

STATUS OF THESIS

Title of thesis

DEVELOPMENT AND TESTING OF UNIVERSAL PRESSURE
DROP MODELS IN PIPELINES USING ABDUCTIVE AND
ARTIFICIAL NEURAL NETWORKS

I MOHAMMED ABDALLA AYOUB MOHAMMED

hereby allow my thesis to be placed at the information Resource Center (IRC) of
Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

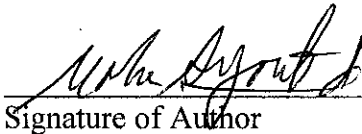
Confidential

Non-confidential

If the thesis is confidential, please state the reason:

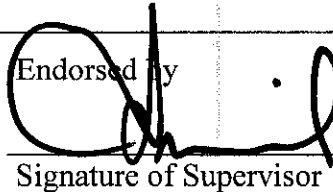
The contents of the thesis will remain confidential for _____ years.

Remarks on disclosure:


Signature of Author

Permanent address:

Dept. of Petroleum Engineering
Faculty of Engineering and Arch.
University of Khartoum
Postal Code: 11115 - P.O. Box 321
Khartoum - Sudan.
Date: 5/9/2011

Endorsed by

Signature of Supervisor

Prof. Dr. Birol M.R. Demiral

Date: 5/9/2011

UNIVERSITI TEKNOLOGI PETRONAS

DEVELOPMENT AND TESTING OF UNIVERSAL PRESSURE
DROP MODELS IN PIPELINES USING ABDUCTIVE AND
ARTIFICIAL NEURAL NETWORKS

MOHAMMED ABDALLA AYOUB MOHAMMED

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

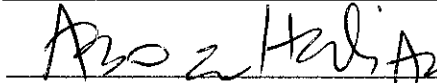
Signature:



Main Supervisor:

Prof. Dr. Birol M.R. Demiral

Signature:



Head of Department:

Assoc. Prof. Dr. Abdul Hadi Abd Rahman

Date:

5/9/11

Assoc. Prof. Dr Abdul Hadi B. Abd Rahman
Head
Faculty of Geosciences &
Petroleum Engineering
Universiti Teknologi PETRONAS

DEVELOPMENT AND TESTING OF UNIVERSAL PRESSURE
DROP MODELS IN PIPELINES USING ABDUCTIVE AND
ARTIFICIAL NEURAL NETWORKS

MOHAMMED ABDALLA AYOUB MOHAMMED

A Thesis

Submitted to the Postgraduate Studies Programme
as a Requirement for the Degree of

DOCTOR OF PHILOSOPHY

GEOSCIENCE AND PETROLEUM ENGINEERING DEPARTMENT

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SRI ISKANDAR

PERAK

SEPTEMBER 2011

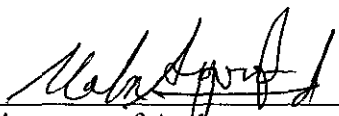
DECLARATION OF THESIS

Title of thesis

DEVELOPMENT AND TESTING OF UNIVERSAL PRESSURE
DROP MODELS IN PIPELINES USING ABDUCTIVE AND
ARTIFICIAL NEURAL NETWORKS

I MOHAMMED ABDALLA AYOUB MOHAMMED

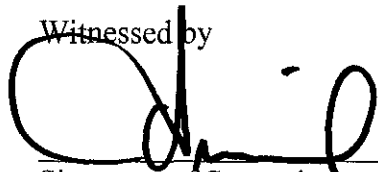
hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.



Signature of Author

Permanent address:
Dept. of Petroleum Engineering
Faculty of Engineering and Arch.
University of Khartoum
Postal Code: 11115 - P.O. Box 321
Khartoum - Sudan.

Date: 5/9/2011

Witnessed by


Signature of Supervisor

Name of Supervisor
Prof. Dr. Birol M.R. Demiral

Date: 5/9/2011

ACKNOWLEDGEMENTS

All praise is due to Allah the most beneficent, the most compassionate, and peace is upon his prophet Mohammed. I wish firstly to express my sincere gratitude and thanks to PETRONAS-Sudan for providing me full financial support during the entire research period. I owe my deepest gratitude to my supervisor Professor Birol Mehmet Raef Demiral, for his continuous guidance, steadfast encouragement and advice that made this work possible. He has provided me with full support throughout my thesis with his patience and knowledge whilst allowing me the room to work in my own way. I attribute the level of my PhD degree to his continuous support and endeavor and without him this thesis, too, would not have been completed or finalized. One simply could not wish for a better or friendlier supervisor ever.

I would like also to thank my wife for offering me excellent living environment throughout the course of this study without expecting anything in return. My special appreciation goes to my deepest friends Dr. Abdelbagi and Dr. Maysara for their invaluable recommendations. They have shared precious insights in the relevance of the study and thesis formatting. A word of thank is also due Mr. Gint Jekabsons who helped me a lot in developing the GMDH model through those invaluable suggestions delivered by e-mail. I am indebted to many of my colleagues for supporting me. A word of thank is due to Dr. Osman Ahmed for his considerable effort in generating the GUI of the ANN model.

DEDICATION

THIS IS WORK IS COMPLETELY DEDICATED TO
MY BELOVED WIFE AMJAAD AND LOVELY KIDS
ABDALLA & ADNAN. WITHOUT THEM, I COULD
NOT ENJOY STAYING HERE.

ABSTRACT

Determination of pressure drop in pipeline system is difficult. Conventional methods (empirical correlations and mechanistic methods) were not successful in providing accurate estimate. Artificial Neural Networks and polynomial Group Method of Data Handling techniques had received wide recognition in terms of discovering hidden and highly nonlinear relationships between input and output patterns. The potential of both Artificial Neural Networks (ANN) and Abductory Induction Mechanism (AIM) techniques has been revealed in this study by generating generic models for pressure drop estimation in pipeline systems that carry multiphase fluids (oil, gas, and water) and with wide range of angles of inclination. No past study was found that utilizes both techniques in an attempt to solve this problem. A total number of 335 data sets collected from different Middle Eastern fields have been used in developing the models. The data covered a wide range of variables at different values such as oil rate (2200 to 25000 bbl/d), water rate (up to 8424 bbl/d), angles of inclination (-52 to 208 degrees), length of the pipe (500 to 26700 ft) and gas rate (1078 to 19658 MSCFD). For the ANN model, a ratio of 2: 1: 1 between training, validation, and testing sets yielded the best training/testing performance. The ANN model has been developed using resilient back-propagation learning algorithm. The purpose for generating another model using the polynomial Group Method of Data Handling technique was to reduce the problem of dimensionality that affects the accuracy of ANN modeling. It was found that (by the Group Method of Data Handling algorithm), length of the pipe, wellhead pressure, and angle of inclination have a pronounced effect on the pressure drop estimation under these conditions. The best available empirical correlations and mechanistic models adopted by the industry had been tested against the data and the developed models.

Graphical and statistical tools had been utilized for comparing the performance of the new models and other empirical correlations and mechanistic models.

Thorough verifications have indicated that the developed Artificial Neural Networks model outperforms all tested empirical correlations and mechanistic models as well as the polynomial Group Method of Data Handling model in terms of highest correlation coefficient, lowest average absolute percent error, lowest standard deviation, lowest maximum error, and lowest root mean square error.

The study offers reliable and quick means for pressure drop estimation in pipelines carrying multiphase fluids with wide range of angles of inclination using Artificial Neural Networks and Group Method of Data Handling techniques. Graphical User Interface (GUI) has been generated to help apply the ANN model results while an applicable equation can be used for Group Method of Data Handling model. While the conventional methods were not successful in providing accurate estimate of this property, the second approach (Group Method of Data Handling technique) was able to provide a reliable estimate with only three-input parameters involved. The modeling accuracy was not greatly harmed using this technique.

ABSTRAK

Penentuan kejatuhan tekanan dalam sistem talian paip adalah sukar. Kaedah konvensional (korelasi empirik dan kaedah mekanistik) gagal untuk memberi anggaran yang tepat. Rangkaian Neural Buatan dan Kaedah Kumpulan polinomial-teknik Pengendalian Data telah mendapat pengiktirafan yang meluas dari segi menemui hubungan antara input dan pola output yang tersembunyi dan sangat tak linear. Potensi kedua-dua Rangkaian Neural Buatan (ANN) dan teknik Mekanisme Induksi Abductory (AIM) telah didedahkan dalam kajian ini dengan menjana model generik untuk anggaran kejatuhan tekanan dalam sistem saluran paip yang membawa berbilang fasa cecair (minyak, gas dan air) dan dengan luas pelbagai sudut kecondongan. Tiada kajian yang lepas telah ditemui yang menggunakan kedua-dua teknik dalam usaha untuk menyelesaikan masalah ini. Menetapkan jumlah 335 data yang dikutip dari bidang Timur Tengah yang berbeza telah digunakan dalam membangunkan model. Data meliputi pelbagai pemboleh ubah pada nilai yang berbeza seperti kadar minyak (2200-25000 bbl/d), kadar air (sehingga 8424 bbl/d), sudut kecondongan (-52 hingga 208 darjah), panjang paip (500-26700 kaki) dan kadar gas (1078-19658 MSCFD). Bagi model ANN, nisbah 2: 1: 1 antara latihan, pengesahan, dan ujian set menghasilkan latihan / ujian prestasi yang terbaik. Model ANN telah dibangunkan dengan menggunakan pembelajaran algoritma perambatan balik berdaya tahan. Tujuan untuk menghasilkan satu lagi model yang menggunakan Kaedah Kumpulan polinomial teknik Pengendalian Data adalah untuk mengurangkan masalah kematraan yang menjejaskan ketepatan permodelan ANN. Ia didapati bahawa (Kaedah Kumpulan algoritma Pengendalian Data), panjang paip, tekanan kepala telaga, dan sudut kecenderungan mempunyai kesan ketara ke atas anggaran kejatuhan tekanan di bawah syarat-syarat ini. Korelasi terbaik tersedia empirik dan model mekanistik yang diguna pakai oleh industri telah diuji terhadap data dan model yang dibangunkan.

Alat grafik dan statistik telah digunakan untuk membandingkan prestasi model baru dan lain-lain korelasi empirik dan model mekanistik.

Pengesahan yang teliti telah menyatakan bahawa maju model Rangkaian Neural Buatan melebihi performa semua korelasi empirik diuji dan model mekanistik serta. Kaedah Kumpulan polinomial Data Mengendalikan model dari segi pekali korelasi tertinggi, terendah purata peratus ralat mutlak, paling rendah sisihan piawai, ralat maksimum terendah, dan akar paling rendah bermakna kesilapan persegi. Kajian ini menawarkan meansfor anggaran kejatuhan tekanan yang boleh dipercayai dan cepat dalam saluran paip yang membawa cecair berbilang dengan pelbagai sudut kecenderungan menggunakan Rangkaian Neural Buatan dan Kaedah Kumpulan teknik Data Pengendalian. Antara Muka Pengguna grafik (GUI) telah dijana untuk membantu memohon keputusan model ANN manakala satu persamaan yang berkenaan boleh digunakan bagi Kaedah Kumpulan model Pengendalian Data. Walaupun kaedah konvensional gagal untuk menyediakan anggaran tepat harta ini, pendekatan kedua (Kumpulan Kaedah teknik Pengendalian Data) dapat menyediakan suatu anggaran yang boleh dipercayai dengan hanya tiga input parameter yang terlibat. Ketepatan peragaan tidak menganiaya banyak menggunakan teknik ini.

In compliance with the terms of the Copyright Act 1987 and the IP Policy of the university, the copyright of this thesis has been reassigned by the author to the legal entity of the university,

Institute of Technology PETRONAS Sdn Bhd.

Due acknowledgement shall always be made of the use of any material contained in, or derived from, this thesis.

© MOHAMMED ABDALLA AYOUB MOHAMMED, 2011
Institute of Technology PETRONAS Sdn Bhd
All rights reserved.

TABLE OF CONTENTS

Acknowledgements.....	v
Dedication.....	vi
Abstract.....	vii
Abstrak.....	ix
Table of Contents.....	xii
list of figures.....	xvii
List of tables.....	xxi
List of Abbreviations.....	xxiii
Nomenclature.....	xxvi
CHAPTER 1.....	1
INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Motivation of Study.....	2
1.3 Approach.....	4
1.4 Objectives of the Research.....	5
1.5 Benefits of the Research.....	5
1.6 Summary.....	6
CHAPTER 2.....	7
LITERATURE REVIEW.....	7
2.1 Overview.....	7
2.2 Introduction.....	8
2.3 Empirical Correlations.....	9
2.3.1 Background.....	9
2.4 Mechanistic Models.....	16
2.4.1 Pipeline Mechanistic Models.....	17
2.4.1.1 Pipeline Mechanistic Models of Single Flow Regime.....	17
2.4.2 Unified Mechanistic Models.....	23
2.5 State of the Art.....	25
2.6 Artificial Intelligence.....	27

2.6.1 Artificial Neural Network	28
2.6.1.1 Historical Background	28
2.6.1.2 Definition	29
2.6.1.3 Brain system.....	30
2.6.2 The Use of Artificial Neural Networks in Petroleum Industry	32
2.6.3 Artificial Neural Networks in Multiphase Flow	32
2.7 Abductive Induction Mechanism (AIM)	34
2.7.1 Short History	34
2.7.2 Advantages and Disadvantages of AIM	36
2.8 The Use of Abductive Networks in Geosciences and Petroleum Industry	36
2.9 Summary	40
CHAPTER 3	41
RESEARCH METHODOLOGY	41
3.1 Overview	41
3.1.1 SCADA System	43
3.2 Data Selection	45
3.2.1 Database Queries	45
3.2.2 Data Filtration	45
3.3 Data Preprocessing.....	47
3.3.1 Data Clearing	47
3.3.2 Data Reduction.....	47
3.3.3 Data Transformation	47
3.4 Data Handling for ANN Model	48
3.4.1 Data Collection and Partitioning for ANN Model.....	48
3.4.2 Partitioning.....	49
3.5 ANN Model Development.....	50
3.5.1 Introduction.....	50
3.5.2 ANN Model's Features	51
3.5.3 ANN Model Architecture	51
3.6 Network Selection.....	52
3.6.1 Network Training.....	53
3.6.1.1 Application of Validation Set	54
3.7 Output Post-Processing (Denormalization)	54

3.8 Software Used.....	54
3.9 Network Performance Comparison.....	55
3.9.1 Trend Analysis.....	56
3.9.2 Group Error Analysis.....	56
3.9.3 Statistical Error Analysis.....	56
3.9.3.1 Average Percent Relative Error (APE).....	57
3.9.3.2 Average Absolute Percent Relative Error (AAPE).....	57
3.9.3.3 Minimum Absolute Percent Relative Error.....	58
3.9.3.4 Maximum Absolute Percent Relative Error.....	58
3.9.3.5 Root Mean Square Error (RMSE).....	58
3.9.3.6 Standard Deviation (SD).....	59
3.9.3.7 The Correlation Coefficient (R).....	59
3.9.4 Graphical Error Analysis.....	60
3.9.4.1 Cross-plots.....	60
3.9.4.2 Error Distributions.....	60
3.9.4.3 Residual Analysis.....	61
3.10 Building AIM Model.....	61
3.11 Uncertainty Study.....	61
3.11.1 Definition.....	62
3.11.2 Uncertainty in ANN Modeling.....	62
3.11.2.1 Measurement of Uncertainty in ANN Model.....	63
3.11.2.2 Propagation of Uncertainty.....	63
3.12 Sensitivity Analysis.....	65
3.13 Limitations.....	66
3.14 Summary.....	66
CHAPTER 4.....	69
RESULTS AND DISCUSSION.....	69
4.1 Introduction.....	69
4.1.1 ANN Model Optimization.....	70
4.1.2 The Resilient Backpropagation Algorithm (RPROP).....	72
4.1.3 Objective Function for ANN Model.....	77
4.2 Trend Analysis for the Proposed ANN Model.....	78

4.3 Group Error Analysis for the Proposed ANN Model against Other Investigated Models.....	85
4.4 Statistical and Graphical Comparisons of the Proposed ANN Model against Other Investigated Models	90
4.4.1 Statistical Error Analysis	90
4.4.2 Graphical Error Analysis of the Proposed ANN Model against Other Investigated Models	91
4.4.2.1 Cross-plots of the Proposed ANN Model against Other Investigated Models	91
4.4.2.2 Error Distributions of the Proposed ANN Model against Other Investigated Models	98
4.4.2.3 Residual Analysis Error Distributions of the Proposed ANN Model against Other Investigated Models	104
4.5 Development of AIM Model	109
4.5.1 Introduction.....	109
4.5.2 Summary of Model's Equation.....	110
4.6 Trend Analysis for the AIM Model	111
4.7 Group Error Analysis for the AIM Model against Other Investigated Models ...	113
4.8 Statistical and Graphical Comparisons of the Polynomial GMDH Model.....	115
4.8.1 Statistical Error Analysis	115
4.8.2 Graphical Error Analysis of the Polynomial GMDH Model	115
4.8.2.1 Cross-plots of the Polynomial GMDH Model.....	116
4.8.2.1 Error Distributions of the Polynomial GMDH Model against Other Investigated Models,.....	122
4.8.2.2 Residual Analysis Error Distributions of the Polynomial GMDH Model against all Investigated Models	126
4.9 Uncertainty Study	130
4.10 Sensitivity Analysis	132
4.11 Summary	133
CHAPTER 5.....	135
CONCLUSIONS AND RECOMMENDATIONS	135
5.1 Conclusions.....	135
5.2 Recommendations.....	136

REFERENCES	137
LIST OF PUBLICATIONS	152
APPENDIX A	153
THEORY OF ARTIFICIAL NEURAL NETWORKS AND ABDUCTIVE NETWORKS	153
APPENDIX B	178
PROGRAMS LIST	178
APPENDIX C	201
GRAPHICAL USER INTERFACE OF ANN MODEL	201
APPENDIX D	211
RESEARCH DATA.....	211

LIST OF FIGURES

Fig 2.1:	Major Structure of Biologic Nerve Cell, reprinted with permission [James and David, 1991].....	31
Fig 2.2:	Artificial Neuron, reprinted with permission [James and David, 1991]	31
Fig 3.1:	Methodology Chart for Models Generation.....	43
Fig 3.2:	A simplified SCADA System	44
Fig 3.3:	Finding outliers using linear regression, reprinted with permission [Tamraparni and Johnson, 2003].	46
Fig 4.1:	Effect of Changing Number of Neurons in First Hidden Layer on Average Absolute Percent Error	71
Fig 4.2:	Effect of Changing Number of Neurons in First Hidden Layer on Correlation Coefficient.	71
Fig 4.3:	Effect of Changing Number of Neurons on Average Absolute Percent Error using Resilient Back-Propagation Training Algorithm.....	74
Fig 4.4:	Effect of Changing Number of Neurons on Maximum Error for each set using Resilient Back-Propagation Training Algorithm.....	74
Fig 4.5:	Effect of Changing Number of Neurons on Correlation Coefficient for each set using Resilient Back-Propagation Training Algorithm.....	75
Fig 4.6:	Effect of Changing Number of Neurons on Root Mean Square Errors for Testing and Validation sets using Resilient Back-Propagation Training Algorithm.....	76
Fig 4.7:	Effect of Changing Number of Neurons on Standard Deviation of Errors for Testing and Validation sets using Resilient Back-Propagation Training Algorithm.....	76
Fig 4.8:	Schematic Diagram of the Developed ANN Model.	77
Fig 4.9:	Effect of Gas Rate on Pressure Drop at Four Different Angles of Inclination.	79
Fig 4.10:	Effect of Oil Rate on Pressure Drop at Four Different Angles of Inclination.	80

