6502	0.6834	0.7090
$R_{sb}$	688	688.0279	668.7892
	702	695.2134	693.5055
	633	634.6559	602.0884

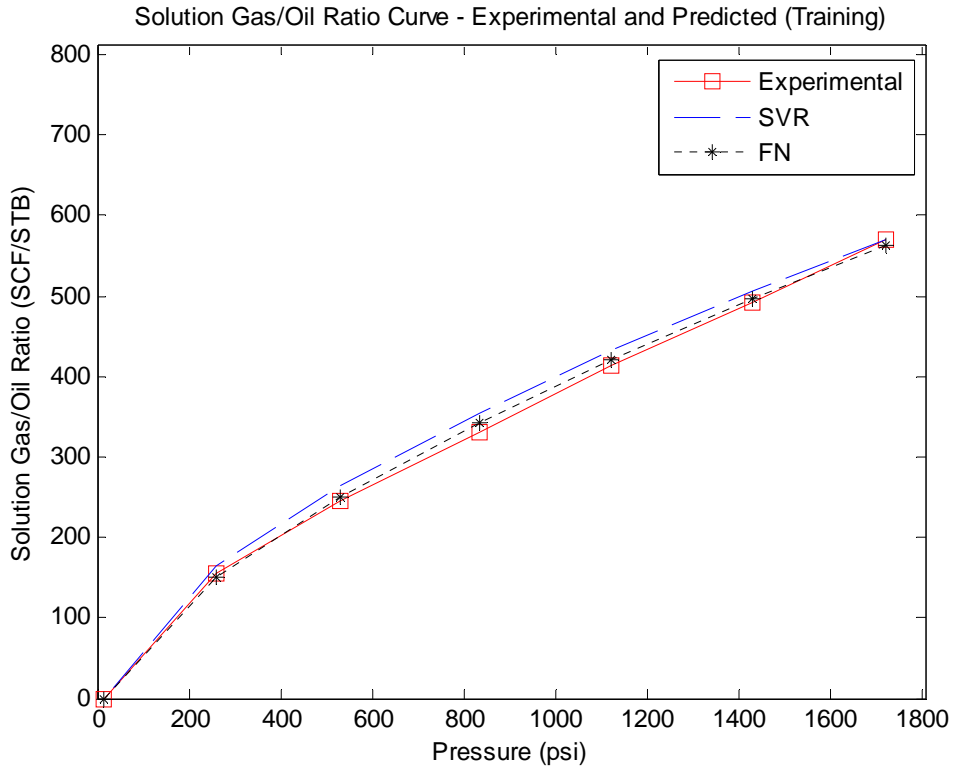


Figure 5.29: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1

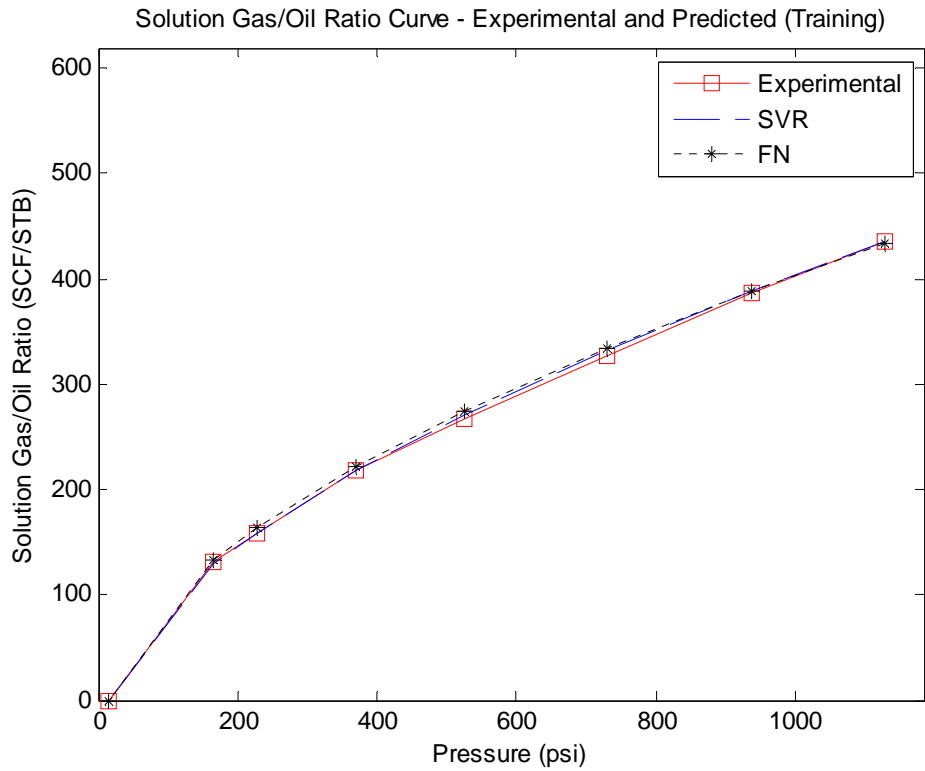


Figure 5.30: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2

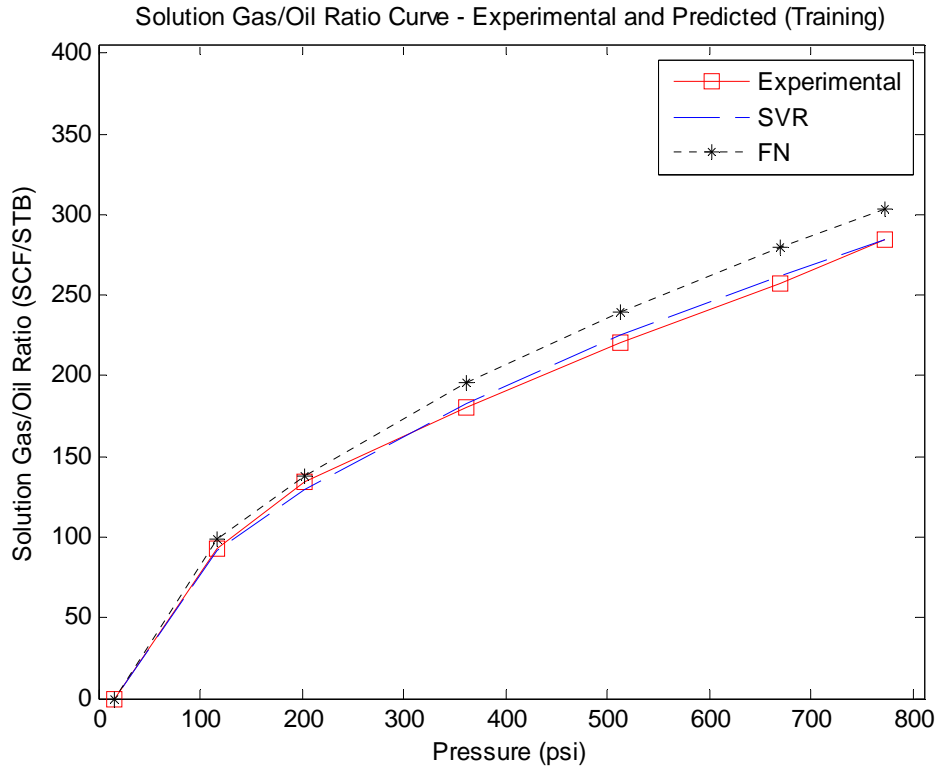


Figure 5.31: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3

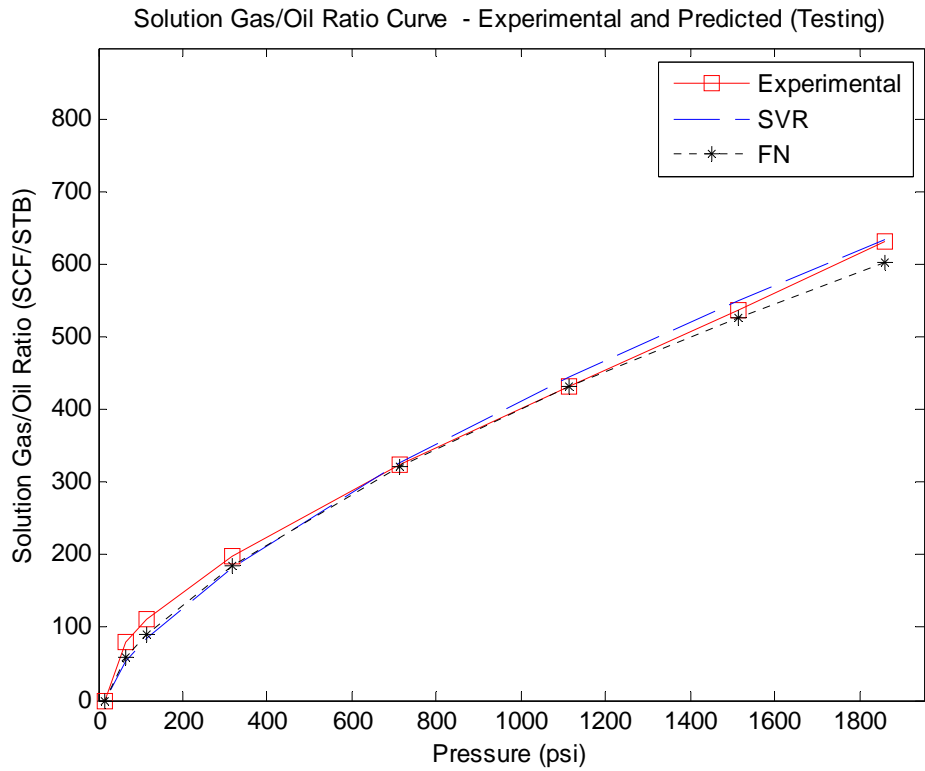


Figure 5.32: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1

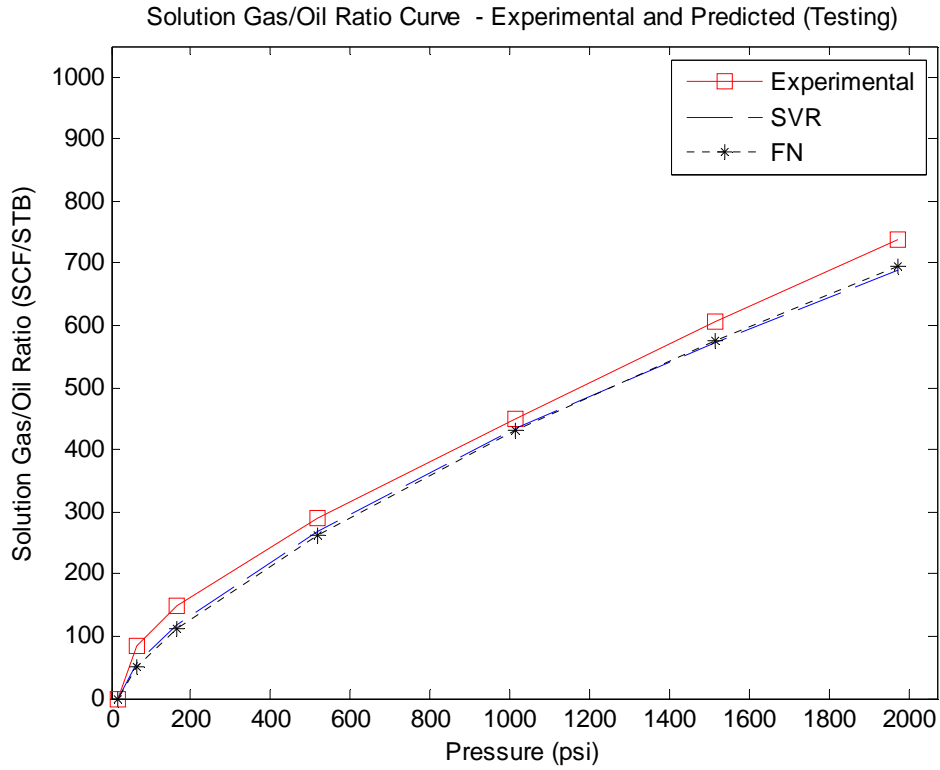


Figure 5.33: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2

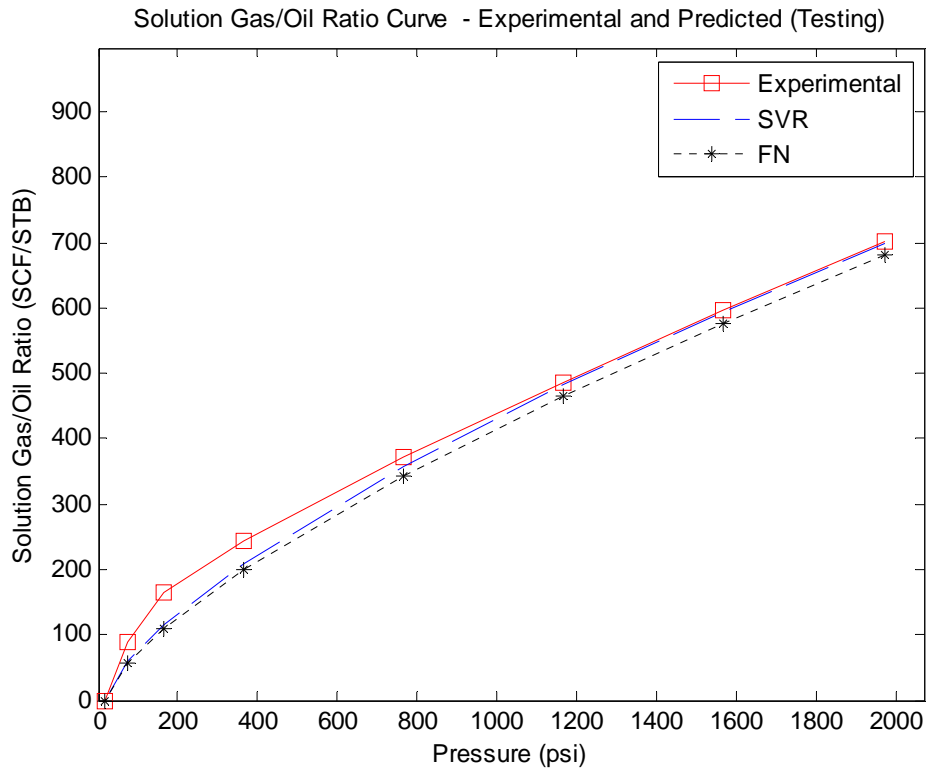


Figure 5.34: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3



### 5.3.3. ANFIS and GA+ANFIS for Gas/Oil Ratio Curve Prediction

Details of how these two frameworks were implemented for gas/oil ratio curve prediction have been highlighted in sub-section 4.4.3. As explained under section 5.2.3, selection of *radii* in ANFIS implementation with subtractive clustering is very critical for its good performance. It affects the overall performance of the adaptive fuzzy inference system. Hence, there is a need to make correct selection of the subtractive clustering *radii*. This is the role that GA plays in the GA+ANFIS hybrid framework. Without incorporating intelligent searching algorithm into the ANFIS, trial-and-error method has to be used for selecting optimal *radii*. However, this does not guarantee optimal *radii*.

Similar to other previous cases, sample plots of the predicted curves for training and testing wells from the two frameworks are shown in Figures 5.35 through 5.40. The predicted curves for training and testing show a very good matching with the experimental curves. Also, Table 5.12 shows the statistical measures used for evaluating the performance of the two models. It is obvious that GA+ANFIS has better performance than ANFIS. GA+ANFIS has lower RMSE and AAPRE for both training and testing cases. Likewise, Table 5.13 shows the predicted parameters of the gas/oil ratio curves in Figures 5.35 through 5.40 based on ANFIS and GA+ANFIS models.

Table 5.12: Statistical Performance Measures of ANFIS and GA+ANFIS Models for Gas/Oil Ratio Curve Prediction

MODEL	ANFIS		GA+ANFIS	
	RMSE	AAPRE%	RMSE	AAPRE%
TRAINING	22.8593	8.5440	19.8938	7.6141
TESTING	29.1187	8.8059	26.0679	8.3404

Table 5.13: Sample Predicted Gas/Oil Ratio Curve Parameters by ANFIS and GA+ANFIS Models

	ACTUAL	ANFIS	GA+ANFIS
TRAINING: $\tau$	0.7662	0.657453	0.678454
	0.6081	0.595331	0.609004
	0.7114	0.653415	0.650429
$R_{sb}$	548	569.4485	587.3192
	400	410.4573	403.0655
	571	564.5945	573.7231
TESTING: $\tau$	0.6200	0.617005	0.612312
	0.7014	0.658996	0.651872
	0.6502	0.658569	0.682761
$R_{sb}$	688	674.8576	687.2239
	702	701.0758	699.2593
	633	620.3292	613.7657