Fig 4.11: Effect of Water Rate on Pressure Drop at Four Different Angles of Inclination.	80
Fig 4.12: Effect of Pipe Diameter on Pressure Drop at four Different Angles of Inclination.	82
Fig 4.13: Effect of Pipe Length on Pressure Drop at four Different Angles of Inclination.	82
Fig 4.14: Effect of Angle of Inclination on Pressure Drop at Four Different Pipe Diameters.	83
Fig 4.15: Effect of Changing Oil Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.	84
Fig 4.16: Effect of Changing Gas Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.	84
Fig 4.17: Effect of Changing Water Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.	85
Fig 4.18: Effect of Changing Pipe Length for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.	86
Fig 4.19: Statistical Accuracy of Pressure Drop Grouped by Oil Rate (With Corresponding Data Points).	87
Fig 4.20: Statistical Accuracy of Pressure Drop Grouped by Gas Rate (With corresponding Data Points).	87
Fig 4.21: Statistical Accuracy of Pressure Drop Grouped by Water Rate (With Corresponding Data Points).	88
Fig 4.22: Statistical Accuracy of Pressure Drop Grouped by Pipe Diameter (With Corresponding Data Points).	89
Fig 4.23: Statistical Accuracy of Pressure Drop Grouped by Pipe Length (With Corresponding Data Points).	89
Fig 4.24: Statistical Accuracy of Pressure Drop Grouped by Angle of Inclination (With Corresponding Data Points).	90
Fig 4.25: Cross-plot of Predicted vs. Measured Pressure Drop for Xiao et al. Model.	92
Fig 4.26: Cross-plot of Predicted vs. Measured Pressure Drop for Gomez et al. Model.	92

Fig 4.27: Cross-plot of Predicted vs. Measured Pressure Drop for Beggs and Brill Model.....	93
Fig 4.28: Cross-plot of Predicted vs. Measured Pressure Drop for Training Set (Proposed ANN Model).....	93
Fig 4.29: Cross-plot of Predicted vs. Measured Pressure Drop for Validation Set (Proposed ANN Model).....	94
Fig 4.30: Cross-plot of Predicted vs. Measured Pressure Drop for Testing Set (Proposed ANN Model).....	94
Fig 4.31: Comparison of Correlation Coefficients for the Proposed ANN Model against other Investigated Models.	97
Fig 4.32: Comparison of Root Mean Square Errors for the Proposed ANN Model against other Investigated Models.	97
Fig 4.33: Comparison of Average Absolute Percent Relative Errors for the Proposed ANN Model against other Investigated Models.	98
Fig 4.34: Error Distribution for Training Set (Proposed ANN Model).....	99
Fig 4.35: Error Distribution for Validation Set (Proposed ANN Model).	100
Fig 4.36: Error Distribution for Testing Set (Proposed ANN Model).	100
Fig 4.37: Error Distribution for Gomez et al. Model	101
Fig 4.38: Error Distribution for Beggs and Brill Correlation.....	102
Fig 4.39: Error Distribution for Xiao et al. Model	102
Fig 4.40: Residual Graph for Training Set (Proposed ANN Model).	106
Fig 4.41: Residual Graph for Validation Set (Proposed ANN Model).	106
Fig 4.42: Residual Graph for Testing Set (Proposed ANN model).....	107
Fig 4.43: Residual Graph for Gomez et al. Model.	107
Fig 4.44: Residual Graph for Beggs & Brill Correlation.	108
Fig 4.45: Residual Graph for Xiao et al. Model.....	108
Fig 4.46: Schematic Diagram of the Proposed AIM Topology	110
Fig 4.47: Effect of Angle of Inclination on Pressure Drop	112
Fig 4.48: Effect of Pipe Length on Pressure Drop at four Different Angles of Inclination.....	113
Fig 4.49: Statistical Accuracy of Pressure Drop for the Polynomial GMDH Model and other Investigated Models Grouped by Pipe Length (With Corresponding Data Points).....	114

Fig 4.50: Statistical Accuracy of Pressure Drop for the Polynomial GMDH Model and other Investigated Models Grouped by Angle of Inclination (With Corresponding Data Points)	115
Fig 4.51: Cross-plot of Predicted vs. Measured Pressure Drop for Training Set (Polynomial GMDH Model).....	117
Fig 4.52: Cross-plot of Predicted vs. Measured Pressure Drop for Validation Set (Polynomial GMDH Model).....	117
Fig 4.53: Cross-plot of Predicted vs. Measured Pressure Drop for Testing Set (Polynomial GMDH Model).....	119
Fig 4.54: Comparison of Correlation Coefficients for the Polynomial GMDH Model against All Investigated Models	120
Fig 4.55: Comparison of Root Mean Square Errors for the Polynomial GMDH Model against All Investigated Models	120
Fig 4.56: Comparison of Average Absolute Percent Relative Errors for the Polynomial GMDH Model against All Investigated Models	121
Fig 4.57: Comparison of Standard Deviation for the Polynomial GMDH Model against All Investigated Models	121
Fig 4.58: Error Distribution for Training Set (Polynomial GMDH Model)	123
Fig 4.59: Error Distribution for Validation Set (Polynomial GMDH Model)	124
Fig 4.60: Error Distribution for Testing Set (Polynomial GMDH Model)	125
Fig 4.61: Residual Graph for Training Set (Polynomial GMDH Model)	128
Fig 4.62: Residual Graph for Validation Set (Polynomial GMDH Model)	128
Fig 4.63: Residual Graph for Testing Set (Polynomial GMDH Model)	129
Fig 4.64: Estimated Pressure Drop against Actual Pressure Drop for the Testing Set of ANN Model with 95% Confidence Interval.....	131

LIST OF TABLES

Table 2-1:	Summary of models performance at 1-inch pipe diameter, reprinted with permission (Hong. Y. and Zhou study 2008)	22
Table 2-2:	Summary of models performance at 2-inch pipe diameter, reprinted with permission (Hong. Y. and Zhou study 2008)	23
Table 2-3:	Statistical Comparison of all Investigated Models, reprinted with permission (Park and Kang 2006)	38
Table 4-1:	Effect of Changing Number of Neurons with respect to Average Absolute Percent Error and Correlation Coefficient	72
Table 4-2:	Effect of Changing Number of Neurons with respect to Maximum Error for Testing, Training, and Validation Sets.....	73
Table 4-3:	Effect of Changing Number of Neurons with respect to Root Mean Square Error and Standard Deviation of Errors.....	73
Table 4-4:	Statistical Analysis Results of the Proposed ANN Model.....	91
Table 4-5:	Statistical Analysis Results of Empirical Correlations, Mechanistic Models, and the Proposed ANN Model	96
Table 4-6:	Evaluating Models Performance by Average Absolute Percent Errors and Correlation coefficient	104
Table 4-7:	Evaluating Models Performance by Root Mean Square Errors and Standard Deviation of Errors	104
Table 4-8:	Residual limits of the Proposed ANN Model against the Best Investigated Models.....	109
Table 4-9:	Statistical Analysis Results of the Polynomial GMDH Model.....	116
Table 4-10:	Statistical Analysis Results of Empirical Correlations, Mechanistic Models, against the Two Developed AIM & ANN models	122
Table 4-11:	Evaluating Models Performance by Average Absolute Percent Errors and Correlation coefficient (Including GMDH Model).....	126
Table 4-12:	Evaluating Models Performance by Root Mean Square Errors and Standard Deviation of Errors (Including GMDH Model).....	126

Table 4-13: Residual limits of the Polynomial GMDH &ANN Models against the Best Investigated Models.....	129
Table 4-14 : Uncertainty of Input Variables Used in Testing Models	130
Table 4-15: Uncertainty Values of All Models	131
Table 4-16: Relative Importance of Input Variable on ANN Model	132

LIST OF ABBREVIATIONS

A.G.A	American Gas Association
AAPE	Average Absolute Percent Relative Error
adaline	Adaptive Linear Element
AICC	Akaike's Information Criterion (used by GMDH algorithm)
AIM	Abductory Induction Mechanism
ANN	Artificial Neural Networks
APE	Average Percent Relative Error
critNum	Defines the criterion for evaluation of neurons and for stopping (used by GMDH algorithm)
D	Defines the number of input variables in the training data set (used by GMDH algorithm)
decNumNeurons	Defines the procedure of how the number of neurons are decreased (used by GMDH algorithm)
delta	Defines how much lower the criterion value of the network's new layer must be comparing the network's preceding layer (used by GMDH algorithm)
DP	Pressure Drop
<i>DPDLSystem</i>	Pressure drop evaluation software
ERNN	Elman Recurrent Neural Network
GDR	Generalized Delta Rule
GMDH	Group Method of Data Handling
GOR	Gas-Oil-Ratio

ID	Inner Diameter
IFP	French Institute of Petroleum (Institut FranCais du Petrole)
inputsMore	Defines the procedure of neuron connection (used by GMDH algorithm)
JOGMEC	Japan Oil, Gas and Metals National Corporation
JRNN	Jordan Recurrent Neural Network
KFUPM	King Fahd University of Petroleum and Minerals
LMS	Least Mean Square
madaline	Multiple-Adaline
maxNumInputs	Maximum number of inputs for individual neurons (used by GMDH algorithm)
maxNumNeurons	Defines the maximal number of neurons in a layer (used by GMDH algorithm)
MDL	Minimum Description Length (used by GMDH algorithm)
MGA	Mandhane, Gregory & Aziz correlation
MPF	Multiphase Flow
numLayers	Defines the number of layers in the network (used by GMDH algorithm)
OLGA	Oil and Gas Multiphase Simulator
p	Defines the degree of polynomials in neurons (used by GMDH algorithm)
PEPITE	Pressure and Temperature calculations Software in pipelines
PNN	Polynomial Neural Network
PVT	Pressure-Volume-Temperature
R	Correlation Coefficient
RMSE	Root Mean Square Error

RNN	Recurrent Neural Network
RPROP	Resilient Back-Propagation
RTU	Remote Terminal Unit
SCADA	Supervisory Control and Data Acquisition
SD	Standard Deviation
SINTEF	Stiftelsen for Industriell Og Teknisk Forskning
SQL	Structured Query Language
TACITE	Transient Multi-Component, Multiphase Flow Simulation Tool
time	Execution Time (in seconds), set (used by GMDH algorithm)
TUFFP	Tulsa University Fluid Flow Projects
VMP	Virtual Measurement in Pipes
WC	Water-Cut
WELLSIM	An Integrated Geothermal Wellbore Simulator and Analysis Package
Xtr	Training Data Variables
Xv	Validation Data Variables
Ytr	Training Target
Yv	Validation Target

NOMENCLATURE

Symbol	Definition
η^-	Factor between 1 and 0
θ	Dip Angle from Horizontal Direction, degrees
θ_j^h	Bias Term
μ_G	Gas Viscosity
μ_L	Liquid Viscosity
μ_{H_2O}	Water Viscosity at the Temperature of Some Actual Fluid
μ_m	Mixture Viscosity
v_m	Mixture Velocity
v_{sg}	Superficial Gas Velocity
ρ_G	Gas Density
ρ_L	Liquid Density
ρ_m	Mixture Density
ρ_s	Slip Density
σ	Standard Deviation of the Transfer Function
ϕ	Activation Function
a	Slope Parameter of the Sigmoid Function
Φ_{pj}	Threshold
y_{pk}	Desired Output Value from the Kth Unit.
$(\Delta P)_{est}$	Estimated Value of Pressure Drop
$(\Delta P)_{meas}$	Actual Value of Pressure Drop
$(dP/dZ)_{acc}$	Acceleration Pressure Drop
$(dP/dZ)_{elv}$	Pressure Gradient due to Elevation
$(dP/dZ)_f$	Pressure Gradient due to Friction
$(dP/dZ)_{TOT}$	Total Pressure Gradient
$\Delta_{ij}(t)$	Individual Weight Update-Value
Δ_{max}	Maximum step size by Resilient Back-propagation Algorithm
Δ_{min}	Minimum step size by Resilient Back-propagation algorithm

A	Cross Sectional Area
CV	Covariance
D	Diameter
d_k	Desired Output
E_a	AAPE (Average Absolute Percentage Error)
E_i	Relative Deviation of an Estimated Value from an Experimental Value
e_i	Residual
\hat{e}_i	Semi-Studentised Residual (or Standard Residual)
E_K	Dimensionless Kinetic-Energy Pressure Gradient
E_{\max}	Maximum Absolute Percent Relative Error
E_{\min}	Minimum Absolute Percent Relative Error
E_p	Total Error
e^s	Ratio of the Two-Phase to No-Slip Friction
FF	Dimensionless Parameter (Calculated in Equation 2.2)
f_i	Interfacial Friction Factor
f_n	No Slip Factor
$f_{t,p}$	Two-Phase Slip Factor
g	Acceleration of Gravity, ft/sec ²
g_c	Gravitational Conversion Factor = 32.17, ft-lbm/lbf-sec ²
h	Subscript Refers to the Quantities on the Hidden Layer
k	A constant Threshold Function Equal 1 or 0
k	Subscript Refers to the k^{th} Output Unit
L	Pipe Length, ft
L_α	Lower Bound of Confidence Interval
m_G	Mass of Flow Rate of Gas
$m-n-1$	Represents the Degree of Freedom in Multiple- Regression
MSE	Mean Square Error of the Data
NET_{pj}	Linear Combiner Output (in Neuron Equation)
o	Superscript Refers to Quantities of the Output Layer Unit
o_{pk}	Actual Output Value from the k^{th} Unit
P	Perimeter. L
p	Subscript Refers to the p^{th} Training Vector

P_1	Indicates the Upstream Two-Phase Condition
P_2	Indicates the Downstream Two-Phase Condition
Q	Flow Rate
Re	Reynolds Number
Re_m	Mixture Reynolds Number
r_j^2	External Criterion or Regularity Criterion (Root Mean Squared Values)
RSD	Relative Standard Deviation
T	a Constant Threshold Value
VFL	Volume Fraction of the Liquid (Calculated in Equation 2.1)
V'_G	The Volume of Gas of a particular Mass Assuming that the Density is that of Air at 70 ⁰ F., in Cubic Feet/Second
V'_L	The Volume of Liquid of a particular Mass Assuming that the Density is that of Water at 70 ⁰ F., and 14.7 psi Absolute, in Cubic Feet/Second
V_G	The Volume of Gas, in Cubic Feet/Second
V_L	The Volume of Liquid, in Cubic Feet/Second
U_α	Upper Bound of Confidence Interval
w_{ji}^h	Weight of the Connection from the i th Input Unit
$w_{j1}, w_{j2}, \dots, w_{jk}$	Synaptic Weights of Neuron j
x_5	Length of the pipe, ft
x_7	Angle of Inclination, Degrees
x_8	Wellhead Pressure, psia
$x_{p1}, x_{p2}, \dots, x_{pN}$	Input Signals
y	Simulated Pressure Drop by GMDH Model.
y_k	Actual Output
y_{pj}	Output Signal of the Neuron
α	liquid Holdup
ϵ_k	General Error for the k^{th} Input Vector
η^+	Factor >1
σ_i	Uncertainty of Variable i

CHAPTER 1

INTRODUCTION

1.1 Overview

Two phase flow phenomenon; namely liquid and gas, or what is synonymously called Multiphase flow (MPF), occurs in almost all upstream oil production, as well as in many surface downstream facilities.

It can be defined terminologically as a concurrent flow of a stream containing a liquid hydrocarbon phase (crude oil or condensate), a gaseous phase (natural gas, and non hydrocarbon gases), a produced water phase, and solids phase (wax, asphaltene sand, or even hydrates). Usually the amount of solid phase can be neglected because of its low contribution in the stream line.

This process has raised considerable attention from nuclear and chemical engineering disciplines as well as petroleum engineering. The phenomenon is governed mainly by bubble point pressure; whenever the pressure drops below bubble point in any point inside the production conduit, gas will evolve from liquid, and from that point to surface, multiphase gas-liquid flow will occur. Additional governing factor is the gas-liquid components and their changing physical characteristics along the pipe length and configuration with the change of temperature. Furthermore, certain flow patterns will develop while the pressure decreases gradually below the bubble point. The flow patterns depend mainly on the relative velocities of gas and liquid, and gas/liquid ratio. Needless to mention that sharp distinction between these regimes is quite intricate, [Ayoub, 2004]. However, multiphase flow mixture can be transported horizontally, vertically, or at any angle of inclination.

Furthermore, defining the pressure profile as a general case for all these configurations has quite limitations in relation with changing liquid hold-up and flow patterns, slippage criterion, and friction factor determination. In addition to that, velocity profile of each phase is hard to determine inside the pipe. The pressure drop (DP) mainly occurs between wellhead and separator facility. It needs to be estimated with a high degree of precision in order to execute certain design considerations. Such considerations include tubing size and operating wellhead pressure in a flowing well; direct input for surface flow line and equipment design calculations, [Ayoub, 2004]. Determination of pressure drop is very important because it provides the designer with the suitable and applicable pump type for a given set of operational parameters. In addition, it can be used as a guideline for the operational cost estimation in terms of pipeline sizing. Generally, the proper estimation of pressure drop in pipeline can help in design of gas-liquid transportation systems.

1.2 Motivation of Study

The need for accurate pressure drop estimation in multiphase flow piping is of great importance in the oil industry. Basically, it is well known that pipeline system is offering a cheapest way for transporting unprocessed raw crude oil and gas to separation stations with minimum maintenance costs. Long-multiphase flow lines have high pressure losses which affect the design of the whole system. The lines connecting the wellhead and separator facility should be well-sized in order to minimize the total system pressure drop, [Eaton *et al.*, 1967].

However, prediction of pressure drop is quite complicated and problematical due to the complex relationships between the various parameters involved. These parameters include pipe diameter, slippage of gas past liquid, fluid properties, and the flow rate of each phase inside the pipeline. Another parameter, which adds to the difficulty, is the flow patterns and their transition boundaries inside the pipe along with the heat exchange across boundaries of these patterns. Therefore, an accurate analytical solution for this problem is difficult.

There is a pressing need for estimating the pressure drop in pipeline systems using a simple procedure that would eliminate the tedious and yet the inaccurate and cumbersome methods.

Numerous attempts have been tried since the early fifties to come up with precise procedures to estimate pressure drop in multiphase flow pipes using conventional ways. The latter, were managed through the application of empirical correlations and mechanistic models, [Beggs, H. D. and Brill, 1973]. Previous attempts fail to provide satisfactory accuracy for estimation of pressure drop in multiphase flow pipe systems. Most of these correlations were derived for two phase flow and none of them had accounted for the water phase, which may add to the difficulty and accuracy of modeling. These correlations and mechanistic models had been used by the industry despite of their low accuracies because there is no alternative. The conventional approach proved to be unsuitable for dealing with highly complex problem.

Empirical correlations were derived from limited set of laboratory data, which are susceptible to produce erroneous results when scaled up to oilfield. Many of these correlations exhibit large discontinuities at the flow pattern transitions. This can lead to convergence problems when these models are utilized for simultaneous simulation of petroleum reservoir and associated production facilities, [Aziz and Petalas, 1994]. Mechanistic models are following the semi-empirical approach, which are based on physical phenomenon and conservation of mass and energy. Also, most of the mechanistic models in literature are either incomplete (they consider only the flow patterns determination), [Taitel *et al.*, 1980], or they have limitation in their applicability to certain angles of inclination, [Ansari *et al.*, 1994] and [Xiao *et al.*, 1990].

Thus, there is a pressing need for accurate modeling of pressure drop in pipeline systems under multiphase flow conditions using real field data. This should be done by using the most relevant data and the right technique. This can be achieved through the application of the latest statistical and computing technique which will be able to discover the highly nonlinear relationship between relevant input parameters and the output.

1.3 Approach

The approach that will be followed to model the pressure drop for pipeline system with a wide range of inclination angles is through the Artificial Neural Network (ANN) and Abductory Induction Mechanism (AIM).

Neural Networks technique recently has gained enormous popularity, especially in Petroleum Engineering, [Mohaghegh and Ameri, 1995]. Its ability in differentiating between parametric and non-parametric relationship makes it a successful means for solving hard-known problems. This technique has the ability to acquire, store, and utilize experiential knowledge. Besides, it can differentiate, depending on the training data set, between complex patterns if it is well trained, [Haykin, 1994].

In this study, an artificial neural network model for prediction of pressure drop in pipelines carrying multiphase fluids will be developed and tested against real field data from selected fields. Neural Networks will be utilized in attempt at this study to produce a generic model for predicting pressure drop in multiphase flow pipes that accounts for a wide range of angles of inclination.