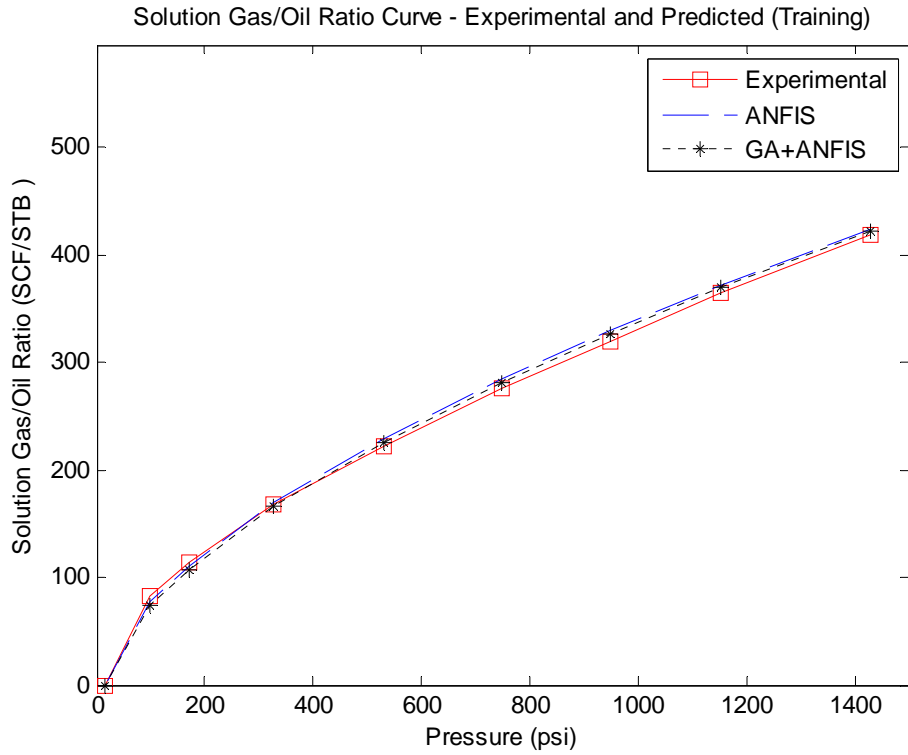


Figure 5.35: Gas/Oil Ratio vs Pressure Plot for Sample Well TR1

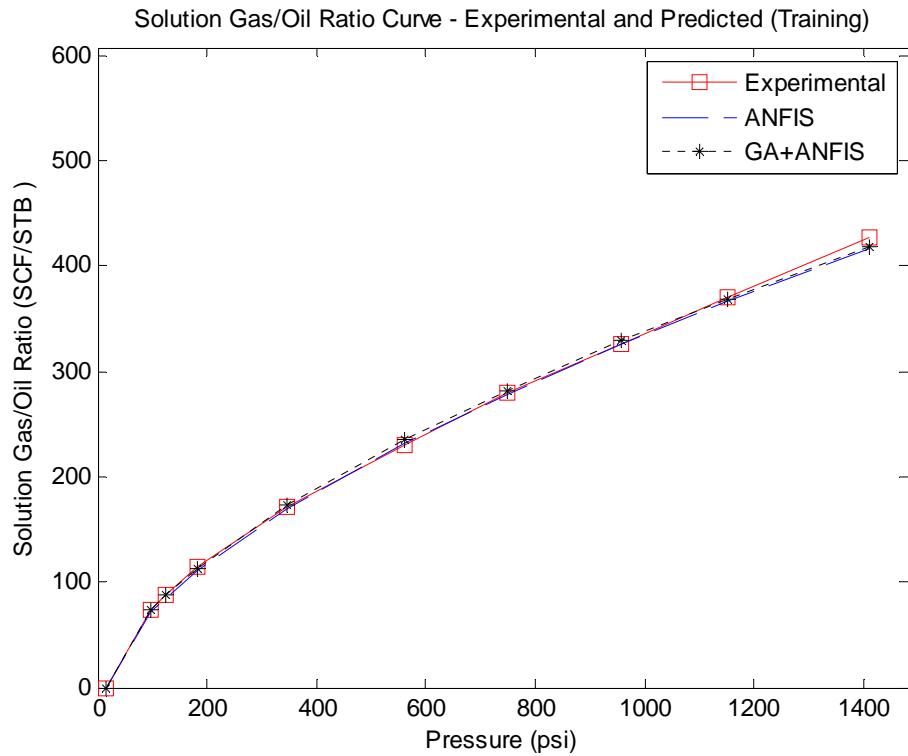


Figure 5.36: Gas/Oil Ratio vs Pressure Plot for Sample Well TR2

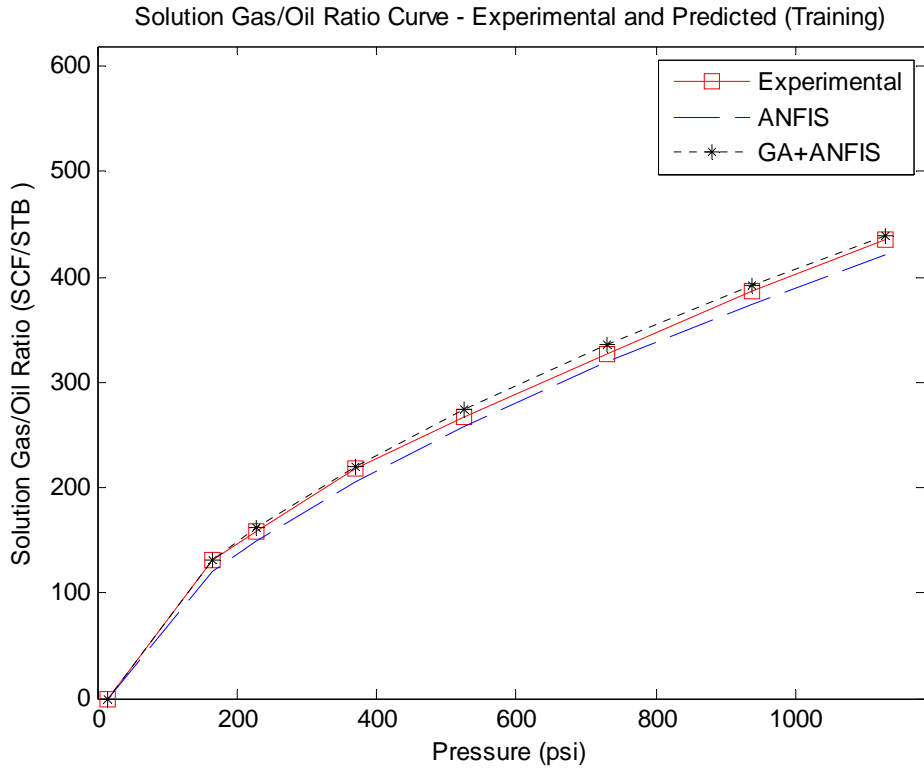


Figure 5.37: Gas/Oil Ratio vs Pressure Plot for Sample Well TR3

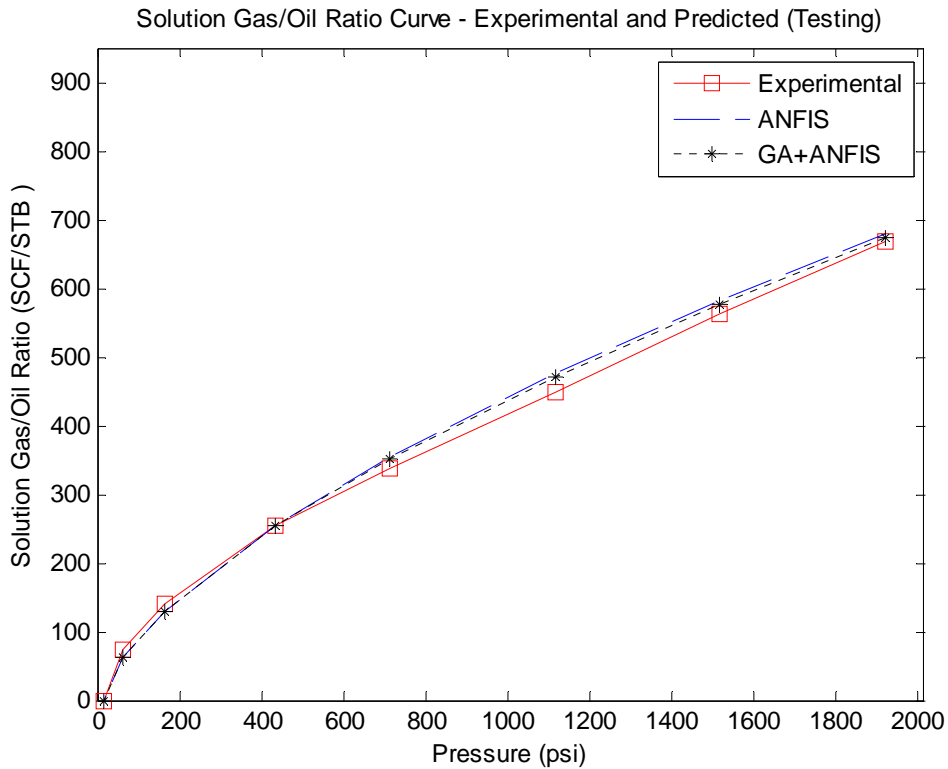


Figure 5.38: Gas/Oil Ratio vs Pressure Plot for Sample Well TS1

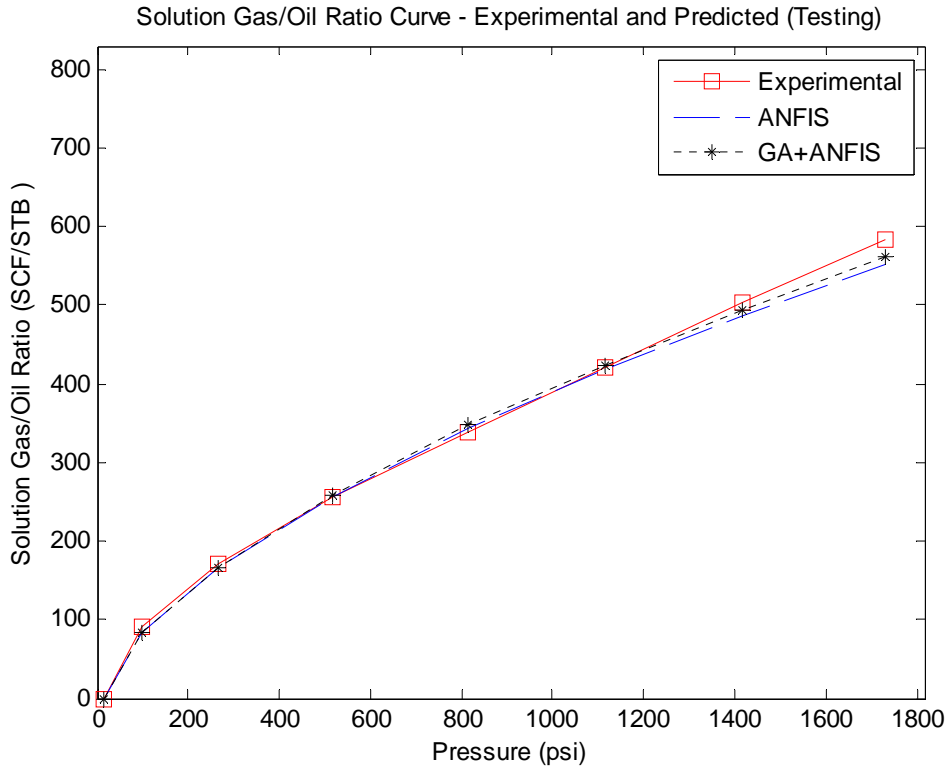


Figure 5.39: Gas/Oil Ratio vs Pressure Plot for Sample Well TS2

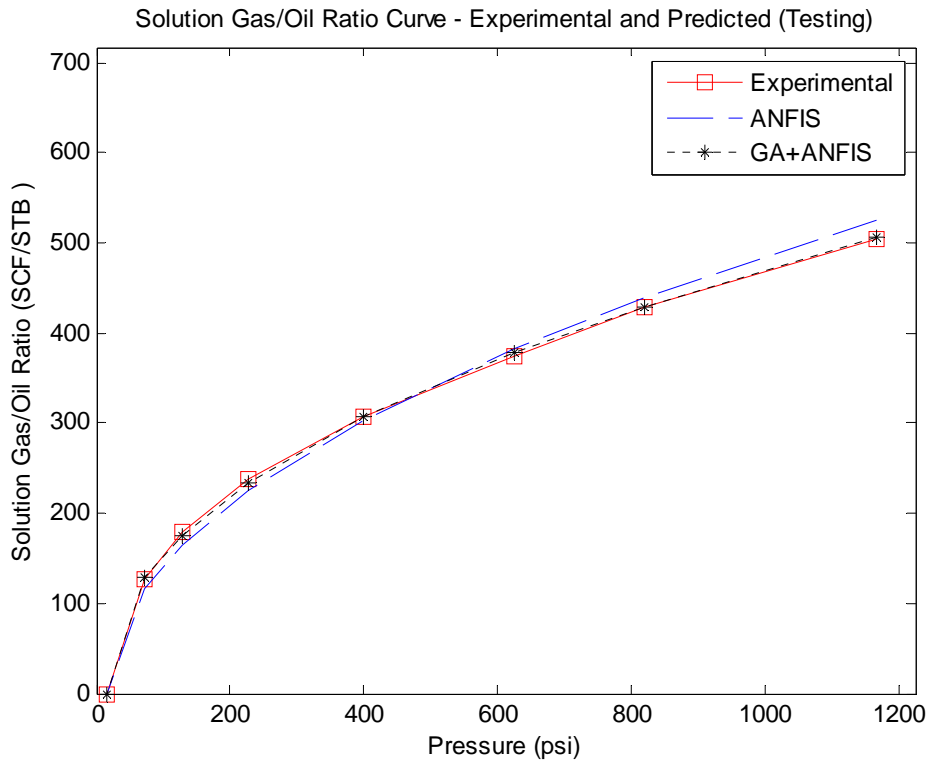


Figure 5.40: Gas/Oil Ratio vs Pressure Plot for Sample Well TS3

### 5.3.4. Performance of All the Techniques for Gas/Oil Ratio Curve Prediction

Similar to what was done for viscosity curve prediction, the results of all the soft computing techniques that have been implemented for gas/oil ratio curve prediction are also compared based on statistical