However, ANNs suffered major drawbacks such as network usually stuck in local minima; defining the optimum network structure (in terms of number and size of the hidden layers and defining the optimum transfer function) is human-biased; over-fitting and poor network generalization are clearly evident. However, part of the used data will be reserved for validation purposes. This reservation lessens the amount of trained data which is precious in light of scarce data nowadays. Additionally, some users cannot come up with a clear conclusion about how the model performs. In general, they treat the produced model as a black box which reduces its value.

In order to overcome such limitations, a new approach has been developed by a Ukraine scientist named Alexy G. Ivakhnenko, which has gained wide acceptance in the past few years called Group Method of Data Handling (GMDH) or Abductory Induction Mechanism (AIM) will be utilized, [Osman, E.A. and Abdel-Aal, 2002]. In brief, GMDH approach is a formalized paradigm for iterated (multi-phase) polynomial regression capable of producing a high-degree polynomial model in effective predictors. The process is evolutionary in nature, using initially simple

regression relationships to derive more accurate representations in the next iteration. To prevent exponential growth and limit model complexity, the algorithm only selects relationships having good predicting powers within each phase. Iterations will stop when the new generation regression equations start to have poor prediction performance than those of previous generation. The algorithm has three main elements; representation, selection, and stopping. It applies abduction heuristics for making decisions concerning some or all of these three steps, [Osman, E.A. and Abdel-Aal, 2002].

1.4 Objectives of the Research

The overall objective of this study is to minimize the uncertainty in the multi-phase pipeline design by developing representative models for pressure drop determination in downstream facilities (gathering lines) with the use of the most relevant input variables and with a wide range of angles of inclination.

Two approaches will be utilized to achieve the overall objectives; the artificial neural network (ANN) and the Abductory Induction Mechanism (AIM) techniques. Data from selected different fields from Middle East will be used in this study. Specific objective is:

1. To construct and test two models for predicting pressure drop in pipeline systems under multiphase flow conditions with real field data for a wide range of angles of inclination (from -52° to 208°) using ANN and AIM techniques.

1.5 Benefits of the Research

The benefits of the current research to the oil and gas industry can be summarized as follows:

1. Modelling of pressure drop in pipeline system can aid in offering sound design considerations for the pipeline engineer and designer in terms of choosing the best pumping components of the system that are consistent with the physical properties.

2. Determining the most relevant and influential input parameters involved in estimating pressure drop can improve the modelling procedure. This can be done through the automated framework to exploit information inherent in modelled data sets in order to estimate the pressure drop by GMDH approach. This helps reduce the curse of dimensionality, which is greatly affecting modelling running time, overfitting, suspected collinearity and numerical instability, [Verleysen and François, 2005].
3. Investigating the potential of using ANN and AIM techniques in this new area, while no past research had been conducted to model such a feature (generic models).
4. Exploring the suitability of the best current empirical correlations and mechanistic models in estimating pressure drop in pipeline systems with a wide range of angle of inclinations and under field conditions.
5. ANN model will serve as a new “tool” to be used by the oil & gas industry to aid in estimating pressure drop in pipeline systems with wide range of angles of inclination.
6. GMDH model will serve as an easy and applicable mathematical correlation with the most relevant input parameters to the pressure drop target.

1.6 Summary

This chapter introduced main concepts related to multiphase phenomenon. Main problems encountered during pressure drop estimation using the conventional methods (empirical correlations and mechanistic models) have been thoroughly discussed. Additionally, the motivation behind conducting this study has been stated. The approach that will be followed to solve the problem has been, in brief, highlighted. In addition, the general and specific objectives have been clearly stated. Finally, the Chapter concluded with stating the benefits that could be acquired as a result from this research.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

This part of the research deals with the revision of the most commonly used correlations and mechanistic models and their drawbacks in estimating pipeline pressure drop in multiphase flow. The science of multiphase flow is broad and so many studies have been conducted in many relevant subtopics since early 1950's. The main concern was the prediction of flow pattern, liquid void fraction (liquid holdup), and pressure drop. However, prediction of all these parameters is necessary for optimum design of gas-liquid systems that is consistent with the physical and hydrodynamic properties of mixture. Furthermore, the results obtained by the study of the Beggs and Brill, [Beggs, H. D. and Brill, 1973] showed that accurate prediction of pressure drop and liquid holdup requires optimum evaluation of two-phase flow patterns and pipe inclination.

This important finding stimulates carrying on this research and to investigate the effect of pipe inclination through generation of models for estimating the pressure drop while taking into consideration all possible pipe configurations and the available data set. It is worthy to mention that no single study in the literature could be found presenting pressure drop estimation in pipelines under multiphase conditions using artificial neural networks or Abductive networks and taking into consideration wide range of angles of inclination. In this chapter, only publications from literature that have pronounced major contribution to this study will be reviewed. Special emphasis will be given to Beggs & Brill correlation, because it has been designed originally to be applied for all angles of pipe inclination, [Beggs, H. D. and Brill, 1973].

Additional prominence will be devoted to some mechanistic models, which show reliable performance in estimating pressure drop by industry. The use of the steady state simulator (state of the art) will be also presented as a common solution adopted by the industry. The concepts of artificial neural network and Abductive network are being presented along with their applications in petroleum industry as well as in multiphase flow area.

2.2 Introduction

Multiphase flow panacea is quite complex since the problem has no analytical solution. Numerous factors are contributing to the nature of this problem such as slippage of the gas past the oil, change of the flow patterns with decreasing pressure to the surface, and mass transfer change between coexisting phases. Two schemes had been proposed in literature to solve this problem, namely empirical correlations and mechanistic models.

The first approach had been conceived in 1940's, [Lockhart and Martinelli, 1949] and was based on experimental observations and limited laboratory data. The main target of this approach had to meet certain individual design considerations. The second (semi-empirical) approach called mechanistic modeling had appeared in the early 1980's, [Gomez *et al.*, 1999] which had been based on combining the resulting steady state equations and experimental data of multiphase fluids. This approach received wide acceptance from the oil industry since it was adopting the physical phenomenon and conservation of mass and energy principles.

There are many correlations and mechanistic models used for estimating pressure drop in pipelines. However, only few of them are designed to estimate the pressure drop at all angles of inclination. Researchers had noticed that most of these correlations were developed under laboratory conditions and are, consequently, inaccurate when scaled-up to oil field conditions, [Tackacs, 2001].

Empirical correlations fail to address the true and complex behavior of multiphase flow since adding more data to the latest empirical models resulted in no

improvement in accuracy of pressure drop estimation and design of multiphase systems. Application of empirical correlations to a broad range of data usually results in errors in the range of $\pm 20\%$ in pressure drop prediction, [Brill, 1987].

2.3 Empirical Correlations

2.3.1 Background

Numerous correlations have been developed since the early 1940's, [Lockhart and Martinelli, 1949] on the subject of vertical and horizontal multiphase flow, as well as for inclined flow. These empirical correlations were initially developed either for pipelines or wellbores. The work of multiphase flow was initiated originally in the nuclear industry where several findings were utilized in resolving many petroleum related problems. Pressure gradient was estimated initially in which the effect of slip (gas traveled with higher velocity compared to liquid) and the flow regime were disregarded. The no slip approach has a tendency to underestimate pressure drop because the volume of liquid predicted to be produced through the well was smaller than its true value. Advances to the no-slip methods utilized empirical liquid holdup correlations to account for slippage between the liquid and gas phases, [Brill and Mukherjee, 1999].

A revolutionary step was the generation of flow pattern map for simultaneous flow of oil and gas, [Baker, O., 1954]. However, no usage of this flow pattern map was reported in the literature for estimating the multiphase pressure gradient in horizontal pipes. A similar approach was presented for estimating pressure drop in systems slightly inclined from horizontal (hilly terrain), [Flanigan, 1958]. Flanigan's method was proposed for long transmission lines in hilly terrain and ignores any pressure gain in downhill sections. The correlation is limited to pipe diameters of 4 to 10 inches only. The last two approaches suffered great inaccuracy as tested by several authors later, [Brookbank and Fagiano, 1975], [Hong and Zhou, 2008], [Al-Ne'aim *et al.*, 1995]. The flow of air and water mixtures through a 0.8245 inch-diameter pipe inclined at angles of ± 90 , 60, 30, 15, 10, 5, and at 0 degrees, was studied [Sevigny,

1962]. The author came up with a correlation for two-phase friction factor as a function of input liquid content, gas Reynolds number, and liquid Reynolds Number as follows:

$$VFL = \frac{V_L'}{V_G' + V_L'} \quad (2.1)$$

and

$$FF = \frac{\frac{m_G}{\mu_G} - \frac{m_L}{\mu_{H_2O}}}{\frac{m_G}{\mu_G} + \frac{m_L}{\mu_{H_2O}}} \exp \left[\frac{2g_c DA^2}{4(V_G' + V_L')} \cdot \frac{V_G' P_1 \ln \left(\frac{P_1}{P_2} \right) + V_L' (P_1 - P_2)}{m_G + m_L} + (z_2 - z_1) \frac{g}{g_c} \right]^{\sqrt{VFL}} \quad (2.2)$$

The volume fraction of the liquid has been plotted against FF, (the dimensionless parameter) and the produced curve had been verified according to different situations. The precision of this correlation was extremely doubtful since many parameters were omitted such as holdup and the effect of elevation change in the friction term was not accounted for in pressure drop calculation.

A substantial contribution was reported when first flow pattern dependent approach for vertical multiphase flow in tubes launched, [Duns and Ros, 1963]. The authors identified 13 important variables, which result in 10 dimensionless groups that helped described multiphase flow behavior. In their work, they classified the flow patterns into three regions in which they found slip factor, slip velocity, liquid holdup, friction factor and static gradient due to mixture flowing density (in the mist flow region only) were the most important parameters for each phase. The authors concluded that four dimensionless groups were important for predicting flow patterns and degree of slippage. Later, several researchers have utilized their main findings to add some improvement to their own correlations, [Baker, A. *et al.*, 1988], [Ilobi and Ikoku, 1981].

The first work reported in the literature for pipelines correlations was the model generated by [Dukler *et al.*, 1964]. This model suffered a great shortcoming where an effective two-phase flow friction factor was applied in approximately calculating the

multiphase pressure gradient. In addition, an empirical correlation for the liquid holdup was employed rather than varying it with flow patterns.

The studies by Beggs (1972) and Beggs and Brill (1973) [[Beggs, H. D., 1972] and [Beggs, H. D. and Brill, 1973]] are the most comprehensive applicable correlation originally designed for inclined pipelines with or without water-cut and are probably the best choice available for deviated wells. Beggs & Brill Model was derived from a huge number of database (584 data points) but in a small scale test facility where air and water were used as testing fluids and with 1 inch and 1.5 inches diameter pipes. The model was generated to serve for all angles of inclination ranging from -90° to 90° . The factors used for correlating are gas flow rate, liquid flow rate, pipe diameter, inclination angle, liquid holdup, pressure gradient and horizontal flow regime. The correlation was designed for horizontal wells and later modified to account for wells' inclinations. The flow pattern is predicted initially while different constants have been formulated for different flow regimes providing that the flow is horizontal. A horizontal correlation was then generated and adjusted accordingly to match different angles of inclination through using a correction factor. In their approach, Beggs and Brill suggested the use of mixture fluid properties to determine the friction factor. The mixture concept was being widely accepted by the industry for the prediction of pressure loss for homogenous flow. A two-phase friction factor was calculated independent of flow regime but depends on liquid holdup. Hold up factor was calculated as a function of horizontal hold up. Flow patterns also were determined using dimensionless groups.

The no-slip friction factor was determined from the smooth pipe curve on a moody diagram or from;

$$f_n = 1 / \left[2 \log \left(\frac{R_e}{4.5223 \log R_e - 3.8215} \right) \right]^2 \quad (2.3)$$

Mixture Reynolds number was given by;

$$Re_m = \frac{\rho_m v_m D}{\mu_m} \quad (2.4)$$

While mixture density and mixture viscosity were defined using the formula described by the following;

$$\rho_m = \alpha\rho_L + (1 - \alpha)\rho_G \quad (2.5)$$

$$\mu_m = \alpha\mu_L + (1 - \alpha)\mu_G \quad (2.6)$$

Where α was the void fraction (liquid holdup).

The ratio of the two-phase to no-slip friction was calculated from;

$$\frac{f_{t,p}}{f_n} = e^s \quad (2.7)$$

Where:

$$s = \left[\frac{\ln(y)}{-0.0523 + 3.182 \ln(y) - 0.8725[\ln(y)]^2 + 0.01853[\ln(y)]^4} \right] \quad (2.8)$$

And y can be defined as;

$$y = \frac{\lambda_L}{[H_{L(\phi)}]^2} \quad (2.9)$$

The value of s became unbounded at a point in the interval $1 < y < 1.2$, then the function s can be calculated as;

$$s = \ln(2.2y - 1.2) \quad (2.10)$$

The acceleration pressure drop gradient is given by;

$$\left(\frac{dP}{dZ} \right)_{acc} = \frac{\rho_s v_m v_{sg}}{g_c P} \frac{dP}{dZ} \quad (2.11)$$

The acceleration term is defined as;

$$E_K = \frac{\rho_s v_m v_{sg}}{g_c P} \quad (2.12)$$

Then, the total pressure gradient can be calculated as stated in equation 2.13:

$$\left(\frac{dP}{dZ}\right)_{TOT} = \frac{\left(\frac{dP}{dZ}\right)_{elv} + \left(\frac{dP}{dZ}\right)_f}{1 - E_K} \quad (2.13)$$

$$\text{where } \left(\frac{dP}{dZ}\right)_{elv} = \frac{g}{g_c} \rho_s \quad (2.14)$$

With a range of conducted experimental investigation, the pressure losses were accurately estimated. Any further increase in tubing size tended to result in an over prediction in the pressure loss.

As mentioned previously, the most commonly used correlation for all angles of inclination reported in the literature is by Beggs & Brill only. This correlation has been evaluated and studied carefully by several investigators to validate its applicability under different ranges of data. Its performance was tested by several researchers and considered to be good for horizontal wells (the correlation underestimated the pressure drop by an error of 25%, which was attributed to the overprediction of liquid hold in downhill flow), [Payne *et al.*, 1979], and for vertical wells, [Stoisits *et al.*, 1999].

A comprehensive study focused on evaluating Beggs & Brill correlation, which indicated that the correlation is applicable for inclined wells with or without water-cut. The author recommended using the correlation for deviated wells, [Bharath, 1998].

Additional supportive study evaluated the performance of vertical multiphase flow correlations and the possibility of applying those set of correlations for conditions in Gulf region where large tubular and high flow rates are common, [Aggour *et al.*, 1994]. They concluded that Beggs & Brill correlation outperformed the rest of correlations in pressure prediction. However, a study conducted in Kuparuk

field (located in North Slope Borough, Alaska, United States) indicated that Beggs & Brill correlation predicted pressure drop within 10% accuracy for all production pipelines, [Stoisits *et al.*, 1999].

While a recent study showed that Beggs & Brill correlation always over predicted pressure gradients, [Yuan and Zhou, 2008]. The latter authors conducted a comparative study for many pressure prediction correlations and mechanistic models that had been widely used by the industry utilizing experimental data with seven angles of inclination. However, the authors claimed that their study can be used as a guideline for selecting two-phase flow pressure drop prediction correlation and mechanistic model in designing and analyzing downward two-phase flow pipelines.

Mukherjee and Brill (1985) published a correlation for 1.5-inch pipe and experimental pressure as low as 100 psig. The system consisted of air as gaseous phase and a combination of kerosene and lube oil as a liquid phase. Upward and downward flow measurements were taken at inclination angles from 0 and 90 degrees from horizontal. The system temperature varied between -7.8 to 55.56°C. Their correlation resulted in a good agreement with experimental data and other correlations and further verified. Their correlation managed to calculate the friction loss in four different flow regimes (Bubble and Slug together, annular, and stratified). Correlations for the first two groups will be provided herein.

For Bubble and Slug Flow Regimes:

$$\Delta P_f = \frac{f_n L \rho_m v_m^2}{2g_c d} \quad (2.15)$$

For Annular Flow Regime:

$$\Delta P_f = \frac{f_c L \rho_{ns} v_m^2}{2g_c d} \quad (2.16)$$

Where;

$$f_c = f_R * f_n \quad (2.17)$$

with data from Prudhoe Bay and North Sea, [Mukherjee and Brill, 1985]. In the same vein, Mukherjee & Brill and Beggs & Brill correlation's performances were evaluated by Arya and Thomas (1981). The purpose of their study was to measure the accuracy of three different correlations for liquid holdup and pressure drop across flow regime boundaries for horizontal and inclined pipes [Arya and Thomas, 1981]. The authors concluded that Mukherjee & Brill correlation exhibited optimum performance when compared to the Beggs & Brill correlation and Mandhane, Gregory & Aziz (MGA) correlation, [Mandhane *et al.*, 1974]. However, the authors commented that Mukherjee & Brill correlation showed some discontinuities for downhill flow.

Another study had been conducted by Abduvayt (2003) to measure the flow patterns, pressure drop and water holdup in oil water flow in horizontal, hilly-terrain ($\pm 0.5 < \theta < \pm 3$) pipe and vertical pipelines at a temperature of $35 (\pm 5) ^\circ\text{C}$ and a pressure of approximately 35.5 psi using the large-scale multiphase-flow test facility of Japan Oil, Gas and Metals National Corp. (JOGMEC). Additionally, test lines of 4.19-inches inner diameter (ID) and 120-m total length were used, which included a 40-m horizontal or hilly terrain (near-horizontal) and a 10-m vertical test section sequentially connected. The flow pattern was determined by visual observation with video recordings, and a flow-pattern map was made for each condition. The authors identified twelve flow patterns which were categorized into three basic classes as segregated, semi-segregated, and semi-dispersed flows. They analyzed the slippage between the phases using measured holdup plotted against input water-cut (WC) with oil flow rate as parameter and came up with a conclusion that slippage changed notably by slightly changing the inclination angle, [Abduvayt *et al.*, 2003].

Pressure drop estimation in vertical well was addressed coherently; where different empirical correlations and mechanistic models had been tested for their accuracy, [Ayoub, 2004]. The author reported that Mukherjee & Brill correlation outperformed other correlations and mechanistic models in terms of lowest average absolute percent error, lowest maximum error, lowest errors standard deviation, lowest average relative error, and the lowest root mean squared error.

The correlations developed by [Beggs, H. D. and Brill, 1973] and [Mukherjee and Brill, 1985] helped to improve pressure drop estimation in inclined wells and hilly terrain pipelines.

Only one study was found in literature that dealt with generating and evaluating pressure drop model at horizontal, vertical and inclined pipelines and wellbores, [Bilgesu, H. and Ternyik, 1994]. The authors constructed a model to estimate pressure drop, fluid properties, and flow pattern determination for multiphase flow and evaluated its performance using data from literature. The model had been evaluated against commercial software and showed good agreement with the tested data. The model predicted pressure drops had an average percent error of less than 2.0%. For horizontal and for the inclined pipes the average percent errors were 1.78% and -1.89%, respectively.

It is noteworthy that Beggs & Brill (1973) and Dukler et al. (1964) correlations are classified to be suitable models for simulating the flow in pipelines, [Beggs, H. D. and Brill, 1973] and [Dukler *et al.*, 1964]. While [Hagedorn and Brown, 1965], [Ros, 1961], and [Duns and Ros, 1963] are considered to be specific correlations for flow in wellbores.

2.4 Mechanistic Models

This is a semi-empirical approach that deals with addressing physical phenomena of multiphase flow. The mechanisms of multiphase flow are established using mathematical modeling approach. Each flow pattern and its transition phase are comprehensively studied using fluid dynamics. Such flow patterns are presented in horizontal, deviated, and vertical flow, [Gomez *et al.*, 1999].

The technique of mechanistic modeling has coupled the laboratory, field measurements and the most important factors affecting the multiphase mechanism. The prediction capability of these models is greatly enhanced when compared to the empirical correlations, [Petalas and Aziz, 2000].

These mechanistic multiphase flow models include three types of models; two-fluid model, the drift-flux model, and the homogenous model, [Manabe *et al.*, 2001]. In terms of model types; the models can be grouped into pipelines models and wellbores models. A critical literature survey will be focused on pipeline models and unified mechanistic models group only.

2.4.1 Pipeline Mechanistic Models

Pipeline models were initially designed for horizontal flow configurations. They can be extended to models with ± 10 degrees deviation from horizontal. Two researchers presented flow pattern map which was based on mechanistic modeling for horizontal and slightly inclined pipelines, [Taitel and Duckler, 1976]. To further continue their effort, another flow pattern map for vertical flow in pipes had been published [Taitel *et al.*, 1980]. Their model had been recognized by setting flow pattern transition for each phase utilizing a group of non-dimensional parameters. It is worthy mentioned that most of these flow patterns were based on experimental observations. They were consequently considered inappropriate if used for field conditions.

After their publication of flow regime map, [Taitel and Duckler, 1976] definite studies had been conducted to address the transition boundaries between each adjacent phases in order to benefit from their characteristics in designing internal separator facilities, [Wallis and Dobson, 1973], [Andritsos *et al.*, 1989]; and [Simmons and Hanratty, 2001]. Most of these studies concentrated on determining the onset of slug and stratified flow phases and their transition boundaries, both theoretically and experimentally.

2.4.1.1 Pipeline Mechanistic Models of Single Flow Regime

Different mechanistic models have been developed for each single flow regime. The mechanism of stratified flow has been addressed by several authors.

Defining the proper transition boundaries for flow patterns of a certain mechanistic model is being a great hindrance suffered by many authors. Taitel and

Duckler (1976) tried to overcome this obstacle by defining the transition boundaries for their mechanistic model that served for horizontal and near horizontal gas-liquid flow. Their model was based on the momentum balance equations for each phase in two phase stratified flow. As the stratified flow occurs at the entry of the pipe, where more predominantly encountered in downhill or horizontal pipe with relatively small flow-rates of both phases.

They started their model by addressing the criteria governing the change from stratified flow to other flow patterns. By making use of the physical concepts of the flow pattern transitions they were able to formulate different mathematical criteria for each transition pattern. The momentum balance equation has been solved using the previously formulated criteria, [Taitel and Duckler, 1976].

It is quite interesting to note that other authors assumed a turbulent liquid phase and employed the correlated eddy's viscosity concept to calculate the interfacial friction factor, [Cheremisinoff and Davis, 1979]. They proposed it as;

$$f_i = 0.008 + 0.00002Re_L \quad (2.18)$$

They found out that liquid Reynolds number up to 1700 is linearly dependent on interfacial friction factor.

Another interesting study was conducted to quantify the effect of stratified turbulent-turbulent gas liquid flow in horizontal and inclined pipes, [Shoham and Taitel, 1984]. The authors discarded the correlation suggested by [Cheremisinoff and Davis, 1979]. Instead, they proposed another constant value for the interfacial friction factor as;

$$f_i = 0.014 \quad (2.19)$$

The authors believed that for Reynolds number above 1700, unrealistic values of interfacial friction factor values were obtained.

Other models were investigating the phenomenon of three-layer stratified flow, where a three phase flow could be treated as three-layer flow.

The stratified flow pattern occurs with oil-water system at a relatively low flow-rate. This could be applied for immiscible liquid flowing in horizontal and slightly inclined pipelines with low flowrate [Hall, 1992], [Taitel *et al.*, 1995].

Khor *et al.* (1997) conducted a comprehensive study for modeling one-dimensional phase holdups in three phase stratified flow. They generated a computer code, which was called later (PRESEBAL) and used it to apply the-three-fluid model with a variety of assumptions. These assumptions varied between modeling of wall and interfacial shear stresses. As an outcome of their model, they managed to measure the interfacial shear stress from standard single-phase flow relationships. Moreover, they estimated phase holdups by comparing the pressure drops in each phase that was derived from the momentum balances. The desired solution is the point where all the three phases have the same pressure gradient. They concluded that the oil water interfacial shear stress value of 0.014 had given the best estimation of holdup, [Khor *et al.*, 1997].

Slug flow is the most common phase in producing wells. It is the most undesirable phase encountered inside the pipelines as well as in wellbores. It causes pressure fluctuations, tanks in surface facilities to flood, and increases the tendency of deposition and corrosion. Separate models addressed the mechanism governs this flow phase, [Kordyban and Ranov, 1970]. The authors suggested that Kelvin-Helmholtz instability is main cause of onset of slug flow.

A more general empirical correlation has been published for the slug gas liquid holdup, [Sylvester, 1987]. The author collected two sets of experimental data to generate his correlation, which was dependent on operational conditions.

Annular flow is highly prevailed in both gas condensate and geothermal producing wells. It is also common in oil wells especially during high-GOR production. This phase flow has been studied by many researchers. For instance, Lournat *et al.* (1985) conducted a study to investigate the effect of pipe size on annular flow of air and water in horizontal pipes. The authors were managed to develop a model for prediction of the distribution of the time averaged film thickness around the pipe circumference, [Laurinat *et al.*, 1985].

Paz and Shoham (1994) carried out an experimental and theoretical investigation on two-phase annular flow in inclined pipes. Their study focused on the effect of the inclination angle on the liquid film thickness distribution. The authors developed a simple analytical model for the prediction of the liquid film thickness at the top and bottom of the pipe. The authors claimed that their model can be applied for the entire range of inclination angles.

Good agreement is observed between the prediction of the model and the experimental data collected in this study and from the literature. A maximum error of 17% has been achieved by the model for vertical angles while different errors values ranging between 19.0% to 62.0% were reported by the model at different angles of inclinations, which indicated that the proposed model by the authors underpredicted the experimental data from both this study and other sources, [Paz and Shoham, 1994].

Hasan and Kabir (2005) generated a mechanistic model that necessitated the estimation of film thickness before computing frictional pressure-drop as gas flows past the wavy-liquid film surrounding the pipe wall. Their model investigated the film thickness and its impact on pressure-drop computation in wellbores producing steam-water, gas-condensate, and gas-oil mixtures. The authors confirmed that when the homogeneous model was used to compute pressure gradient by ignoring the wavy liquid film on frictional pressure-drop, good agreement was achieved with field data and those of a mechanistic model, [Hasan, R. and Kabir, 2005].

Dispersed bubble flow is distinguished by the no-slip phase behavior and uniform velocity distribution, [Wallis, 1969]. Dispersed bubble flow with slip phase behavior was also presented, [Manabe and Arihara, 1996].

A comprehensive mechanistic model for two-phase flow in horizontal and near-horizontal pipelines was presented by [Xiao *et al.*, 1990]. Their work integrated many modeling features such as flow pattern estimation and separate flow models. The flow variables addressed were pressure drop and liquid holdup. This pioneering effort was based on developing a unified mechanistic model that accounts for pipe angle inclination from horizontal (0^0) to vertical (90^0). The authors' contribution was to

develop a flow pattern prediction model and separate models to determine flow variables for the individual flow pattern.

Xiao et al. Model is a comprehensive mechanistic model designed for gas-liquid two phase flow in horizontal and near horizontal pipelines. Among its numerous benefits, the model can predict the pressure drop in pipeline with high degree of accuracy.

The model achieved the lowest average absolute average error among all tested models that reached 30.5%. The authors compared their model performance to other tested correlations and models and it showed superior capabilities. The model performance had been evaluated against a data bank collected from the A.G.A (American Gas Association) database and laboratory data published in literature. Beggs and Brill correlation was found to perform the best over three tested models named Dukler, et al., Dukler-Eaton, and Mukherjee and Brill.

The mechanistic model developed by Xiao et al. has been used as a base for another model expanded by other researchers, [Manabe and Arihara, 1996]. An experimental program was set up to cover all flow patterns. Three models were used for testing the mechanistic model performance. Beggs and Brill correlation ranked second after Dukler et al model. Mukherjee and Brill model was least accurate among the tested models.

Petalas and Aziz (2000) developed a comprehensive mechanistic model using a large set of data from Stanford Multiphase Database. Their model was able to identify flow regimes based on certain assumptions. Additionally, it is applicable to wide range of fluid properties and pipe geometries. The model also incorporated roughness effects as well as liquid entrainment, which were not considered by previous models. The authors finalized their effort by making the model able to calculate the pressure drop at any flow pattern and to calculate the liquid volume fraction efficiently, [Petalas and Aziz, 2000]. Hong and Zhou (2008) presented a comprehensive review of the applicability of some empirical and mechanistic models using commercial software. Data from published work have been used for this purpose, [Hong and Zhou, 2008].

Five empirical correlations and a single mechanistic model were chosen by the authors to compare their model's performance. Those are Beggs-Brill, Dukler-Eaton-Flanigan, Dukler-Flanigan, Dukler, Eaton, and Eaton-Flanigan correlations and Xiao et al. mechanistic model. The authors concluded that Beggs-Brill correlation always overestimates the pressure gradient in all studied cases.

However, for small pipe diameter with superficial-liquid velocities greater than 3 ft/sec the authors noticed that Dukler behaves the best, followed by Xiao and Eaton & Flangian. Moreover, at Superficial Liquid Velocities less than 3 ft/sec, they reported that Xiao behaves the best, followed by Eaton & Flangian and Eaton. for a pipeline with 2-inches in diameter the authors concluded that Xiao model was the best, followed by Eaton. Table 2.1 and Table 2.2 show summary of their study along with cumulative rating for each pipe diameter.

Table 2-1: Summary of models performance at 1-inch pipe diameter, reprinted with permission (Hong, Y. and Zhou study 2008)

Model Name Property	Beggs & Brill	Dukler-Eaton-Flanigan	Dukler-Flanigan	Dukler	Eaton	Eaton-Flanigan	Xiao
SLV range (ft/sec)	1.2 to 7	1.2 to 7	1.2 to 7	1.2 to 7	1.2 to 7	1.2 to 7	1.2 to 7
ANGLE	-1	-1	-1	-1	-1	-1	-1
AAPE	30.29	24.39	32.62	32.96	23.23	21.51	18.6
Rating	5	4	6	7	3	2	1
SLV range (ft/sec)	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10
ANGLE	-1	-1	-1	-1	-1	-1	-1
AAPE	39.19	56.71	38.94	27.44	23.81	23.65	15.12
Rating	6	7	5	4	3	2	1
SLV range (ft/sec)	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10	2 to 10
ANGLE	-9	-9	-9	-9	-9	-9	-9
AAPE	58.38	75.99	62.09	44.56	27.78	21.6	13.36
Rating	5	7	6	4	3	2	1
Cumulative Rating	16	18	17	15	9	6	3

A thorough revision of existing two-phase flow prediction models had been conducted, [Zhang, H. Q. *et al.*, 2003(a)]. The authors built up a unified hydrodynamic model to envisage certain criteria such as flow pattern transitions, pressure gradient and liquid holdup for all angles of inclination from -90° to 90° from horizontal.

Table 2-2: Summary of models performance at 2-inch pipe diameter, reprinted with permission (Hong, Y. and Zhou study 2008)

Model Name Property	Beggs & Brill	Dukler-Eaton-Flanigan	Dukler-Flanigan	Dukler	Eaton	Eaton-Flanigan	Xiao
SLV range (ft/sec)	1.2 to 6.56	1.2 to 6.56	1.2 to 6.56	1.2 to 6.56	1.2 to 6.56	1.2 to 6.56	1.2 to 6.56
ANGLE	-1	-1	-1	-1	-1	-1	-1
AAPE	58.33	65.33	60.13	52.58	38.47	58.47	16.18
Rating	4	7	6	3	2	5	1
SLV range (ft/sec)	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56
ANGLE	-1	-1	-1	-1	-1	-1	-1
AAPE	42.79	89.29	103.61	88.92	28.71	48.06	16.49
Rating	3	6	7	5	2	4	1
SLV range (ft/sec)	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56	1.64 to 6.56
ANGLE	-9	-9	-9	-9	-9	-9	-9
AAPE	114.14	146.9	176.7	154.72	46.18	88.71	76.59
Rating	4	5	7	6	1	3	2
Cumulative Rating	11	18	20	14	5	12	4

A new approach had tested the use of Artificial Neural Networks (ANN) to develop a Virtual Measurement Tool to survey the liquid holdup and flow regimes in multiphase flow in pipelines and wellbores. The method confirmed to be a precise virtual measuring means to predict liquid holdup and flow regimes in these systems [Ternyik *et al.*, 1995(b)].

2.4.2 Unified Mechanistic Models

As the name indicates, unified mechanistic models are claimed to be applicable for all ranges of angles of inclination, [Gomez *et al.*, 1999]. The first attempt had a presentation of a unified flow patterns model that was valid for all inclination angles, [Barnea, 1987]. The model encompassed all the relations that had been proposed in earlier publications. The author had conducted extensive comparisons between the theoretical and experimental maps which were resulted in good agreements for a wide tested range of angles of inclination and internal diameters. Another unified mechanistic model had been proposed for steady-state two-phase flow in wellbores and pipelines, [Gomez *et al.*, 1999].

Their model incorporated a unified flow pattern prediction and unified individual models for stratified, slug, bubble, annular and dispersed bubble flow, applicable to the entire range of inclination angles. Gomez *et al.* (1999) had claimed that their model can be applicable to vertical wellbores, horizontal wells, directional wells, and pipelines, under normal production operation or artificial lift.

Gomez *et al.* presented their comprehensive model for prediction of flow pattern, liquid holdup and pressure drop in wellbores and pipelines. The authors made their model valid for inclination angles ranged from horizontal to upward vertical flow. The model had been validated using laboratory and field data. Furthermore, the model had been tested against field data, from the North Sea and Prudhoe Bay, Alaska.

The model's pressure drop performance also had been compared to other six models and showed outstanding results. The overall performance of the unified model showed an average error of -3.8% and an absolute average error of 12.6%. Additionally, the performance of the proposed unified model was evaluated against eighty six directional well field data cases. The predictions of the unified model show an excellent agreement with data, with an average error of -1.3% and an absolute average error of 5.5%, [Gomez *et al.*, 1999].

Other unified models have been published by several authors, which are applicable for certain flow regimes such as the one developed by Felizola and Shoham (1995) for prediction of slug flow in upward inclined pipes, [Felizola and Shoham, 1995] and another one developed by Zhang *et al.* (2003b) for estimation of gas-liquid pipe flow via slug dynamics, [Zhang, Q. *et al.*, 2003(b)].

The above literature survey reveals that separate comprehensive mechanistic models are available for pipeline flow and wellbore flow. Currently, each of these mechanistic models has an outstanding performance in a specific flow pattern prediction and that makes the adoption for certain model of specific flow pattern by investigators to compare and yield different, advanced and capable mechanistic models.

Tackacs (2001) stated that many researchers agreed upon the fact that no single correlation was found to be applicable over all ranges of variables with suitable

accuracy, [Tackacs, 2001]. It was found that correlations are basically statistically derived, global expressions with limited physical considerations, and thus they do not render to a true physical optimization.

A statistical study on the possible source errors in empirical correlation and mechanistic models has been conducted for comparative purposes, [Tackacs, 2001]. He concluded that there is no pronounced advantage for mechanistic models over the current empirical correlations in pressure prediction ability when fallacious values are excluded. Actually, there is no privilege for mechanistic models over the existing empirical correlation but they behave similarly when mistaken data from the former is taken out.

2.5 State of the Art

Currently, the state of the art in multiphase flow in pipes and wellbore is the merging of both two fluid flow transient simulators and steady state mechanistic models, [Brill and Arirachakaran, 1992]. These two fluid flow transient simulators are able to analyze complex time-dependent problems but they often suffer convergence problems. However, the steady state simulators were shown to be inadequate in designing downstream facilities in which flow rate is fast changing. The use of these steady state mechanistic models allows to suitably forecasting the pressure drop, flow variables under many conditions. Additionally, two approaches have been described in the literature for designing pipe models; those are steady state and transient pipe models.

The steady state pipe module permits solving the steady state flow along the pipe by taking into account latent behavior of all phases. The fluid description is completely compositional. The transient pipe models have been developed at IFP (French Institute of Petroleum) since the early 1990's [Pauchon *et al.*, 1993], [Faille and Heintze, 1996], and [Masella *et al.*, 1998].

Some of these simulator models were designed based on steady state conditions (PEPITE, WELLSIM and TUFFP), while others were fabricated under transient

steady state conditions (OLGA and TACITE). A brief historical background about these simulation models are presented below.

PEPITE has been created by The French Institute of Petroleum (IFP) and the two French oil companies, ELF and TOTAL. The initial design of PEPITE has been conceived in 1974. A final model was developed in the in 1980 after a series of continuous improvement, [Lagiere, 1984], and [Roux *et al.*, 1988]. This cannot be applied to vertical and highly inclined pipes and wellbores.

Additional effort had been done by the same French consortium to overcome these shortcomings. WELLSIM came to existence in 1985 and it has been evaluated by two authors, [Ozon *et al.*, 1987], and [Corteville *et al.*, 1991].

A consortium of some oil companies and Tulsa University Fluid Flow Projects developed a steady state simulator, which was later known as TUFFP. Likewise PEPITE model, the developed model was appropriate to be used for horizontal and/ or near horizontal pipelines. The simulator had been tested for its ability to detect the flow pattern regimes. Additional information and evaluation of this model were presented in literature (refer to Section 2.4.1.1), [Xiao *et al.*, 1990].

Norwegian Institutes SINTEF and FE developed a simulator called OLGA that designed for flow of oil, water and gas in wells and pipelines. Theory and application of this simulator were presented in many publications.

Bendiksen *et al.* (1991) published a paper about the theory and application of OLGA. The authors reported that OLGA was originally based on small diameter data for low-pressure air/water flow. Investigations have shown that the used data were capable of describing the bubble/slug flow regime, while the stratified/annular regime was not. In vertical annular flow, the predicted pressure drops were up to 50% too high. In horizontal flow, the predicted holdups were too high by a factor of two in extreme cases.

The OLGA model has been tested against experimental data over a substantial range in geometrical scale (diameters from 2.5 to 20 cm, some at 76 cm; pipeline

length/diameter ratios up to 5,000; and pipe inclinations of -15 to +90°), pressures from 100 kPa to 10 MPa, and a variety of different fluids. The authors confirmed that the model gave reasonable results compared with transient data in most cases. The model was also tested on a number of different oil and gas field lines. In general, the OLGA predictions are in good agreement with the measurements, [Bendiksen *et al.*, 1991].

The TACITE hydrodynamic module, which was developed in the mid 80s by IFP, was the main model [Pauchon *et al.*, 1993]. The Drift Flux Model and the No Pressure Waves Model had been used to develop the TACITE transient pipe modules. The module was more sophisticated than PEPITE and WELLSIM since it incorporated both pipeline and wellbore configuration with all possible angles of inclination, [Pauchon *et al.*, 1993].

A latest study by Dhulesia and Lopez (1996) showed that the TACITE model performed better than the OLGA model and several tested models for pressure drop estimation for various real pipelines and wells. The study confirmed that TACITE is a reliable tool for predicting pressure drop, [Dhulesia and Lopez, 1996].

2.6 Artificial Intelligence

The science of artificial intelligence or what is synonymously known as soft computing shows better performance over the conventional solutions the aim of artificial intelligence is defined as development of paradigms and algorithms that require machines to perform tasks apparently require cognition when performed by human, [Sage, 1990]. This definition is widely broadened to include preceptrons, language, and problems solving as well as conscious, unconscious processes, [Memmi, 1989]. Many techniques are classified under the name of artificial intelligence such as genetic algorithms, expert systems, and fuzzy logic because of their ability, one at least, to make certain reasoning, representation, problem solving, optimization capabilities and generalization. Artificial neural network can also be considered as one of the important components of artificial intelligence system. The concept of artificial neural network is presented in Appendix A.

2.6.1 Artificial Neural Network

2.6.1.1 Historical Background

The research has been carried on neural network can be dated back to early 1940s. Specifically in 1943, when the low-level structure of biological brain system has been modeled, [McCulloch and Pitts, 1943]. A book entitled “*the organization of behavior*” has been published in which the main focusing was towards an explicit statement of a physiological learning rule for synaptic modification, [Hebb, 1949]. The author proposed that the connectivity of the brain is continually changing as an organism learns differing functional tasks, and the neural assemblies are created by such changes. The book was a source of inspiration for the development of computational models of learning and adaptive systems.