measures. The overall best framework should record minimum RMSE and AAPRE for testing phase. Bar charts of RMSE and AAPRE are shown in Figures 5.41-5.44. From Figures 5.41 and 5.43, SVR has the least RMSE and APPRE for the training case. However, GA+ANFIS has the least RMSE and APPRE in Figures 5.42 and 5.44 respectively. Since the performance in the testing phase is used to rating, the models can be rated from the best as: GA+ANFIS, ANFIS, SVR, FN, DE+ANN and ANN has the least performance.

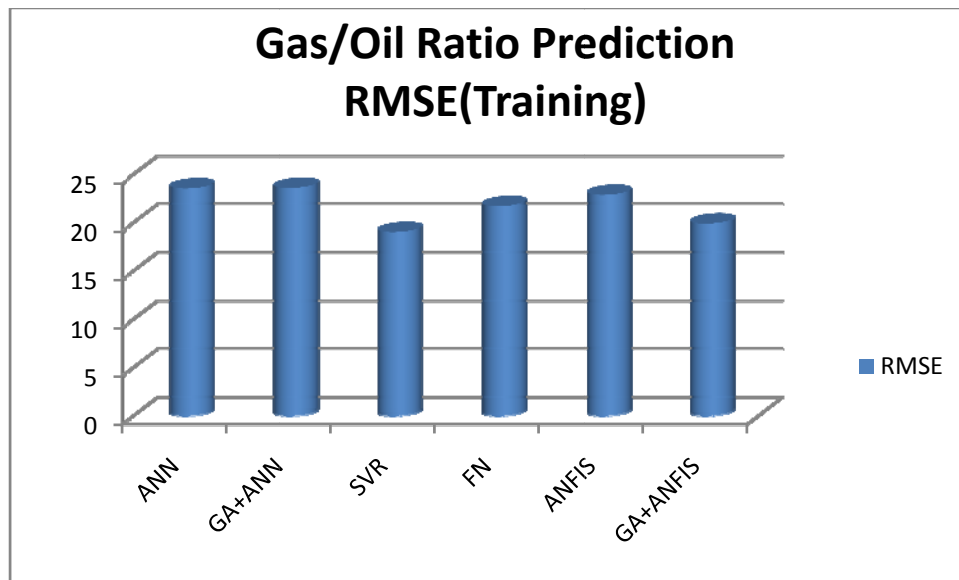


Figure 5.41: The Root Mean Square Error of all the Models for Gas/Oil ratio Curve Prediction (Training)

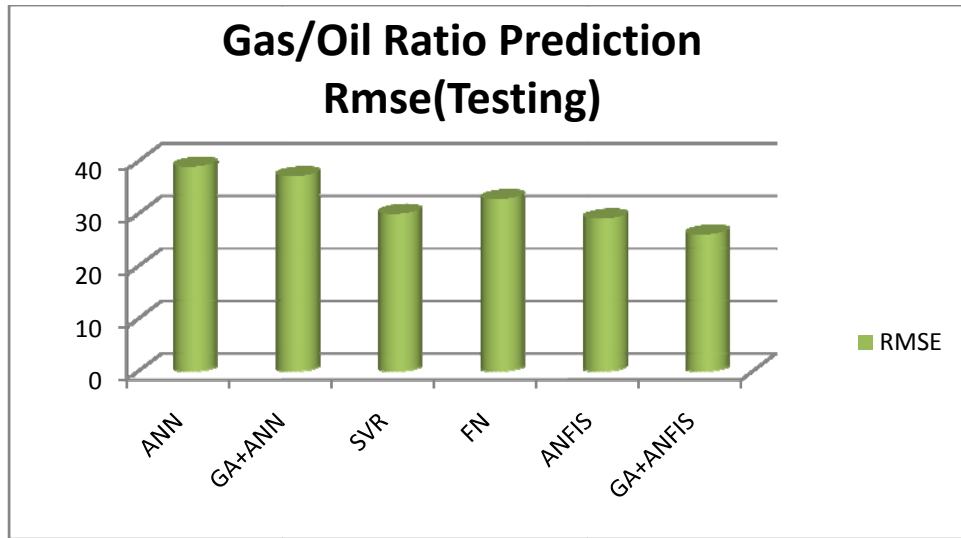


Figure 5.42: The Root Mean Square of all the Models for Gas/Oil ratio Prediction (Testing)

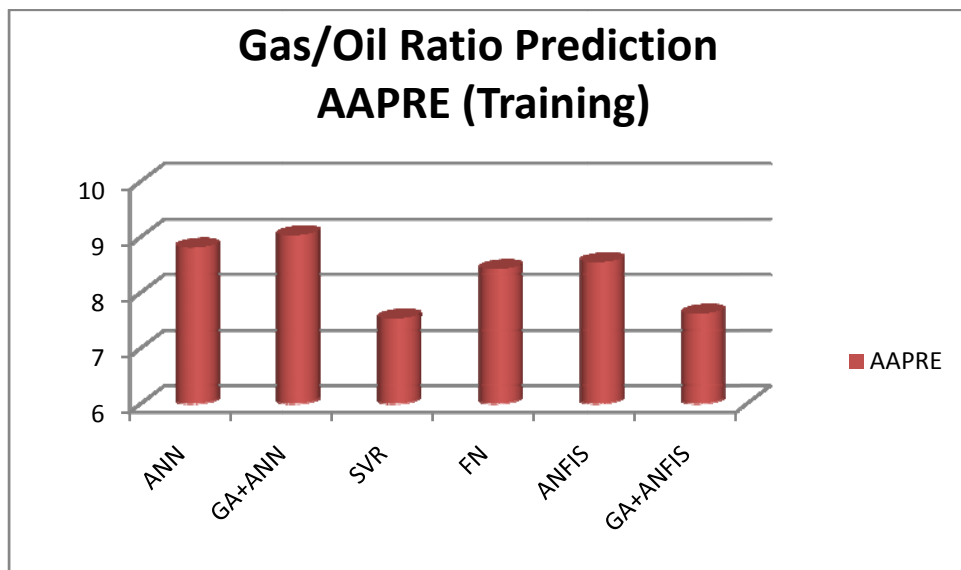


Figure 5.43: The Average Absolute Percent Relative Error of all the Models for Gas/Oil ratio Curve Prediction (Training)

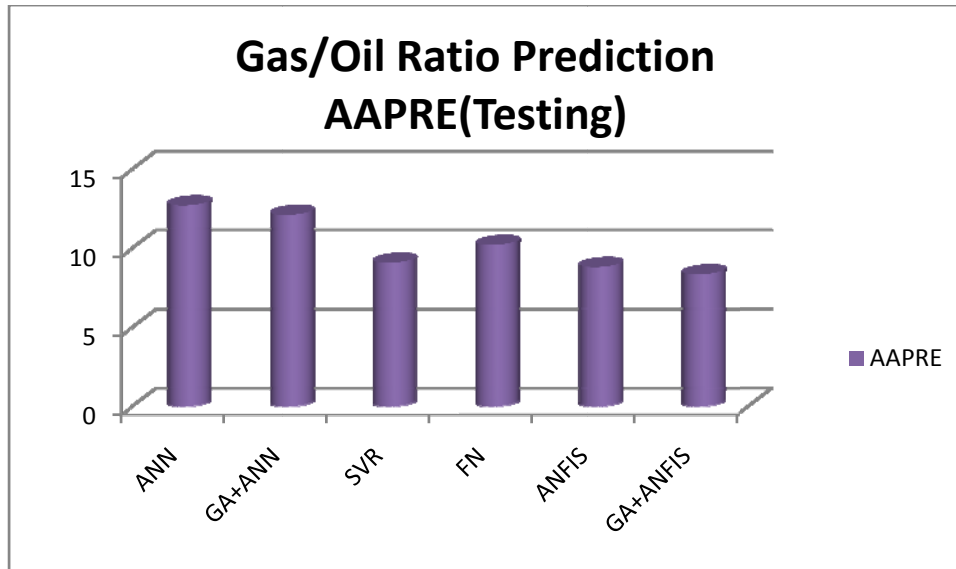


Figure 5.44: The Average Absolute Percent Relative Error of all the Models for Gas/Oil ratio Curve Prediction (Testing)

As we did for viscosity curve prediction, we also compare all the models for gas/oil ratio curve prediction based on time to complete development of each model as shown in Table 5.14. The indicated time is the total time for predicting all the two variables for complete gas/oil ratio curve prediction. The training time may not be necessary or could be traded off, since after development of the model, only the testing phase will be utilised.

Table 5.14: Time Complexity of All Models for Gas/Oil Ratio Curve Prediction

Time Complexity for Gas/Oil Ratio Curve Prediction		
MODEL	CPUTIME (seconds)	
	TRAINING	TESTING
ANN	7.566	0.078
DE+ANN	~57600	0.082
SVR	3.0264	0.0001
FN	2.4804	0.0624
ANFIS	7.024	0.064
GA+ANFIS	~79200	0.066



## CHAPTER SIX

### CONCLUSION

#### 6.1 Summary

We have presented a new approach to predict crude oil Pressure-Volume-temperature (PVT) properties that need to be represented as curves over a specified range of reservoir pressures. Instead of the usual single or multi-data points' prediction, which could distort the consistency of the curve's shape, an efficient approach for predicting such PVT properties that are represented as curves has been introduced and implemented.

In all the predictions, we have implemented four different independent soft computing techniques, viz: Feedforward Neural Network, Support Vector Regression (a variant of Support Vector Machines), Functional Networks and Adaptive Neuro-Fuzzy Inference Systems. We have also implemented two hybrid systems, DE+ANN and GA+ANFIS. We have formulated and implemented our approach for the prediction of viscosity and gas/oil ratio curves.

In the first phase of the experimentation, we examined the raw data from PVT laboratory analyses and explored two different techniques for outliers' detection. This was done only for the predictors, i.e. data set A, but the corresponding viscosity-pressure and gas/oil ratio-pressure measurements for those outlier oil wells were also removed from data sets B (viscosity-pressure measurements) and C (gas/oil ratio-pressure measurements). Only data points that were jointly detected as outliers by the two techniques were declared as such.

We then developed different models based on each of the four soft computing techniques and the hybrid frameworks for predicting both viscosity and gas/oil ratio

curves. All the techniques were categorised into three and comparisons based on graphical representation and statistical errors between the predicted curves and the actual curves have been carried out.

Considering the state of the art comparison of the implemented techniques, GA+ANFIS can be said to be the most sophisticated. Its highest performance can be ascribed to the intelligent searching role played by GA. The next to GA+ANFIS are SVR and ordinary ANFIS. Clearly, standalone ANN (feedforward neural network) is the least robust.

## **6.2. Contribution to Knowledge**

This work has achieved the following:

- A comprehensive literature survey has been carried out vis-à-vis prediction of PVT properties.
- A new approach has been introduced for predicting PVT properties. Instead of single or multi-data points prediction for PVT properties that are generated as curves, our approach predicts the entire curve. Most importantly, the shapes of the predicted curves are consistent with the physical laws and experimental/laboratory results.
- We have successfully implemented our approach through the prediction of viscosity and gas/oil ratio curves. Also, we have developed six different complete models for prediction of these two important PVT properties using the following Soft Computing techniques: Feedforward Neural Network (FFNN or ANN), Support Vector Regression (SVR), Functional Networks and Adaptive Neuro-

Fuzzy Inference Systems (ANFIS), and two hybrid frameworks: DE+ANN and GA+ANFIS.

- Though prediction results from all the models are good for viscosity and gas/oil ratio curves, GA+ANFIS hybrid framework has the best performance, viz: accuracy of results with respect to the performance criteria we have used for evaluation and comparisons for both predicted curves.
- In essence, this work will hopefully be a fast and low cost efficient simulation tool for predicting PVT properties for optimizing oil production operations.

### **6.3. Recommendations**

One of the greatest challenges that researchers of applications of Soft Computing in petroleum engineering do face is the availability of data or adequate data for experimentation. In view of this, there will be need for more collaboration between researchers from relevant fields so as to explore more opportunities in the petroleum industry. Some of the issues in petroleum engineering that still need to be addressed include but are not definitely limited to the following:

- ✚ Slug control
- ✚ Prediction of gas condensate properties and outputs of separator stages
- ✚ Experimentations with some other SC techniques (e.g. Multi-Objective genetic Algorithm) in predicting other PVT properties by utilizing and improvement on the approach used in this study.

## Nomenclature

AAPRE= average absolute percent relative error

$B_o$  = oil formation volume factor (FVF) , RB/STB( $m^3 / m^3$ )

$B_{ob}$  = oil formation volume factor (FVF) at bubble point pressure, RB/STB ( $m^3 / m^3$ )

$P$  = pressure, psi

$P_b$  = bubble point pressure, psi

RMSE= root mean square error

$R_s$  = solution gas/oil ratio, SCF/STB ( $m^3 / m^3$ )

$R_{sb}$  = bubble point solution gas/oil ratio, SCF/STB ( $m^3 / m^3$ )

$T$  = temperature, °F

$V$  = volume,  $m^3$

$\gamma_g$  = gas relative density (air=1)

$\gamma_o$  = oil relative density (water=1)

$\gamma_{API}$  = stock tank oil gravity, °API

$\mu_a$  = viscosity above bubble point,  $cP$

$\mu_b$  = viscosity below bubble point,  $cP$

$\mu_o$  = oil viscosity,  $cP$

$\mu_{ob}$  = bubble point/gas-saturated oil viscosity,  $cP$

$\mu_{od}$  = dead oil viscosity,  $cP$

## REFERENCES

- [1] Al-Marhoun M., “PVT Correlations for Middle East Crude Oils”, Journal of Petroleum Technology, May 1988, pp.650-666.
- [2] Al-Marhoun M. A., “New Correlation for Formation Volume Factor of Oil and Gas Mixtures”, Journal of Canadian Petroleum Technology, 1992, Vol. 31, N0.3, pp. 22-26.
- [3] Al-Marhoun M.A., “Evaluation of Empirically Derived PVT Properties for Middle East Crude Oils”, Journal of Petroleum Science and Engineering, Vol. 42, 2004, pp. 209–221.
- [4] Al-Marhoun M.A., and Osman E. A., “Using Artificial Neural Networks to Develop New PVT Correlations for Saudi Crude Oils”, SPE 78592, 10th Abu Dhabi International Petroleum Exhibition and Conference (ADIPEC), Abu Dhabi, UAE, 13-16 October , 2002.
- [5] Almehaideb R.A., “Improved PVT Correlations For UAE Crude Oils,” SPE 37691, SPE Middle East Oil & Gas Show and Conference, Bahrain, 15–18 March, 1997.
- [6] Ali, J. K., “Neural Networks: A New Tool for the Petroleum Industry”, SPE 27561, European Petroleum Computer Conference, Aberdeen, UK, 15-17 March, 1994.
- [7] Al-Shammasi A.A., “A Review of Bubblepoint Pressure and Oil Formation Volume Factor Correlations”, SPE Reservoir Evaluation & Engineering, 2001.
- [8] Al-Yousef H.Y and Al-Marhoun, M.A.: “Discussion of Correlation of PVT Properties for UAE Crudes”, SPEFE (March 1993) 80.