Additionally, another book entitled “*design for a brain; the origin of adaptive behavior*”, has been also published to shed more light on the adaptive systems and their relationship with brain design, [Ashby, 1952]. The book focused on the basic notion that the adaptive behavior is not inborn but rather learned. The book emphasized the dynamic aspects of living organism as a machine and the related concepts of stability. In the same way, the idea of nonlinear adaptive filters has been adapted to give additional information about filter learning, [Gabor, 1954]. The author mentioned that learning was accomplished in these filters through feeding samples of stochastic process into the machine, together with the target function that the machine was expected to produce. After 15 years of the initiative McCulloch’s paper, [McCulloch and Pitts, 1943], a new approach to the pattern recognition problem was introduced through what’s called later, *perceptrons and associated learning rules*, [Rosenblatt, 1958]. The latter, at the time when discovered, considered as an ideal achievement and the associative theorem “*perceptron convergence theorem*” was approved by several authors, [Minsky and Papert, 1969; Rosenblatt, 1962], [Minsky and Papert, 1988], and [Novikoff, 1963]. The perceptron is the simplest form of a neural network that has been used for classifying patterns.

This achievement followed by the introduction of *LMS* “least mean square algorithm” and *Adaline* “adaptive linear element”, [Widrow, B. and Hoff, 1960]. The difference between the perceptron and Adaline lies in the training procedure, [Haykin, 1999]. That effort had been followed by *Madaline* “multiple-Adaline” in 1962, [Widrow, B., 1962]. This can be considered as the earliest trainable layered neural networks with multiple adaptive elements. A figured study showed that there are several problems cannot be solved by the theorem approved by Rosenblatt, [Rosenblatt, 1958] and therefore countless effort to make such type of improvement will result in nothing, [Minsky and Papert, 1969]. A decade of dormancy in neural network research was witnessed because of that paper’s results.

In 1970s, a competition learning algorithm was invented along with incorporation of self organizing maps. Since that time, several networks and learning algorithms were developed. A discovery of back-propagation learning algorithm was one of these fruitful revolutions, which had been conceived in early 1980s, [Rumelhart *et al.*, 1986].

2.6.1.2 Definition

Generally, ANN is a machine that is designed to model the way in which the brain performs a particular task or function of interest. The system of ANN has received different definitions by statisticians and mathematicians, [Haykin, 1994]. A widely accepted term is that adopted by Aleksander and Morton “*A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use*”, [Aleksander and Morton, 1990]. ANN resembles the brain in two aspects; knowledge is acquired by the network through a learning process, and the interneuron connection strengths known as synaptic weights are used to store the knowledge, [Haykin, 1994]. Moreover, neural networks are simply a way of mapping a set of input variables to a set of output variables through a typical learning process. So, it has certain features in common with biological nervous system. The relationship between the two systems and the brain system mechanism is further explained below.

2.6.1.3 *Brain system*

Human brain is a highly complex, nonlinear, and parallel information-processing system. It has the capability of organizing biological neurons in a fashion to perform certain tasks. In terms of speed, neurons are five to six orders of magnitude slower than silicon logic gates. However, human brain compensates for this shortcoming by having a massive interconnection between neurons.

It is estimated that human brain consists of 10 billion neurons and 60 trillion synapses, [Shepherd and Koch, 1990]. These neurons and synapses are expected to grow and increase in both number and connection over time through learning.

Fig 2.1 is a schematic representation of a biological nerve cell. The biological neuron is mainly composed of three parts; dendrite, the soma, and the axon. A typical neuron collects signals from others through a host of fine structure (dendrite). The soma integrates its received input (over time and space) and thereafter activates an output depending on the total input.

The neuron sends out spikes of electrical activity through a long, thin strand known as an axon, which splits into thousands of branches (tree structure). At the end of each branch, a synapse converts the activity from the axon into electrical effects that inhibit or excite activity in the connected neurons. Learning occurs by changing the effectiveness of synapses so that the influence of one neuron on another changes. Hence, an artificial neuron network, more or less, is an information processing system that can be considered as a rough approximation of the above mentioned biological nerve system.

Fig 2.2 shows a typical neuron in an artificial neuron network. This mathematical neuron is much simpler than the biological one; the integrated information received through input neurons takes place only over space.

Output from other neurons is multiplied by the corresponding weight of the connection and enters the neuron as an input; therefore, an artificial neuron has many inputs and only one output. All signals in a neural network are typically normalized to

operate within certain limit. A neuron can have a threshold level that must be exceeded before any signal is passed.

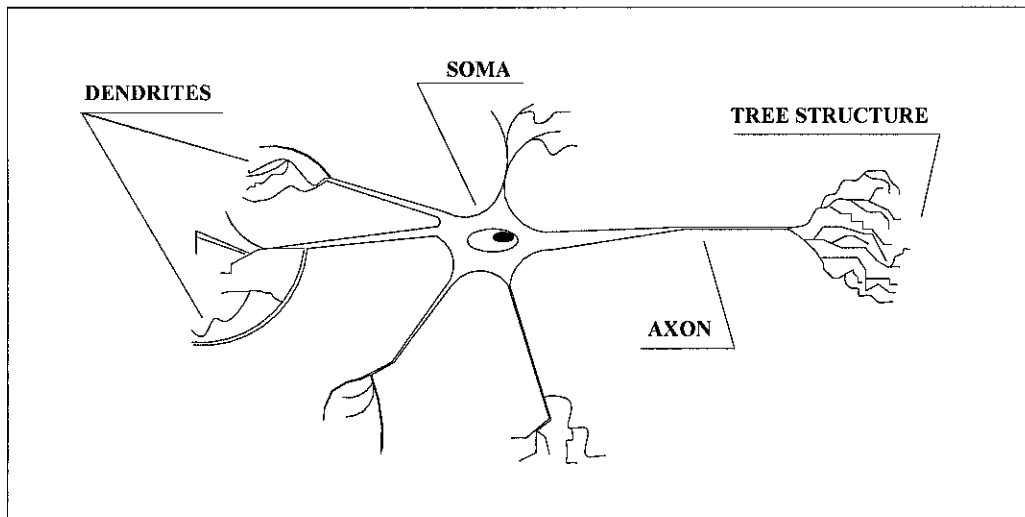


Fig 2.1: Major Structure of Biologic Nerve Cell, reprinted with permission [James and David, 1991]

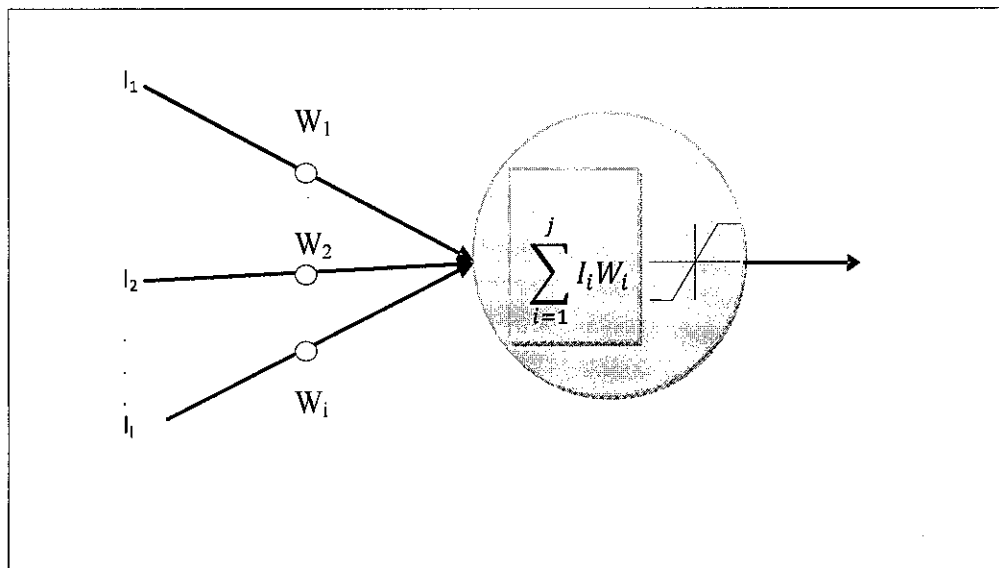


Fig 2.2: Artificial Neuron, reprinted with permission [James and David, 1991]

The net input of the activation function may be increased by employing a bias term rather than a threshold; the bias is the negative of threshold. The inputs are summed and applied to the activation function and finally the output is produced, [Gabor, 1954].

2.6.2 The Use of Artificial Neural Networks in Petroleum Industry

The use of artificial intelligence in petroleum industry can be tracked back to the beginning of 1990's [Mohaghegh and Ameri, 1995]. The use Artificial Neural Network (ANN) in solving many petroleum industry problems was reported in the literature by several authors such as Mohaghegh *et al.* (1995), Bilgesu, H. I. *et al.*, (1998) and Oyenevin and Faga, (1999), [[Mohaghegh and Ameri, 1995]; [Bilgesu, H. I. *et al.*, 1998], and [Oyenevin and Faga, 1999]].

Conventional computing tools have been used to estimate a relationship between permeability and porosity. However, the obtained accuracy was weak. Knowing the behavior of this relationship is of utmost significance for estimating the spatial distribution of permeability in the reservoirs especially those of heterogeneous lithofacies. ANNs was used successfully in determining the relationship between these facies and constructing excellent estimation, [Mohaghegh *et al.*, 1995]. For instance; ANN has a great share in solving problems related to drilling engineering such as drill bit diagnosis and analysis [Bilgesu, H. I. *et al.*, 1998] and [Oyenevin and Faga, 1999]. Moreover, ANNs has been used efficiently to optimize production, and fracture fluid properties, [Holditch *et al.*, 1993].

2.6.3 Artificial Neural Networks in Multiphase Flow

Recently, ANN has been applied in the multiphase flow area and achieved promising results compared to the conventional methods (statistical regression methods, empirical correlations, and mechanistic models). With regard to this field, a few researchers applied ANNs technique to resolve some problems associated with multiphase problems including flow patterns identification, liquid hold up, and estimation of gas and liquid superficial velocities.

Arirachakaran *et al.* (1991) proposed an intelligent program, supported by a knowledge data base and human interaction to interpret the results obtained from prediction of flow pattern by mechanistic models. An expert systems approach that displays some sort of intelligence is capable of thinking like humans and have a learning talent was suggested by the author as a pioneering step of ANN. This expert

system flow pattern simulator, the author suggests, can be intelligently utilized as a computer aided engineering tool in production system optimization, [Arirachakaran *et al.*, 1991].

A solution was presented for predicting flowing bottom-hole pressure in multiphase flow, both for wellbores and pipelines. Ternyik *et al.* (1995a) formulated separate neural networks for each case by using back-propagation method along with different set up and inclination angles, [Ternyik *et al.*, 1995(a)].

Their new approach, which was called virtual measurement in pipes (VMP), was designed to address the development of tools to predict pressure drops in pipes. It outperforms the conventional method (five empirical correlations were used to compare results) in its generality and prediction capability. His approach worked reasonably with lower standard deviation and mean values when used for oil wells. The small number of data sets and high number of variables used in his study in hidden layer, which might limit their model generality. Also, they proceeded with the application of VMP in prediction of liquid holdup and flow regimes in pipes and wells.

ANN utility of differentiating complex pattern has proved to be a good tool in this area especially where complex relationship between flow patterns present. The model can fit correctly at any inclination angle and might be claimed as a unified model for flow patterns and liquid hold up prediction. experimental data were extracted from Mukherjee thesis, [Mukherjee, 1979] due to wide reported coverage of inclination angles to provide more generality and confidence to the output results.

A Kohonen type network was utilized due to the ability of this network to self learning without depending on the output in each case. His model was restricted to a 1.5 inch tubing diameter and low operating condition, which limit the generality of his model.

The need for accurate hold up and flow patterns prediction stimulated another researcher; [Osman, S. A., 2001] to propose an artificial neural networks model for accurate prediction of these two variables under different conditions. One hundreds and ninty nine data points were used to construct his model. Neural Network

performed perfectly in predicting liquid hold up in terms of lowest standard deviation and average absolute percent error when compared to published models. His model did not work efficiently in the transition phases.

An artificial neural networks model was presented for predicting pressure drop in horizontal and near-horizontal multiphase flow, [Osman, S. and Aggour, 2002]. A three-layer back-propagation ANN model was developed using a wide range of data. Thirteen variables were considered as the most effective variables incorporated in pressure drop prediction. Their model achieved outstanding performance when compared to some of the existing correlations and two mechanistic models. The model was also found to correctly simulate the physical process.

2.7 Abductive Induction Mechanism (AIM)

This part will give historical background about the AIM and when it was conceived and types of updates occurred. Detailed description about fundamentals of the algorithm is given in Appendix A.

2.7.1 Short History

The GMDH-based abductive networks algorithm was built up by Professor Alexey G. Ivakhnenko in the year 1968 at the Institute of Cybernetics in Kiev (Ukraine). The major purpose of its introduction was the recognition of relationships in large complex non-linear multi-dimensional systems, their approximation, and prediction. To reach its current status, the GMDH-based abductive network algorithm has passed several rejuvenations and modifications by several researchers. However, Japanese and Polish scientists had contributed significantly to the update of the algorithm, [Sawaragi *et al.*, 1979].

They concluded that “GMDH is the best method for solving the AI problems - identification, short-term and long-term forecast of random processes and pattern recognition in complex systems”. Mathematical GMDH theory showed that

regression analysis can be described as the particular case of GMDH, [Ivakhnenko and Yurachkovsky, 1986].

Most of the updated GMDH theory has been reported in Ukrainian journal "Automatica"). It is clearly shown that the journal subdivided the progress in GMDH theory into five sub eras. Major contributions will be reported as follows:

Period 1968-1971: This period is distinguished by application of one regularity criterion for solving of the problems of clustering, pattern detection and short-term forecasting. As reference functions polynomials, logical nets and Bayes probability formulas were used. However, noise-immunity was not investigated in this period.

Period 1972-1975: This period is featured by solving the problem of modeling of noised data and with incomplete information basis. Multi-criteria selection and utilization of additional priory information for noise-immunity increasing were proposed.

Period 1976-1979: This period is marked by the investigation of the convergence of multilayered GMDH algorithms. It was shown that some multilayered algorithms have "multilayeredness error". The solution of objective systems analysis problems by multilayered GMDH algorithms was proposed.

Period 1980-1988: Many important theoretical results were received. It became evident that full physical models are not suitable to be used for long-term forecasting. It was confirmed, that non-physical models of GMDH are more accurate for approximation and forecast than physical models of regression analysis.

Period 1989-to present time: This period is characterized by the development of new algorithms for non-parametric modeling of fuzzy objects and Simplified Learning Programming algorithm for expert systems.

The current progress is devoted to development of twice-multilayered neuronets and parallel combinatorial algorithms for multiprocessor computers.

2.7.2 Advantages and Disadvantages of AIM

Compared to neural networks, the AIM approach has quite a lot of advantages, [Madala and Ivakhnenko, 1994].

1. It does not require time-consuming training and gives an explicit model of the system.
2. The optimal number of layers and neurons is determined automatically without any user interference. The polynomial GMDH network learns the weights very fast through the standard fitting procedure, which produces locally guaranteed model.
3. The weights will be constant for a certain set of data while in ANN the weights are initiated randomly.
4. The final model is easy to be interpreted through a set of simple equations while in ANNs still the interpretation of the final model is questionable.

Disadvantages of AIM can be stated as follows:

1. A tendency to construct a quite complex polynomial for reasonably simple systems, unless certain stoppage criterion is applied.
2. A propensity to producing overly complex model when dealing with highly nonlinear systems.
3. Compromising between model's generality and complexity is hard to obtain

2.8 The Use of Abductive Networks in Geosciences and Petroleum Industry

Abductive networks have many applications in engineering as well as in oil & gas industry. Extensive search in the petroleum engineering literature for the application of abductive network resulted in many published papers in drilling optimization [Lee *et al.*, 1995], reservoir properties [Osman, E.A. and Abdel-Aal, 2002], as well as in multiphase flow area [Park and Kang, 2006]. A research had been done to apply Abductive Network for predicting tool life in drilling operations. Optimal network structure was constructed automatically to include drill diameter, cutting speed and feed-rate as effective parameters for predicting tool life. The network was able to

predict drill life under varying cutting conditions with the estimation error of drill life is less than 10%, [Lee *et al.*, 1995]. Additional work by the same authors was devoted to model and optimize drilling process. The same input parameters were used to predict tool life, metal removal rate, thrust force and torque [Lee *et al.*, 1998]. A global optimization algorithm was utilized for these networks in order to search for optimal drilling process parameters subjected to an adjustable objective function and inequality constraints. The results were confirmed by applying several drilling tests with the optimal process parameters.

A collaborative work between two of King Fahd University of Petroleum and Minerals researchers has resulted in an invaluable review paper that discusses the futuristic and potential applications of Abductive networks in petroleum engineering and opens the door for a new era of intelligent modeling [Osman, E.A. and Abdel-Aal, 2002]. The authors concluded their effort by building two models for estimating bubble point pressure and formation volume factor. Their models outperformed the rest of empirical correlations in terms of lowest errors and highest correlation coefficients.

With regard to the application of Abductive networks in multiphase studies, recent study has been conducted to model liquid holdup in horizontal two phase flow. A polynomial neural network (PNN) model for estimating liquid holdup in horizontal two-phase flow is proposed by two researchers, [Park and Kang, 2006]. The PNN utilizes the concept of Abductive networks with the most active neurons being involved in generating the successful model. Data have been collected from literature.

The developed model has been compared against two empirical correlations (Abdul-Majeed (1996) and Minami and Brill (1987)) and two Backpropagation ANN models (Gradient decent algorithm and Polak-Ribiere conjugate gradient). The authors noticed that most of the investigated models produced reasonable accuracy at low range of liquid hold up value while lost their accuracy when liquid hold up values increased. The polynomial neural network model outperforms previous models in overall accuracy across liquid holdup ranges. Three statistical parameters were chosen to compare models performances; these were the average root mean square error

(ARMS); the average percent error (APE); and the average absolute percent error (AAPE). Summary of these statistical comparisons are shown in Table 2-3.

Table 2-3: Statistical Comparison of all Investigated Models, reprinted with permission (Park and Kang 2006)

Statistical Criterion	ARMS	APE	AAPE
Prediction method			
Abdul-Majeed (1996)	0.1759	3.12	29.31
Minami and Brill (1987)	0.0828	11.78	26.17
Backpropagation neural network			
Gradient decent algorithm	0.1139	-4.89	22.94
Polak-Ribiere conjugate gradient	0.042	-2.87	14.79
Polynomial neural network	0.0372	-1.79	11.57

Determination of an accurate permeability model from well logs, especially from heterogeneous formation, is a formidable task. To overcome such a problematic issue, a new hybrid intelligent approach consisted of combining polynomial neural networks and genetic algorithm had been proposed. Data were collected from eight well logs within offshore field based in Korea. The new approach had a profound superiority over the conventional artificial neural networks. For a comparative study, both conventional ANN and advanced PNN were applied to the well log data. The computed results from both models were compared with core measured permeability. Two statistical features were selected by the authors as the evaluation criteria; those are coefficient of determination and average root mean square error. With regard to coefficient of determination, the closer this value to 100% the better the model is.

The polynomial neural network model managed to achieve a value of 98.7%, while the conventional neural network model obtained poor result of 55.4%. With regard to second statistical evaluation feature (average root mean square error), the smallest value indicates the betterness of the model. The polynomial neural network model achieved a value of 2.7, while the conventional neural network model obtained poor result of 5.5, [Lim *et al.*, 2006].

A recently published study had been devoted to estimate some of reservoir properties from seismic attributes using Abductive networks, [Ahmed *et al.*, 2010]. The authors applied Abductive networks to estimate porosity from seismic data of an area within the 'Uthmaniyah' portion of the Ghawar oil field, Saudi Arabia. Their data contained the following parameters; normal seismic amplitude, acoustic impedance, sixteen other seismic attributes, and porosity logs from seven wells located in the study area. The abductive network managed to select out the best two to six attributes of twenty seven attributes. The newly developed model outperformed common neural-network predictors for porosity estimation. The authors claimed that their model provided adequate predictions in spite of the limited well data available. This can be interpreted in achieving a mean absolute prediction error of 0.038 and correlation coefficient of 91.1% for the five evaluation wells. On the hand, the traditional ANN approaches such as regularized neural networks produced some negative porosity values, which indicate the unsuitability of such approaches for prediction of this feature.