- [9] Ayoub M.A., Raja D.M. and Al-Marhoun M.A., “Evaluation of Below Bubblepoint Viscosity Correlations & Construction of a New Neural Network Model”, SPE 108439, SPE Asia Pacific Oil & Gas Conference and Exhibition, Indonesia, 30 October – 1 November, 2007.
- [10] Black M., “Reasoning with Loose Concepts”, *Dialogue* (2), pp. 1–12, 1963
- [11] Castillo E., “Functional Networks” *Neural Processing Letters* **7**: 151–159, 1998.
- [12] Castillo E., Cobo A., Guteirrez J.M., Pruneda R.E. (2000a), “Functional Networks: A New Network-Based Methodology” *Computer-Aided Civil and Infrastructure Engineering*, Volume 15 Issue 2 Page 90-106, March, 2000.
- [13] Castillo E., Cobo A., Guteirrez J.M. and Pruneda R.E., “Functional Networks with Application: A Neural-Based Paradigm”, Boston Kluwer Academic Publishers, 1999.
- [14] Cortes C. and Vapnik V., “Support-Vector Networks”, *Machine Learning*, Vol.20, pp.273-297, 1995.
- [15] Cox E., “Adaptive Fuzzy Systems”, *IEEE Spectrum*, pp. 27-31, February, 1993.
- [16] Dake L.P., “Fundamentals of Reservoir Engineering”, Elsevier Science, The Netherlands, 1978.
- [17] Darrell W. “A Genetic Algorithm Tutorial”, *Statistics and Computing* (4):65-85, 1994 .
- [18] Dokla M.E. and Osman M.E., “Correlation of PVT Properties for UAE Crudes”, *SPE Formation Evaluation*, pp. 41-46, March 1992.

- [19] Drucker H., Burges C.J.C, Kaufman L., Smola, A. and Vapnik, V., “Support Vector Regression Machines”, *Advances in Neural Information Processing Systems*, MIT Press, 9: 155-161, 1997
- [20] El-Sebakhy E., Sheltami T., Al-Bokhitan S., Shaaban Y., Raharja I. and Khaeruzzaman Y., “Support Vector Machines Framework for Predicting PVT Properties of Crude-Oil Systems”, SPE 105698, 15<sup>th</sup> SPE Middle East Oil & Gas Show and Conference, Bahrain, 11–14 March, 2007.
- [21] Elsharkawy A.M., Elgibaly A. and Alikhan A.A., “Assessment of the PVT Correlations for Predicting the Properties of the Kuwaiti Crude Oils”, 6<sup>th</sup> Abu Dhabi International Petroleum Exhibition & Conference, 16-19 October, 1994.
- [22] Elsharkawy A.M., “Modeling the Properties of Crude Oil and Gas Systems Using RBF Network”, SPE 49961, SPE Asia Pacific Oil & Gas Conference, Perth, Australia, 12-14 October, 1998.
- [23] Farshad F., LeBlanc J.L. and Garber J. D, “Empirical PVT Correlations for Colombian Crude Oils”, SPE 36105, Fourth Latin American and Caribbean Petroleum Engineering Conference, Tobago, 23-26 April, 1996.
- [24] Glaso O., “Generalized Pressure-Volume Temperature Correlations”, *Journal of Petroleum Technology*, May 1980, pp.785-795.
- [25] Gharbi R.B. and Elsharkawy A.M., “Neural-Network Model for Estimating the PVT Properties of Middle East Crude Oils”, SPE 37695, SPE Middle East Oil Show and Conference, Bahrain, 15–18 March, 1997.

- [26] Gharbi R.B., Elsharkawy A.M. and Karkoub M., “Universal Neural-Network Based Model for Estimating the PVT Properties of Crude Oil Systems”, *Energy & Fuels*, Vol.13, 1999, pp. 454-458.
- [27] Ghetto G.D., Paone, F. and Villa M., “Reliability Analysis on PVT correlation”, SPE 28904, SPE European Petroleum Conference, London, UK, 25-27 October, 1994.
- [28] Goda H.M., El-M Shokir E.M., Fattah K.A. and Sayyoub M.H., "Prediction of the PVT Data using Neural Network Computing Theory", SPE85650, 27<sup>th</sup> Annual SPE International Technical Conference and Exhibition, Nigeria, 4-6 August, 2003.
- [29] Hagan M. T. and Fun M. H., "Levenberge-Marquardt Training Algorithm for Modular Networks", IEEE International Conference on Neural Networks, vol. 1, 1996, pp. 468-473.
- [30] Hajizadeh Y., “Viscosity Prediction of Crude Oils with Genetic Algorithms”, SPE 108439, SPE Latin American and Caribbean Petroleum Engineering Conference, Argentina, 15-18 April, 2007.
- [31] Hanafy H.H., Macary S.M., El-Naldy Y.M., Bayomi A.A. and El-Batanony M.H., “Empirical PVT Correlations Applied to Egyptian Crude Oils Exemplify Significance of Using Regional Correlations”, SPE 37295, SPE International Symposium on Oilfield Chemistry, Houston, Texas, 18-21 February, 1997.
- [32] Haykin S., “Neural Networks: A Comprehensive Foundation”, Macmillan College Publishing Company, Inc., 1994.



- [33] Hemmati M.N. and Kharrat R., “A Correlation Approach for Prediction of Crude- Oil PVT Properties”, SPE 104543, 15th SPE Middle East Oil & Gas Show and Conference, Bahrain, 11–14 March, 2007.
- [34] Hemmati M.N. and Kharrat R., “Evaluation of Empirically Derived PVT Properties for Middle East Crude Oils”, Scientia Iranica, Vol.14, No4, 2007, pp. 358-368.
- [35] Holman J.P., “Experimental Methods for Engineers”, 7th Edition, McGraw-Hill 2001.
- [36] Hornik K. M., Stinchcombe M. and White H., “Multilayer Feedforward Networks are Universal Approximators”, Neural networks 2(5), 1986, pp.359-366.
- [37] Ilonen J., Kamarainen J.K., Lampinen J., “Differential Evolution Training Algorithm for Feed-Forward Neural Networks”, Neural Processing Letters 7, 1 (2003), pp.93-105.
- [38] Jang J.S.R., "ANFIS: Adaptive-Network-Based Fuzzy Inference Systems", IEEE Transactions on Systems, Man and Cybernetics, Vol. 23, No. 3, pp. 665-685, May 1993.
- [39] Jose C. P., Neil R.E. and Curt L.W., “Neural and Adaptive Systems”, John Willey and Sons, INC, 2000.
- [40] Katz D. L., “Prediction of Shrinkage of Crude Oils”, Drilling & Production Practice, API, Dallas, (1942), pp. 137-147.
- [41] Keefe R. and Smith, P., “Vagueness: A Reader”, MIT Press, 1997.

- [42] Price K. and Storn R., "Differential Evolution". Available at;  
<http://www.icsi.berkeley.edu/~storn/code.html>
- [43] Khan S.A., Al-Marhoun, M. A, Duffuaa S.O. and Abu-Khamsin S.A., "Development of Viscosity Correlations for Crude Oils", SPE 15720, Fifth SPE Middle East Oil Show, Bahrain, 7-10 March, 1987.
- [44] Kumoluyi A.O. and Daltaban T.S., "High-Order Neural Networks in Petroleum Engineering", SPE 27905, SPE Western Regional Meeting, California, USA, 23-25 March, 1994.
- [45] Labedi R., "Use of Production Data to Estimate Volume Factor Density and Compressibility of Reservoir Fluids", Journal of Petroleum Science & Engineering, Vol.4, 1990, 357-390.
- [46] Littman W., "Introduction to Support Vector Machines", Lecture notes on CS 536: Machine Learning, Department of Computer Science, Rutgers, The State University of New Jersey, USA.
- [47] Mahmood M. M. and Al-Marhoun M. A., "Evaluation of Empirically Derived PVT Properties for Pakistani crude oils", Journal of Petroleum Science and Engineering, Vol. 16, 1996, pp.275-290.
- [48] McCain W. D. Jr., "The Properties of Petroleum Fluids", PennWell Publishing Company, 1989.
- [49] McCain W. D. Jr., Soto R. B., Valko P.P. and Blasingame T. A., "Correlation of Bubble point Pressures For Reservoir Oils—A Comparative Study", SPE 51086 , SPE Eastern Regional Conference and Exhibition, Pittsburgh, PA, 9-11 November, 1998.

- [50] Mead C.A. and Conway L., "Introduction to VLSI System", Reading MA, Addison Wesley, 1980.
- [51] Mendel J.M., "Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions", Prentice-Hall, USA, 2001.
- [52] Mohaghegh S., " Virtual Intelligence Applications in Petroleum Engineering: Part 1-Artificial Neural Networks," Journal of Petroleum Technology, September, 2000.
- [53] Mohaghegh S., " Neural Network: What It Can Do for Petroleum Engineers", Journal of Petroleum Technology, January, 1995.
- [54] Nikraves M., Aminzadeh F., Zadeh L.A. (editors), "Soft Computing and Intelligent Data Analysis in Oil Exploration", Elsevier, 2003.
- [55] Omar, M.I. and Todd, A.C., "Development of New Modified Black oil Correlation for Malaysian Crudes", SPE 25338, SPE Asia Pacific Oil & Gas Conference and Exhibition, Singapore, 8-10 February, 1993.
- [56] Osman E. A., Abdel-Wahhab O. A. and Al-Marhoun M. A., "Prediction of Oil PVT Properties Using Neural Networks", SPE 68233, SPE Middle East Oil Show, Bahrain, 17-20 March, 2001.
- [57] Osman E. A. and Abdel-Aal R.E., "Abductive Networks: A New Modeling Tool for the Oil and Gas Industry", SPE 77882, SPE Asia Pacific Oil and Gas Conference and Exhibition Melbourne, Australia, 8–10 October 2002
- [58] Osman E. and Al-Marhoun M., "Artificial Neural Networks Models for Predicting PVT Properties of Oil Field Brines", SPE 93765, 14th SPE Middle East Oil & Gas Show and Conference, Bahrain, 12-15 March, 2005.

- [59] Petrosky G.E. Jr. and Farshad, F.F., “Viscosity Correlations for Gulf of Mexico Crude Oils”, SPE 29468, Production Operations Symposium, Oklahoma City, OK, U.S.A., 2-4 April, 1995.
- [60] Petrosky G.E. Jr. and Farshad F.F., “Pressure Volume Temperature Correlations for Gulf of Mexico Crude Oils”, SPE Reservoir Evaluation & Engineering, October 1998, pp. 416-420.
- [61] Russell B., “Vagueness”, Austrian Journal of Philosophy (1), pp. 84–92, 1923.
- [62] Sidqi A. and Al-Marhoun M.A., “Development of a New Correlation for Bubble point Oil Viscosity”, The Arabian Journal for Science and Engineering, Vol. 16, No. 2A, 1991.
- [63] Standing M.B., “A Pressure-Volume-Temperature Correlation for Mixtures of California Oils and Gases”, Drilling & Production Practice, API, Dallas, 1947, pp. 275-87.
- [64] Standing M.B., “Oil-System Correlation”, Petroleum Production Handbook, McGraw-Hill Book Co., New York City (1962) 2, Chap. 19.
- [65] Standing M. B., “Volumetric and Phase Behavior of Oil Field Hydrocarbon Systems”, Millet Print Inc., Dallas, TX (1977) 124.
- [66] Sunday S.I., John M. and Lanre D., “Black Oil Empirical PVT Correlations Screening for the Niger Delta Crude”, SPE 105984, 30<sup>th</sup> Annual SPE International Technical Conference and Exhibition, Nigeria, July 31-August 2, 2006.
- [67] Takagi T. and Sugeno M., “Derivation of Fuzzy Control Rules from Human Operator’s Control Actions”, Proceeding of the IFAC Symposium on Fuzzy

- Information, Knowledge Representation and Decision Analysis, pages 55- 60,  
July 1983.
- [68] Turksen I. B., “Measurement of Membership Functions and their Acquisition”,  
Fuzzy Sets and Systems, pp.405-438, 1991.
- [69] Varotsis N., Gaganis V., Nighswander J., and Guieze P., “A Novel Non-  
Iterative Method for the Prediction of the PVT Behavior of Reservoir  
Fluids”, SPE 56745 SPE Annual Technical Conference and Exhibition,  
Houston, Texas, 3-6 October, 1999.
- [70] Vazquez M. and Beggs H.D., “Correlation for Fluid Physical Property  
Prediction”, Journal of Petroleum Technology, 1980, pp. 968-970.
- [71] Velarde J., Blasingame T.A. and McCain W.D., “Correlation of Black Oil  
Properties at Pressures below Bubble point Pressure—A New Approach”, Journal  
of Canadian Petroleum Technology, Special Edition, Vol. 38, No. 13, 1999.
- [72] Wang S., “Generating Fuzzy Membership Functions: A Monotonic Neural  
Network Model”, Fuzzy Sets and Systems, vol. 61, pp.71-81, 1994.
- [73] Watanabe N., “Statistical Methods for Estimating Membership Functions”,  
Japanese Journal of Fuzzy Theory and Systems, 5(4), 1979.
- [74] Zadeh L., “Fuzzy Logic and Its Application to Approximate Reasoning”,  
Information Processing, Vol. 74, 1994, pp. 591- 594.
- [75] Zbigniew M., “Genetic Algorithm + Data Structures = Evolution Program”,  
Springer, New York, 1992.

## Vita

### **Personal Information**

Born: 1979, Ibadan, Nigeria

Marital Status: Married

### **Education**

- 2009 - MSc in Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia
- 2006- BSc in Electrical & Electronic Engineering, University of Ibadan, Nigeria
- 2000- Diploma in Computer Science, The Polytechnic Ibadan, Nigeria

### **Research Interests**

Optimization, Instrumentation and Control ,Condition Monitoring, SCADA & DCS, General Applications of Artificial Intelligence Techniques in Control and Power electronics

### **Professional Membership**

Student member of The Institute of Engineering and Technology