Following the same approach, a similar study had been conducted to show the superiority of GMDH technique in estimating reservoir properties from well logs data, [Semenov *et al.*, 2010]. An example was presented where porosity model of Vankor field (Dolgan formation) had been generated from different logs using linear regression, neural networks, and GMDH approach. The latter had shown the best prediction capability compared to the other two investigated methods where the best input logs had been selected automatically. In terms of statistical comparison, the GMDH model was able to get higher core data correlation coefficient of 38% (resistivity, neutron, and density logs used) while the conventional neural network approach obtained 27% (spontaneous potential, neutron, and density logs used) and the traditional linear regression model achieved 24% (spontaneous potential log used).

As stated by different authors and researchers, and as discussed earlier, the empirical correlations and mechanistic models failed to provide a satisfactorily and a reliable tool for estimating pressure in pipeline systems under multiphase flow conditions.

High errors are usually associated with these mechanistic models and empirical correlations, which has encouraged new approaches to be investigated for solving this problem. An interesting study showed that when empirical correlations are compared with mechanistic models for their performance in predicting pressure drop for the following models, [Ansari *et al.*, 1994] and [Hasan, A. and Kabir, 1988], no privilege for the mechanistic models over the empirical correlations were found in estimating pressure drop, [Pucknell *et al.*, 1993]. This finding strengthens the fact that both empirical and mechanistic models are designed for certain conditions, and consequently applicable to special set of conditions. So, this study proposes a new means for estimating the pressure drop at a wide range of angles of inclination.

Two models will be generated using the latest computing techniques. Additionally, the outcome of this research is an attempt to shift the industry's attention towards the potential of using these latest computing techniques in this highly complicated area. The utmost goal is to overcome the accuracy problem that encountered on those old conventional methods (empirical correlations and mechanistic models).

2.9 Summary

This chapter presented the author's search in the previous work related to the areas of multiphase flow, Artificial Neural Networks, and Abductory Inductive Mechanism. The Chapter constituted comprehensive coverage of the conventional means used for prediction of pressure drop. It contained critical evaluation and discussion of other related researches. However, great emphasis has been devoted to models that are designed originally for pressure drop estimation covering wide range of angles of inclination. Additionally, basic concepts and fundamentals of ANN and AIM techniques have been presented along with their applications in Petroleum Engineering and Multiphase Flow areas.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Overview

The research methodology involves filling the gap existing in the literature by assessing and evaluating the best MPF (multiphase flow) empirical correlations and mechanistic models. The assessment will deal with their performance in estimating pressure drop whilst using available statistical and graphical techniques. The performance of the developed models will be compared against the best available correlations used by the industry. The following schematic diagram (Fig 3.1) illustrates the sequence of research events. ANN and AIM techniques will be utilized in this study in an attempt to overcome the degraded accuracy of old models used for prediction of pressure drop. ANN technique is well-known for its ability to discover highly complicated relationships between different data sets and the targeted output. While AIM technique will be utilized to discover the most important input attributes that are having the great effect and contribution in pressure drop estimation.

It is clearly evident that data collection is the first step in generating a successful modelling study. Data collection consisted of gathering the relevant information pertinent to the course of study. The attributes should be well known to be contributing to the desired output. Irrelevant information can mislead the desired target. In addition to that the collected data must answer a simple question. Are the quality and quantity of collected data able to provide an improvement for the solution of current problem? Without collecting useful data nobody can tell if the generated model will simply succeed in providing an answer for the questions posed.

In the problem of estimating pressure drop in pipelines with a wide range of angles of inclination, so many parameters are known to be contributing in the estimation of pressure drop such as; Temperature at Standard Conditions ($^{\circ}\text{F}$), Surface Temperature ($^{\circ}\text{F}$), Separator Temperature ($^{\circ}\text{F}$), Pressure at Standard Conditions (psi), Surface (psig), Separator Pressure (psig), Tubing Inner Diameter (inches), Pipe Roughness Value (inches), Angle from Horizontal (degrees), Length of Pipe (ft), Specific Gravity of Gas, Oil Gravity, Specific Gravity of Water, Oil Flow Rate (Stb/day), Water Flow Rate (Stb/day), Gas Liquid Ratio (Scf/stb), Oil Density (API) and Viscosity (cp). However, not all these parameters might be significantly contributed to the final output. Interestingly, some of these parameters cannot be available in the collected data due to some technical problems such as missing of the assigned reports that contain the said attributes. Such missing data attributes include Pipe Roughness Value (inches), Specific Gravity of Gas, Oil Gravity, Specific Gravity of Water, Oil Density (API) and Viscosity (cp). Although this insufficiency in the data can reduce the information fed to the model, on the other hand, it might not significantly affect the precision of modeling procedure. Additionally, some of these input parameters were removed from the final data selection due to their low ranges.

A total number of 335 data sets had been utilized during the course of this study for modeling purposes (range of collected data had been presented in Appendix D). Relevant input variables were selected based on the most commonly used empirical correlations and mechanistic models used by the industry. Eight attributes were thought to have a strong impact on the pressure drop estimation, which are; oil rate, water rate, gas rate, diameter of the pipe, length of pipe, wellhead pressure, wellhead temperature, and angle of deviation. An automated system used to collect all these data variables is called SCADA. A short discussion about its function will be provided in the next subsection.

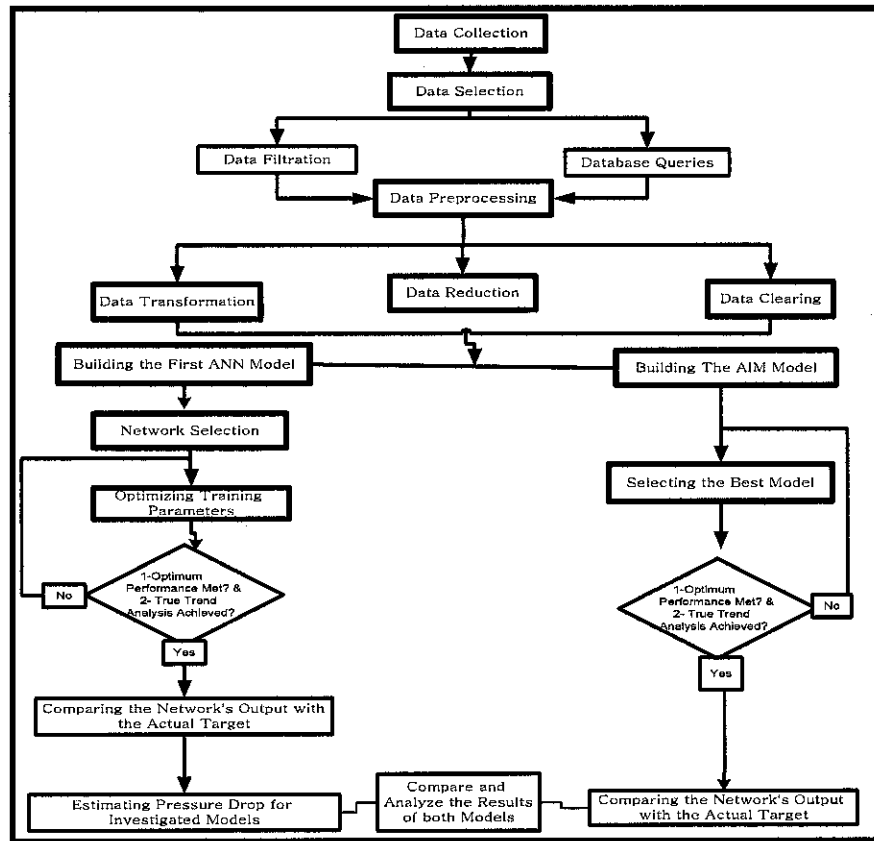


Fig 3.1: Methodology Chart for Models Generation

3.1.1 SCADA System

SCADA (supervisory control and data acquisition) system is a stand-alone system used for collecting large amount of raw data that needs continuous analysis to convert it into meaningful action. It is used for controlling the pipeline operation within a pre-set of parameters. A simplified SCADA system is depicted in Fig 3.2. The data is being collected based on real-time manner and the continuous variables have been stored in database management system for further analysis. The SCADA system consists of the following hardware:

1. The main (host) computer
2. Communication equipment. This varies between local and remote equipment
3. Operator interface video display units. This is the visible part of SCADA seen by the operator

4. Remote terminal units (RTU), these are responsible of collecting process variable data and storing it for further interpretations
5. Process interface units. This can be categorized as digital inputs, analog inputs (for continuous functions such as temperature and pressure measurements), accumulator inputs, and digital outputs (such as flow controller).

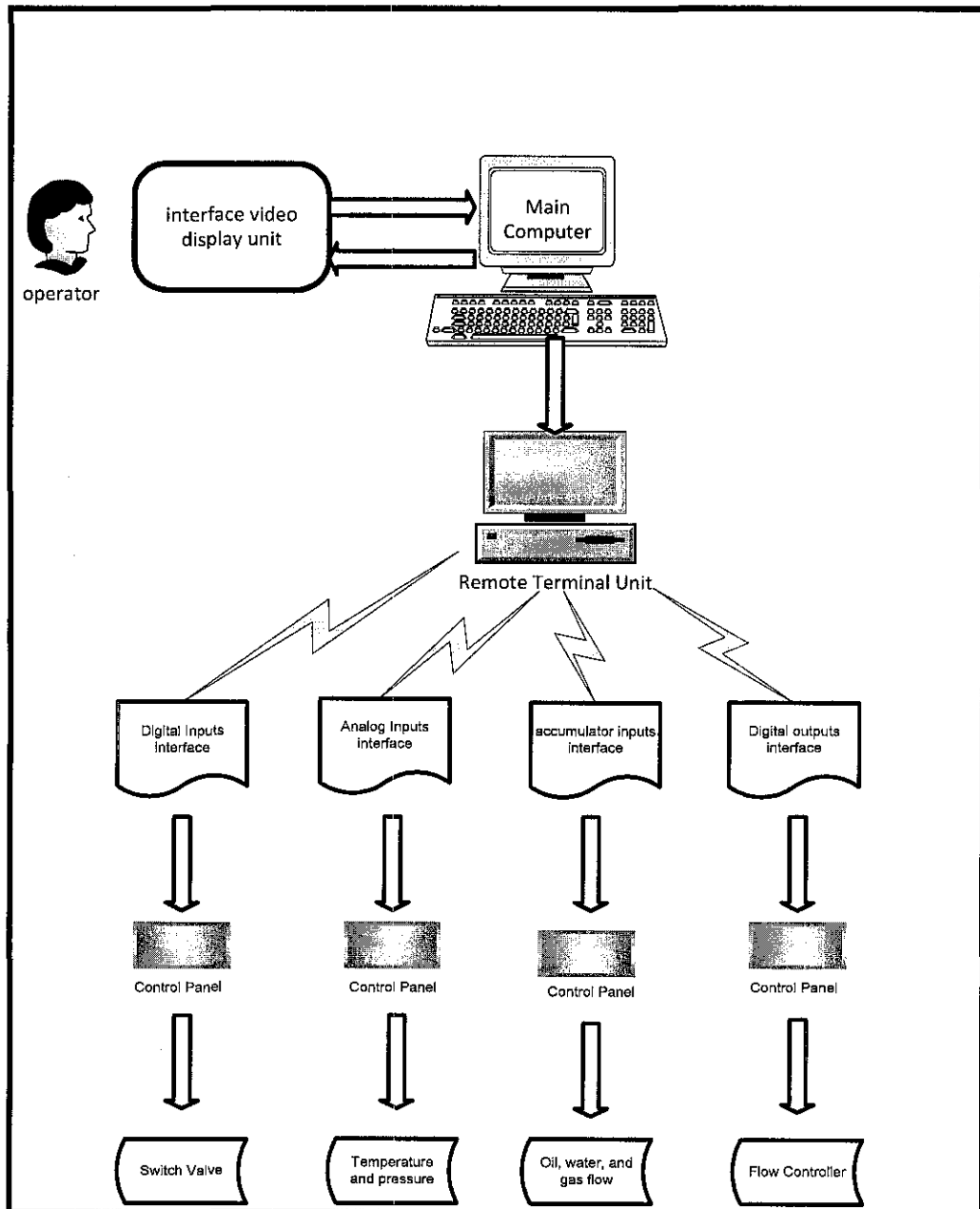


Fig 3.2: A simplified SCADA System

3.2 Data Selection

In this part two main processes are normally carried out in order to check for the optimum data selection, which are, database queries and data filtration.

3.2.1 Database Queries

Database queries can be considered as a pivotal step in retrieving and selecting the required data for a specific task. Usually data are stored in a tabulated form with so much irrelevant information kept inside with mismatched ones. The term ‘query’ means to find, search and to question. The purpose of data enquiry is to get back some relevant information from the main data bank (database) by questioning and searching the database. This can be done through applying certain code for retrieving this part of the data. However, so many high-level programming languages do exist for performing such a task such as SQL (Structured Query Language). In this study, minor database queries have been performed on the current data.

3.2.2 Data Filtration

Data filtration includes, but not limited to, removing data outliers. It also includes finding non-normal distributions and other anomalies within the data. The reason for performing this step is that the collected data and measurements are usually affected by noise. Additionally, data points are improperly recorded and saved, and because of current device malfunctions.

Outliers are those points which depart from what they are expected to be. Also, “*it is an observation which appears to be inconsistent with the remainder of that set of data*”, [Barnett, 1978]. This depends on the error bounds. So many statistical methods are used to detect the outliers. Linear regression and other regression methods can be used to track outliers where the confidence interval (bound) is used to shape data points.

Fig 3.3 shows the use of linear regression for outlier detection.

The solid line indicates the best fit by the linear regression model while the dotted lines showed the selected confidence interval, [Tamraparni and Johnson, 2003]. As it appears from the same graph, the black point might be suspected as an outlier point. In this study outliers (anomalies) have been detected using semi-studentised technique or alternatively known as standard residual, this will be demonstrated in the next chapter.

Missing values in the recorded data are treated using several ways. The missing values and their treatments are considered very rigorous and neat step in data quality assurance. Guessing the missing value in data gathering has a defined acronym called 'imputing missing value'. In some cases, where meaningful records do exist, the missing values are replaced by representative values that are collected from the mean or median value. Each record is treated separately by simulating the normal trend and then imputing the missing value. Normal regression and interpolation can aid in guessing the missing values. However, defining the optimum methodology or treatment for the missing values can only depend on the type of the data at hand and quality and precision of the subsequent analysis.

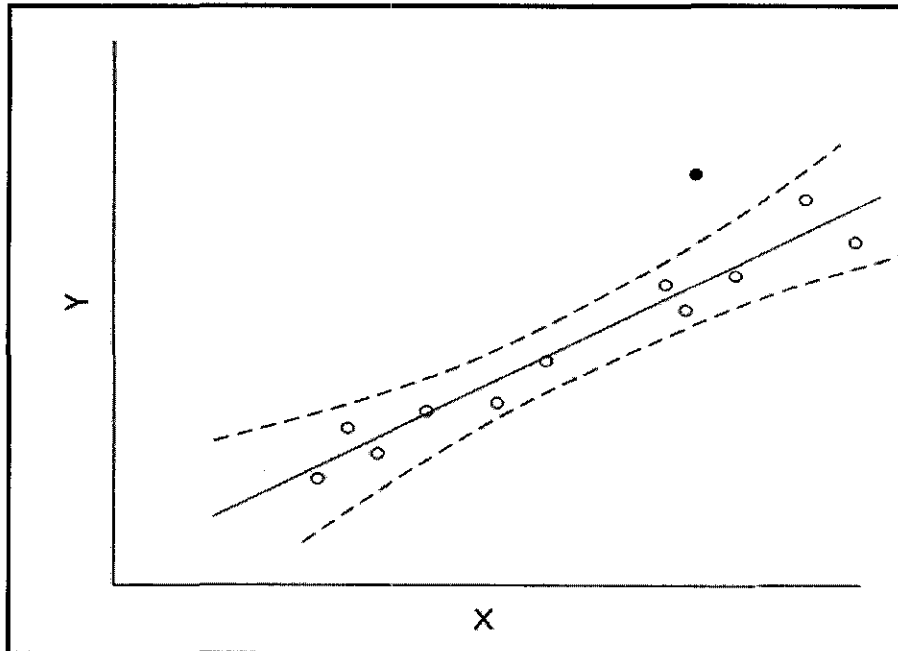


Fig 3.3: Finding outliers using linear regression, reprinted with permission [Tamraparni and Johnson, 2003].

3.3 Data Preprocessing

In this part three different data preprocessing techniques will be discussed thoroughly, which are data clearing, data reduction, and data transformation.

3.3.1 Data Clearing

The term data clearing or “cleaning” includes two operations; filling in the missing value and identifying the outliers and smoothing out noisy data “if do exist”. The first step is to fill in the missing value by any well known techniques such as those defined in Section 3.2.2. The second step is to identify the outliers and smooth out noisy data. This step can be done using several approaches as described by many statistical books such as binning, clustering and regression techniques.

3.3.2 Data Reduction

Data reduction involves different techniques where data are being reduced even in number of values and/or number of attributes. However, number of attributes has been reduced by removing irrelevant attributes throughout the entire data. Attributes that have shown minimum effect on the target output have been selected out to be irrelevantly contributing to the final output. The latter, was verified by most relevant input parameters that were used extensively in deriving correlations and mechanistic models.

3.3.3 Data Transformation

Data transformation was used for ANN model where all data samples had been transformed or scaled to fall within a pre-specified range. This step was crucial before generating a successful ANN model because it eliminated the harmful effect of varied input ranges. This step was needed to transform the data into a suitable form to the network inputs and targets. The approach used for scaling network inputs and targets was to normalize the training set through using mapminmax function (built-in

function in MATLAB) within a pre-specified range [-1, 1]. The function can be mathematically expressed by;

$$y = \frac{(y_{\max} - y_{\min})(x - x_{\min})}{(x_{\max} - x_{\min})} + y_{\min} \quad (3.1)$$

In order to transform the x value to y value, the above formula had to be implemented providing that the range of the data fall between y_{\min} & y_{\max} , which was selected to be between -1 and 1.

By performing this step it became easy for the network to cope with high and small range of data columns (sometimes known as scaling). This can ensure equalizing the importance of each input variable. This step was quite important because it guaranteed that the size of the parameter did not reflect its importance to the output, hence, no single variable will be dominant against the other variables.

3.4 Data Handling for ANN Model

Data handling is the most important step before feeding to the network because it determines the success of any neural network model. Neural network training can be made more efficient if certain pre-processing steps are performed on the network inputs and targets. Another post-processing step is needed to transform the output of the trained network to its original format. These two steps are explained below.

3.4.1 Data Collection and Partitioning for ANN Model

A total of 338 data sets were collected from Middle East fields. Three data sets had been removed as outliers according to the semi-studentised residual or (standard residual). The data set with a semi-studentized residual value of 2.0 and above has been considered as an outlier and hence removed accordingly.

$$e_i^* = \frac{e_i}{\sqrt{MSE}} \quad (3.2)$$

Where; e_i^* is the semi-studentised residual (or standard residual);

MSE is the mean square error of the data;

e_i is the residual

So, a total of 335 data sets had been used for the generation of the ANN model. The most relevant parameters involved in estimation of pressure drop in pipeline systems were carefully selected. Validity of the collected data was first examined to remove the data that were suspected to be in error. For this purpose, the most extensively used empirical correlations and mechanistic models were used to obtain predictions of the pressure drop for all data. These were the mechanistic models of Xiao et al, Gomez et al and the correlation of Beggs and Brill. The reason for selecting the above mentioned models and correlation is that they have been extensively used by the industry for the estimation of pressure drop in pipelines under all angles of inclination, [Xiao *et al.*, 1990] and [Gomez *et al.*, 1999].

3.4.2 Partitioning

Partitioning the data is the process of dividing the data into three different sets: training sets, validation sets, and test sets. By definition, the *training set* is used to develop and adjust the weights in a network; the *validation set* is presented to the network during training phase to ensure the optimum generalization of the developed network, and the *test set*, which is not be seen by the network during training, is used to examine the final performance of the network. The primary concerns should be to ensure that: (a) the training set contains enough data, and suitable data distribution to adequately cover the entire range of data, and (b) there is no unnecessary similarity between data in different data sets. Different partitioning ratios were tested (2:1:1, 3:1:1, and 4:1:1).

Normally, the more training cases submitted to the network the better performance can be obtained. However, the hazard of memorization becomes possible if this partitioning ratio is applied. So a ratio of 2:1:1 was used in this study.

One half of the data had been reserved for training; one quarter of the data had been kept for validation and one quarter had been maintained for testing network performance. This categorization corresponds to 168 data set reserved for training the model while 83 data sets were utilized for validation purposes. The last 84 data set had been kept aside for testing the new model performance. The testing set was hidden by the network during training and validation.

3.5 ANN Model Development

This section introduces in details the proposed ANN topology, model features, and the final model architecture.

3.5.1 Introduction

Neural networks are used as computational tools with the capacity to learn, with or without teacher, and with the ability to generalize. Among all types of available networks, the most widely used are a multiple-layer feed forward networks that are capable of representing non-linear functional mappings between inputs and outputs. The developed model consisted of one input layer (contains eight input neurons or nodes), which represent the parameters involved in estimating pressure drop in pipelines (oil rate, water rate, gas rate, diameter of the pipe, length of pipe, wellhead pressure, wellhead temperature, and angle of deviation), two hidden layers (the first one contained nine nodes, the second hidden layer contained four nodes) and one output layer (contains one node) which is pressure drop. This topology had been achieved after a series of optimization processes by monitoring the performance of the network until the best network structure was accomplished. The procedure of network optimization will be described thoroughly in the Section 4.1.1.

3.5.2 ANN Model's Features

The developed model simply pivoted on a set of processing units called neurons equivalent to eight input variables: oil rate, water rate, gas rate, pipe diameter, pipe length, wellhead pressure, wellhead temperature, and angle of deviation.

The model also contained an activation state for each unit, which is equivalent to the output of the unit. Moreover, links between the units were utilized to determine the effect of the signal of each unit. Besides, a propagation rule was used to determine the effective input of the unit from its external inputs. An activation function (in this model logistic function was used for hidden units and linear for output unit), which were applied to find out the new level of activation based on the effective input and the current activation. Additional term was included in the final topology, which was an external input bias for each hidden layer to offer a constant offset and to minimize the number of iterations during training process. The key feature of the model was the ability to learn from the input environment through information gathering (learning rule).

3.5.3 ANN Model Architecture

The number of layers, the number of processing units per layer, and the interconnection patterns between layers define the architecture of the model. Therefore, defining the optimal network that simulates the actual behavior within the data sets is not an easy task. To achieve this task, certain performance criteria were followed. The design started with a few numbers of hidden units in the only hidden layer that it acts as a feature detector. Some rules of thumb were used as guides; for instance, the number of hidden units should never be more than twice as large as the input layer, [Berry and Linoff, 1997].

In addition to this rules, several rules were suggested by different authors. Those rules can only be treated as a rough estimation for defining hidden layers size. Those rules ignored several facts such as the complexity and the discontinuities in the behavior under study. In addition, they did not count for the number of training set size. The basic approach used in constructing the successful network was trial and

error. The generalization error of each inspected network design was visualized and monitored carefully through plotting the governing statistical parameters such as correlation coefficient, root mean squared errors, standard deviation of errors, and average absolute percent error of each inspected topology. Another statistical criterion (maximum validation error) was utilized as a measure of accuracy of the trained model. Besides, a trend analysis for each inspected model was conducted to see whether that model simulated the real behavior. Data randomization is necessary in constructing a successful model, while a frequently found suggestion is that input data should describe events exhaustively; this rule of thumb can be translated into the use of all input variables that are thought to have a problem-oriented relevance. These eight selected input parameters were found to have pronounced effect in estimating pressure drop.

3.6 Network Selection

Different network topologies had been tried in an essence of finding the optimum network architecture. Among them, back-propagation network with feed-forward algorithm gained pronounced publicity in solving hard problems, especially in petroleum engineering. However, back-propagation network with feed-forward cycle reported to have several shortcomings. One of the main problems associated with this type of networks is its trapping in local minima instead of global minima. In addition, slow convergence where the network fails in several occasions to converge to the optimum solution is witnessed.

To avoid such shortcomings, resilient back-propagation network (special type of general back-propagation scheme) had been tried in this research in an attempt to generate a successful model for estimating pressure drop in pipeline with a wide range of angles of inclination. This algorithm is working under the scheme of local adaptive learning for supervised learning feed-forward neural networks. The reason for selecting such network topology is its fast convergence compared to other network schemes. Additional reason is that resilient back-propagation, on contrary to other gradient descent algorithms, which count for the change of magnitude of weight

derivative and its sign; it only counts for the sign of the direction of weight. More elaboration on the resilient back-propagation was given in Appendix A.

3.6.1 Network Training

The network had been trained using resilient back-propagation training scheme. The training parameters were modified several times until the optimum performance had been achieved. In this part number of modified training parameters will be presented along with justification of each case. Maximum number of iterations had been set to 500 epochs since the resilient back-propagation is famous of its fast convergence. After small number of iterations, the network converged to the optimum solution. Maximum validation failures had been set to 6 cases only since great number of failed validation cases may affect the network stability and generality when new cases are presented to the network.

Learning rate is used to enhance the training speed and efficiency. This factor had been varied between the values of 0.5 to 1.5 while the performance was monitored carefully. A value of 1.05 was found to achieve the fastest and most efficient training performance. However, the increase and decrease factors η^+ and η^- were set to fixed values: $\eta^- = 0.5$ and $\eta^+ = 1.2$. These were reported in MATLAB script as (net.trainParam.delt_dec = 0.5 & net.trainParam.delt_inc = 1.2). Initial weight change was kept at its default value (net.trainParam.delta0 = 0.07) in order to avoid the escalating values of weights. The maximum weight-step determined by the size of the update-value had been limited. The upper bound was set by the second parameter of *RPROP*, Δ_{\max} . The default upper bound was set somewhat arbitrarily to $\Delta_{\max} = 50.0$ and it was reported in MATLAB script as (net.trainParam.deltamax = 50). Usually, the convergence is rather insensitive to this parameter as well. The minimum step size was always fixed to a value $\Delta_{\min} = 1e^{-6}$.

3.6.1.1 *Application of Validation Set*

Validation set was presented to the trained network during the training process in an attempt to avoid over-fitting problem. The over-fitting problem is associated with network memorizing the presented training set rather than learning the hidden relationship between input parameters and the target output. The validation set was in parallel presented with the training set while the error gradient was monitored for each set. Over-fitting started to occur when the validation error started increasing while the training error continued decreasing. At this point of time, the training had to be ceased and weights and biases of the trained network had to be restored while the validation error at its minimum value.

3.7 Output Post-Processing (Denormalization)

This step was needed for presenting results of ANN model. This can be done in a meaningful way after model generation and it can be challenging, yet perhaps the most important task. This was needed to transform the outputs of the network to an understood value by reverting the original value used. It is the stage that comes after the analysis of the data and is basically the reverse process of data pre-processing.

3.8 Software Used

MATLAB software (version R2007a), [MATLAB, 2009], environment was utilized due to its high range of flexibility associated with programming and graphs visualization. Moreover, the software offers a good way to monitor the performance of the three set of data (training, validation, and testing) at the same time. A MATLAB code was developed and training parameters were modified in order to ensure that these parameters are well optimized. The final model structure is shown in Fig 4.8. The problem encountered during training was the trapping of the model in a local minima several times. The reason behind this problem was found to be the low range of certain variables in the data. The concept of local and global minima was discussed by several mathematicians and ANN researchers, [Gori and Tesi, 1992].

It defines the error gradient surface as if it is seen as a hyperbolic surface where the global minima point lies at the bottom of this surface. Hence, beyond this point the error starts to increase dramatically. The training session should be exactly stopped at this point to assure the generalization of the developed model.

The default of the software is to escape only five local minima. This option had been varied several times in order to allow the network to capture the real behavior between input parameters and output (pressure drop). The problem of under-fitting and over-fitting (using too few; and too many units in hidden layer, respectively) was avoided through the use of cross-validation data set and application of early stopping technique. Cross-validation data set was presented to the network after each epoch of training to check the generality (model succeeded to capture minor relationships between input set and the desired output when new cases are submitted to it) and stability of the model, [Haykin, 1999].

Input weight matrix (from input to the hidden layers), hidden layer weight matrices, and the layers bias vectors for the retained network, all were extracted from this program and presented in Appendix B. These weights and biases were utilized in developing an executable code, which provides an easy way for users to implement in predicting pressure drop values.

3.9 Network Performance Comparison

Pressure drop calculation for Beggs and Brill correlation (1973), Gomez et al. model (1999), Xiao et al. model (1990) had been conducted using the freeware *DPDLSystem*. The software allows great flexibility in selecting PVT methods, type of pressure drop correlation (vertical, inclined, and horizontal), operating conditions, and flow-rate type data. Only test data had been chosen for comparison for each selected model against the proposed ANN and AIM models. The network performance comparison had been conducted using the most critical statistical and analytical techniques. Trend analysis, group error analysis, and graphical and statistical analysis are among these techniques.

3.9.1 Trend Analysis

A trend analysis was performed for each generated model to check whether it was physically correct or not. Interchangeably, this analysis is the synonyms of sensitivity analysis. This analysis aids in fully understanding the relationship between input variables and output and increases the robustness of the generated model. However, it serves as a major ingredient in assessing model building and quality assurance.

For this purpose, synthetic sets were prepared where in each set only one input parameter was varied between the minimum and maximum values while other parameters were kept constant at their mean (base) values. This means that each input parameter was changed Once-At-a-Time (OAT) to check its effect at the final output. This helped increase the comparability of the results (all 'effects' are computed with reference to the same central point in space).

3.9.2 Group Error Analysis

To demonstrate the robustness of the developed model, another statistical analysis was conducted, which was group error analysis. The purpose of this analysis was to quantify the error produced by each input when grouped to a number of classes based on the average absolute relative error as an indicator. The reason for selecting average absolute relative error is that it is a good indicator of the accuracy of all empirical correlations, mechanistic model; as well as for the new developed models. This effective comparison of all investigated correlations and mechanistic models provides a good means of evaluating models performance. Average absolute relative error was utilized in this analysis by grouping input parameter and hence plotting the corresponding values of average absolute relative error for each set.

3.9.3 Statistical Error Analysis

This error analysis had been utilized to check the accuracy of the models. The statistical parameters used in the present work were: average percent relative error, average absolute percent relative error, minimum and maximum absolute percent

error, root mean square error, standard deviation of error, and the correlation coefficient. Those statistical parameters are well known for their capabilities to analyze models' performances, and have been utilized by several authors, [Ayoub, 2004], [Osman, E. A. *et al.*, 2001], and [El-Sebakhy *et al.*, 2007]. Equations for those parameters are given below.

3.9.3.1 Average Percent Relative Error (APE)

It is the measure of relative deviation from the experimental data, defined by:

$$E_r = \frac{1}{n} \sum_{i=1}^N E_i \quad (3.3)$$

Where; E_i is the relative deviation of an estimated value from an experimental value

$$E_i = \left[\frac{(\Delta P)_{meas} - (\Delta P)_{est}}{(\Delta P)_{meas}} \right] \times 100, \quad i = 1, 2, 3, \dots, n \quad (3.4)$$

where;

$(\Delta P)_{meas}$ is the actual value of pressure drop

$(\Delta P)_{est}$ is the estimated value of pressure drop

3.9.3.2 Average Absolute Percent Relative Error (AAPE)

It measures the relative absolute deviation from the experimental values, defined by:

$$E_a = \frac{1}{n} \sum_{i=1}^n |E_i| \quad (3.5)$$

This will be considered as the main criterion in statistical error analysis throughout this study. AAPE or MAPE (Mean Absolute Error) has invaluable statistical properties in that it makes use of all observations and has the smallest variability from

sample to sample, [Levy and Lemeshow, 1991]. The term is easy to calculate and simple to understand since it is presented in a percentage unit.

To meet the criteria for a good measure of error, AAPE should satisfy five conditions (as stated by the American National Research Council). These are; measurement validity, reliability, ease of interpretation, clarity of presentation and support of statistical evaluation, [American National Research Council., 1980].

AAPE meets most of these conditions except the validity, which is highly suspected under certain circumstances. One of these circumstances is the distribution of the absolute percent errors and is often asymmetrical and right skewed. Hence, few outliers can affect and dominate it, [Hoaglin *et al.*, 1983]. However, the problem of outliers' dominance had been resolved in modeling process whereas removal of suspected outliers was done before feeding clean data to software.

3.9.3.3 Minimum Absolute Percent Relative Error

$$E_{\min} = \min_{i+1}^n |E_i| \quad (3.6)$$

3.9.3.4 Maximum Absolute Percent Relative Error

$$E_{\max} = \max_{i+1}^n |E_i| \quad (3.7)$$

3.9.3.5 Root Mean Square Error (RMSE)

Measures the data dispersion around zero deviation, defined by:

$$RMSE = \left[\frac{1}{n} \sum_{i=1}^n E_i^2 \right]^{0.5} \quad (3.8)$$

3.9.3.6 Standard Deviation (SD)

It is a measure of dispersion and is expressed as:

$$STD = \sqrt{\left[\left(\frac{1}{(m-n-1)} \right) \sum_{i=1}^m \left\{ \left(\frac{\Delta P_{act} - \Delta P_{est}}{\Delta P_{act}} \right) \right\}^2 \right]} \quad (3.9)$$

Where; (m-n-1) represents the degree of freedom in multiple- regression. A lower value of standard deviation indicates a smaller degree of scatter.

3.9.3.7 The Correlation Coefficient (R)

It represents the degree of success in reducing the standard deviation by regression analysis, defined by:

$$R = \sqrt{1 - \frac{\sum_{I=1}^n [(\Delta P)_{act} - (\Delta P)_{est}]^2}{\sum_{I=1}^n (\Delta P)_{act}^2 - \overline{\Delta \Delta P}^2}} \quad (3.10)$$

Where;

$$\overline{\Delta \Delta P} = \frac{1}{n} \sum_{I=1}^n [(\Delta \Delta P)_{act}]_I \quad (3.11)$$

'R' values range between 0 and 1. The closer value to 1 represents perfect correlation whereas 0 indicates no correlation at all among the independent variables. R-value is a quantity that measures the quality of a least squares fitting to the original data. By definition, least squares fitting is the procedure by which the best fitting curve is found to a given set of points by minimizing the sum of the squares of the offsets (the residuals) of the points from the curve, [Bevington and Robinson, 2003]. However, because squares of the offsets are used, outlying points can have a disproportionate effect on the fit. For this reason R-quantity can be used as an indicative feature for goodness-of-fit.

3.9.4 Graphical Error Analysis

Graphical tools aid in visualizing the performance and accuracy of a correlation or a model. Three graphical analysis techniques are employed; those are cross-plots, error distribution, and residual analysis.

3.9.4.1 Cross-plots

In this graphical based technique, all estimated values had been plotted against the measured values and thus a cross-plot was formed. A 45° straight line between the estimated versus actual data points was drawn on the cross-plot, which denoted a perfect correlation line. The tighter the cluster about the unity slope line, the better the agreement between the actual and the predicted values. This may give a good sign of model coherence.

3.9.4.2 Error Distributions

Error distribution displayed the error sharing histograms for the neural network model, (training, validation, and testing sets) and the AIM model. Normal distribution curves had been fitted to each one of them. The errors are said to be normally distributed with a mean around 0% and the standard deviation equal to 1.0.

The normal distribution is often used to describe, at least roughly, any variable that tends to cluster around the mean. In our case it was used to describe the error tendency around the mean, (which is alternatively known as a normal or Gaussian distribution).

Hence, some of the investigated models showed either slight to considerable negatively skewed error distributed or positively ones.

3.9.4.3 *Residual Analysis*

The relative frequency of deviations between estimated and actual values was conducted for all investigated models. The residual analysis showed the error distribution around the zero line to verify whether models and correlation have error trends. Analysis of residual (predicted pressure drop minus the actual pressure drop) is an effective tool to check model deficiencies.

3.10 Building AIM Model

The process of generating AIM Model started by selecting the same input parameters used for generating the previous ANN Model. The same order of data sets had been maintained during generation of a polynomial GMDH Model. A free software was being used for this purpose [Jekabsons, 2010]. This source code was tested with MATLAB version 7.1 (R14SP3). Despite the software allows great flexibility in selecting the model parameters, it also provides ample interference. The detailed model's inputs and produced outputs notions had been defined in Appendix B (AIM code -polynomial network code). However all of the input parameters had been used in generating the model.

3.11 Uncertainty Study

This Section deals with studying causes of uncertainty in the model generated by ANN and AIM techniques. No prior information has been given for biases in the data used in generating models. However, the uncertainty study will focus on evaluating error estimates of the generated models plus other investigated models for sake of comparison. Uncertainty associated with data has been calculated. Moreover, the study also shows the confidence level of each generated model, as well as for other investigated models. Additionally, mathematical representation of model's uncertainty calculation will be presented. Great emphasis will be devoted to ANN model since it outperformed the rest of studied models.

The study also covers sensitivity analysis of input variables and their effect on the ANN and AIM model's prediction capabilities.

3.11.1 Definition

For a certain derived quantity, a model will be generated to simulate this measured quantity based on certain fundamental scientific principles.

Uncertainty that is related majorly to the type of errors compromises has two components; systematic and random. Systematic uncertainty is mainly due to fault in measuring instrument or the used technique. Unavoidable errors remain as events and can be treated as random uncertainty, [Bevington and Robinson, 2003]. As clearly shown from this definition, systematic uncertainty or the bias can be compensated or minimized through different techniques, while random uncertainty is hard to be accounted for.

3.11.2 Uncertainty in ANN Modeling

Uncertainty in ANN modeling can be attributed to the following:

1- Uncertainty in data acquisition

Uncertainty in data acquisition remains the biggest source of error in generating ANN model. The process of data acquisition includes calibration of measuring devices and gages with limited precision. The source of errors might be due to reading, storing and human errors.

2- Uncertainty in model structure

Uncertainty in model structure includes definition of network type, adopted training algorithm, network topology, and type of transfer and cost functions.

3- Uncertainty due to incomplete information

Pressure drop estimation might need additional input parameters such as viscosity of liquid phase, pipe roughness, specific gravity of gas and water, etc. This can be additional cause of model uncertainty. Due to the lack of data in hand this property could not be investigated precisely.

4- Uncertainty due to model parameter values

The range of the data may contribute enormously to the uncertainty of ANN model. While the current range produced optimum results, a wider range of data can add to the trustworthiness of model and confidence.

5- Uncertainty with model output

Again, the output of the model could be widened to include a broad range of data with more intense variation across the values.

3.11.2.1 *Measurement of Uncertainty in ANN Model*

Quantification of uncertainty in ANN model had been done using Variance-Based method. This method produces robust quantitative results irrespective of the models' behavior. One way to judge the uncertainty of the model output is to use the error variance; a large variance of the model error usually indicates that the model prediction is uncertain. The variance-based method is popular, which derives from the decomposition of the total variance of a model output into variances due to different input variables and their combinations. Random uncertainty is closely attached to standard deviation of error (SD). Additionally, variance (SD^2) involves squaring differences of observed value from the mean.

3.11.2.2 *Propagation of Uncertainty*

The term propagation of uncertainty or what is synonymously known as error propagation is explaining the theory of error analysis which gives a general formula for the uncertainty when a result is found by a calculation from a collection of measurements, [Bevington and Robinson, 2003]. The formula is based on the idea of

a first-order Taylor series expansion of functions of many variables. It is valid when the various uncertainties σ_i of the i different variables are small compared to the values of the quantities and on the requirement that the uncertainties are uncorrelated with each other. Specifically, if the desired result is a well-behaved function $f(x, y, z, \dots)$ of the physical variables x, y, z, \dots which have uncertainties $\sigma_x, \sigma_y, \sigma_z, \dots$, then the uncertainty in the value of the result σ_f is given by the formula:

$$d_f^2 = \sigma_x^2 \left(\frac{df}{dx} \right)^2 + \sigma_y^2 \left(\frac{df}{dy} \right)^2 + \sigma_z^2 \left(\frac{df}{dz} \right)^2 + \dots \quad (3.12)$$

where the partial derivatives are all evaluated at the best known values of x, y, z, \dots

Based on this formula, no dependency between actual and predicted pressure drop values for a given set of data can be reported. Thus for summing propagating uncertainty, the following procedure had been followed:

- 1- Evaluate the standard deviation for each data set
- 2- Get the average value between predicted and actual pressure drop
- 3- Evaluate the covariance around the mean for each set according to the following formula;

$$CV = \frac{STD}{Mean} \quad (3.13)$$

- 4- Square the covariance value for each data set
- 5- Get the summation of the squared covariance values
- 6- Evaluate the relative standard deviation around the mean by using the following formula:

$$RSD = \sqrt{\frac{(CV)^2}{N}} \quad (3.14)$$

Where N is number of samples used for testing models

7- Finally, evaluate the expanded measured uncertainty at 95% confidence interval and at the designated t-student distribution and degree of freedom values.

3.12 Sensitivity Analysis

Sensitivity analysis had been conducted to show the role of each input parameter incorporated in ANN model. This analysis will help give more insight into the contribution of each parameter and lessen the argument that says ANN is a black-box, [Hamby, 1994]. It is the study of how the variation (uncertainty) in the output of a given model can be apportioned, qualitatively or quantitatively, to different sources of variation in the input of the model, [Saltelli and ebrary, 2008].

Some of the shortcomings of sensitivity analysis include:

- The range of outcome values between the high and low percentiles might not reveal some of the uncertainty involved, especially if the maximum divergence from the best-guess value occurs in the interior of the range, [Bankes, 1993].
- It is not possible to model stochastic variability through this method. Thus, it is not a replacement for conducting uncertainty analysis.
- Performing a sensitivity analysis on a given model is based on the premise that the model structure is correct. It does not measure or detect specification error.

For conducting this study, a well known approach had been utilized, [Kemp *et al.*, 2007]. The approach called HIPR (Holdback Input Randomization Method) which involves the following steps to achieve understanding of relative importance of input variables by systematically altering input data patterns.

1- Using the test data set to determine relative input parameter importance:

a- Sequentially feeding each data point in the test data set to the ANN but replacing the values of one input parameter by uniformly distributed random values in the interval (-1,1), the range over which the net was originally trained,

b- Calculating the mean squared error of the ANN when the randomized test set has been presented, and

c- Repeating the procedure for each input parameter, each time substituting the original values with uniformly distributed random values.

The MSE values of the data set with a particular randomized parameter in relation to the MSE of the original data set reflect the relative importance of that input parameter for the prediction of the ANN.

3.13 Limitations

The results of the two generated models may be limited in their nature due to data attributes range. For example extrapolation may produce erroneous results. The models results can be only applied within the trained data range. The assigned results may suffer degradation due to type of data used in generating both models. However, the accuracy obtained by both models depends on the range of each input variable and the availability of that input parameter (parameters). Although the main purpose was to explore the potential of using both ANN and GMDH techniques, the optimum performance can be obtained using this limited data range in attributes and variables. However, care must be taken if obtained results are applied for data type and range beyond that used in generating both models. It is worthy to mention that no assumptions have been made during the process of models generation.

3.14 Summary

This Chapter described the general framework of the problem being addressed. The Chapter also discussed issues related to data preparation before feeding to ANN modeling. A brief description of SCADA system has been presented as well. Data collection and partitioning have been thoroughly discussed. In addition, systematic procedure used for developing the ANN model has been critically presented. The most critical statistical and analytical techniques used for comparing the performance of the two developed models (ANN & AIM) had been presented. The concepts and

mathematical representations of trend analysis, group error analysis, and graphical and statistical analysis were thoroughly addressed. The flow of the Chapter continued with the description of the process of generating AIM Model. Also, issues related to causes of uncertainty of ANN model and how these can be quantified have been discussed. Additionally, a new method for rating the role of each input parameter in ANN model (sensitivity analysis) has been presented. Finally, the Chapter concluded with the presenting the main models' applicability limitations and shortcomings.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the results of the generated models (ANNs and AIM) will be discussed. Firstly, the ANN model's optimization will be discussed. Next, a detailed trend analysis of the developed ANN model is presented to examine whether the model simulates the physical behavior. The model performance will be assessed by statistical and graphical visualization.

The ANN model's performance will be compared against some of the best available empirical correlations and mechanistic models adopted by the industry. The reasons for selecting these empirical correlations and mechanistic models because they have been used extensively by the industry and their results are trusted by the industry too. Secondly, Abductive Network technique will be utilized to generate another model by applying a code generated by MATLAB software (version R2007a) package. The technique is capable of producing a model with the most effective parameters that affect the output target. Again, the model performance will be checked against the proposed ANN model.

The chapter concludes with conducting comprehensive uncertainty study for the two generated models, as well as for the rest of investigated models. Additionally, a modified method for evaluating the relative importance of each input parameter involved in ANN model had been implemented.

4.1.1 ANN Model Optimization

The optimum number of hidden units depends on many factors:

- 1- The number of input and output units
- 2- The number of training cases
- 3- The amount of noise in the targets
- 4- The complexity of the error function
- 5- The network architecture
- 6- The training algorithm.

In most cases, there is no direct way to determine the optimal number of hidden units without training using different numbers of hidden units and estimating the generalization error of each one.

To further describe the process of optimizing the model; Fig 4.1 and Fig 4.2 illustrate the effect of changing number of neurons in the first hidden layer on the average absolute percent error and correlation coefficient. As observed from Fig 4.1 and Fig 4.2, one hidden layer with nine hidden neurons is achieving the lowest average absolute percent error and the highest correlation coefficient. But, on the other hand, the model failed in producing the correct physical trend across the data range. Instead of that, additional hidden layer was added and number of hidden nodes was increased gradually until the correct trend was achieved.

The selection of this model was based on having the highest correlation coefficient for the testing and validation sets. But still the performance of the model was not good enough and the inherent relationship between input variables and the output was not well extracted. The whole procedure was discarded when it was found that obtaining the right trend cannot be achieved easily through application of traditional back-propagation training algorithms such as gradient descent, and gradient descent with momentum. They are very slow when compared to other algorithms such as resilient back-propagation. The latter had been used in training the model because of its great advantages over the other training algorithms.

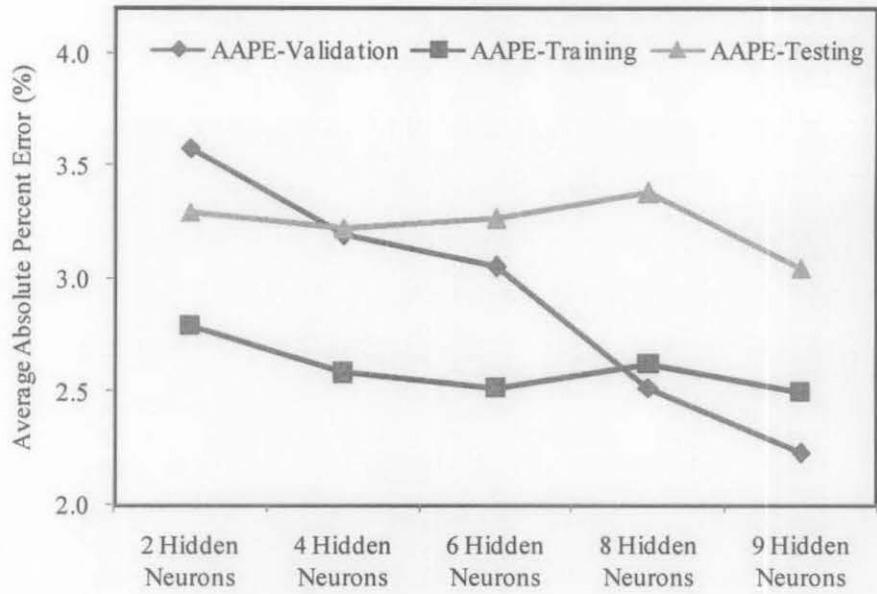


Fig 4.1: Effect of Changing Number of Neurons in First Hidden Layer on Average Absolute Percent Error

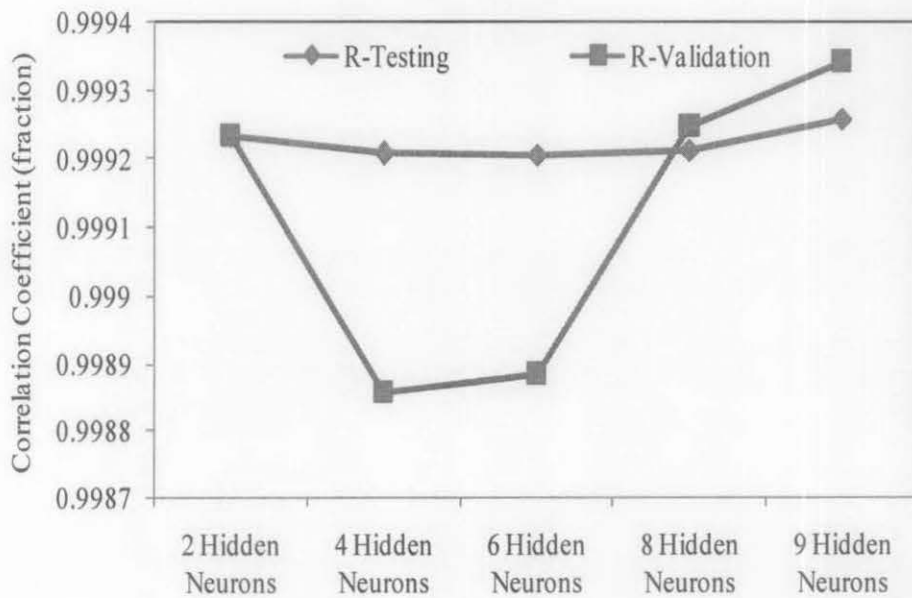


Fig 4.2: Effect of Changing Number of Neurons in First Hidden Layer on Correlation Coefficient.

4.1.2 The Resilient Backpropagation Algorithm (RPROP)

Theoretical background about the RPROP is provided in Appendix A. The model's training started by selecting small number of hidden neurons in the first hidden layer and, hence monitoring (recording) the performance of each topology. Any topology that failed to produce the correct physical trend was discarded. Only three successful topologies had been recorded and prepared for comparison. The first model consisted of seven hidden nodes in the first hidden layer while the second model consisted of twelve hidden nodes in the first hidden layer. The performance of these two networks was not up to satisfaction. It was decided to increase the number of hidden layers to reach two and slightly increasing the number of hidden nodes until a topology that represents the inherent relationship between input parameters and the target output was captured. Only one structure was successful in producing the correct physical trend which was a network of nine nodes in the first hidden layer and four in the second hidden layer. Results of successful networks in terms of average absolute percent error and correlation coefficient are tabulated in Table 4.1. However, maximum error of each set was presented as a good governing statistical criterion for selecting the model of the lowest value, was tabulated in Table 4.2. In addition, Table 4.3 presents the root mean square errors and standard deviations of errors for validation and testing sets which will aid in selecting the best model that has the lowest value.

Table 4-1: Effect of Changing Number of Neurons with respect to Average Absolute Percent Error and Correlation Coefficient

Architecture	AAPE (TEST)	AAPE (TRAIN)	AAPE (VALID)	R (TEST)	R (TRAIN)	R (VALID)
8-7-1	15.44	18.04	19.45	0.98196	0.95567	0.94699
8-12-1	11.61	14.50	22.56	0.98708	0.97842	0.95276
8-9-4-1	12.11	12.38	17.50	0.98821	0.9889	0.96705

Table 4-2: Effect of Changing Number of Neurons with respect to Maximum Error for Testing, Training, and Validation Sets

Architecture	Maximum Error (TEST)	Maximum Error (TRAIN)	Maximum Error (VALID)
8-7-1	56.875	234.338	145.504
8-12-1	45.599	209.472	385.260
8-9-4-1	43.999	96.665	165.312

Table 4-3: Effect of Changing Number of Neurons with respect to Root Mean Square Error and Standard Deviation of Errors

Architecture	RMSE (VALID)	RMSE (TEST)	STD (TEST)	STD (VALID)
8-7-1	32.12	19.91	13.09	15.15
8-12-1	51.50	14.761	10.48	14.14
8-9-4-1	32.92	15.791	10.02	11.78

Graphical representation can help visualize the difference between all sets with respect to each mentioned statistical feature. Fig 4.3 shows the effect of changing number of neurons on average absolute percent error for training, testing and validation sets while using resilient back-propagation training algorithm. Fig 4.4 shows the effect of changing number of neurons on maximum error for each set using resilient back-propagation training algorithm. It is clear from this figure that the topology 8-9-4-1 presented the lower maximum error for all data sets.

Fig 4.5 shows the effect of changing number of neurons on correlation coefficient for each set using resilient back-propagation training algorithm. Again the previously mentioned topology achieved the highest correlation coefficients for all data sets.

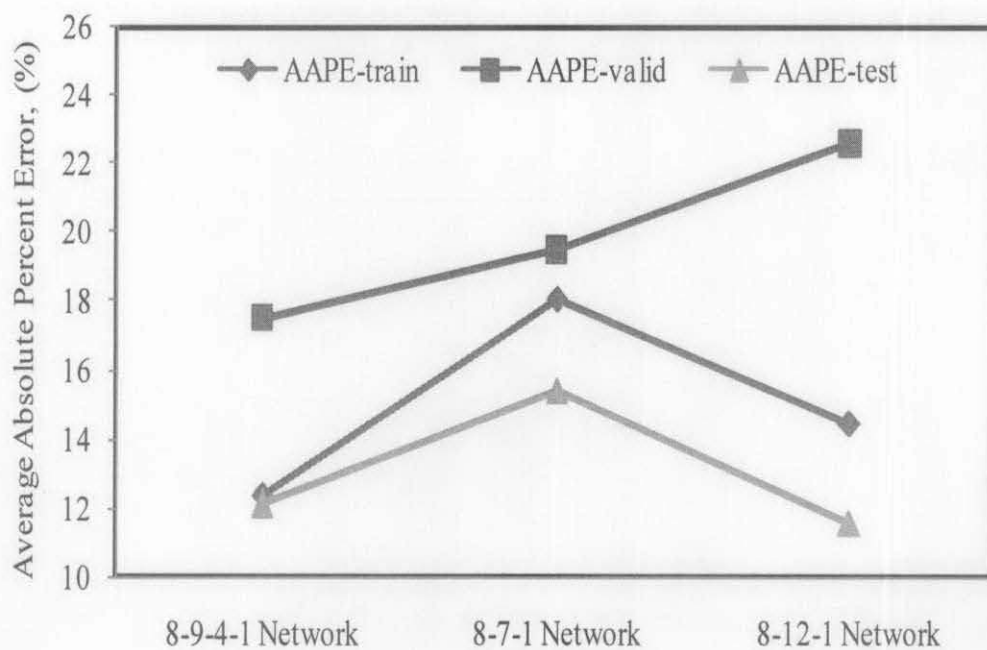


Fig 4.3: Effect of Changing Number of Neurons on Average Absolute Percent Error using Resilient Back-Propagation Training Algorithm

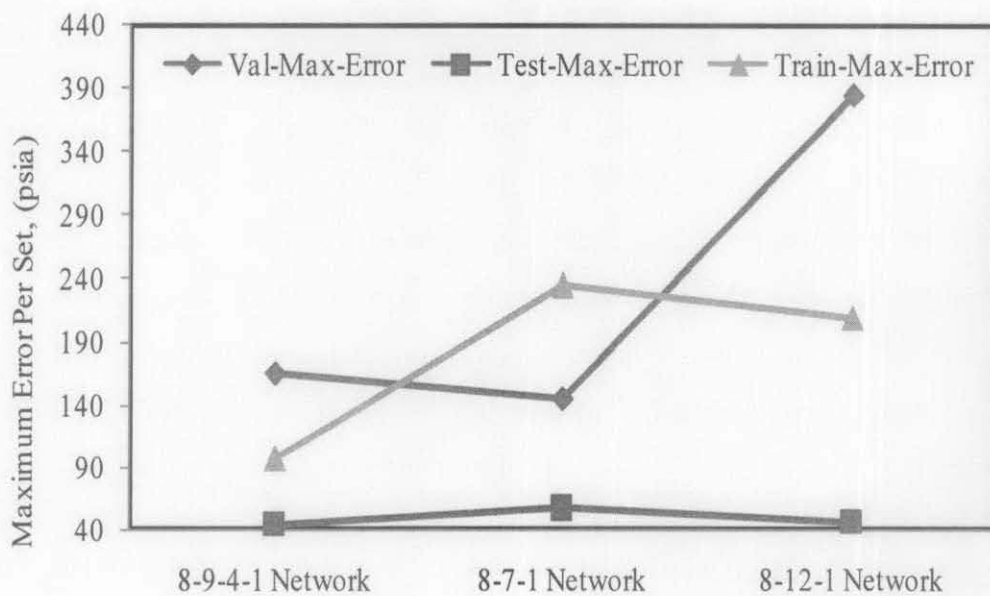


Fig 4.4: Effect of Changing Number of Neurons on Maximum Error for each set using Resilient Back-Propagation Training Algorithm.

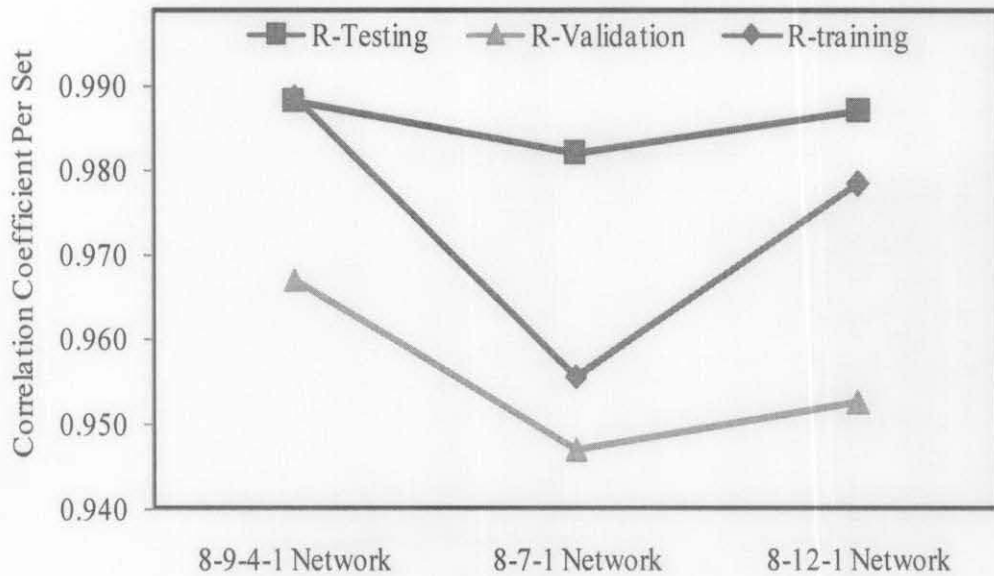


Fig 4.5: Effect of Changing Number of Neurons on Correlation Coefficient for each set using Resilient Back-Propagation Training Algorithm

Fig 4.6 depicts the effect of changing number of neurons on root mean square errors for testing and validation sets using resilient back-propagation training algorithm. In this time validation and testing sets were used as they are verifying the model performance while training set was neglected because output is seen by the network during training. Using these two sets, the same architecture (8-9-4-1) succeeded in producing the lowest root mean square errors compared to other two topologies.

Fig 4.7 illustrates the effect of changing number of neurons on standard deviation of errors for testing and validation sets using resilient back-propagation training algorithm. In this figure the architecture of 8-9-4-1 neurons was capable in attaining the lowest standard deviation of errors among all tested topologies. All these discussions and statistical analyses demonstrated that the topology of 8-9-4-1 was achieving the optimum performance among all presented topologies. In addition to that, all statistical features used to assess the performance of all investigated architectures showed that two hidden layers with nine and four hidden nodes are quite sufficient to map the relationship between the input variables and the total output

(pressure drop). This final selection of model topology was further assessed through conducting a trend analysis. The final network topology has been shown in Fig 4.8

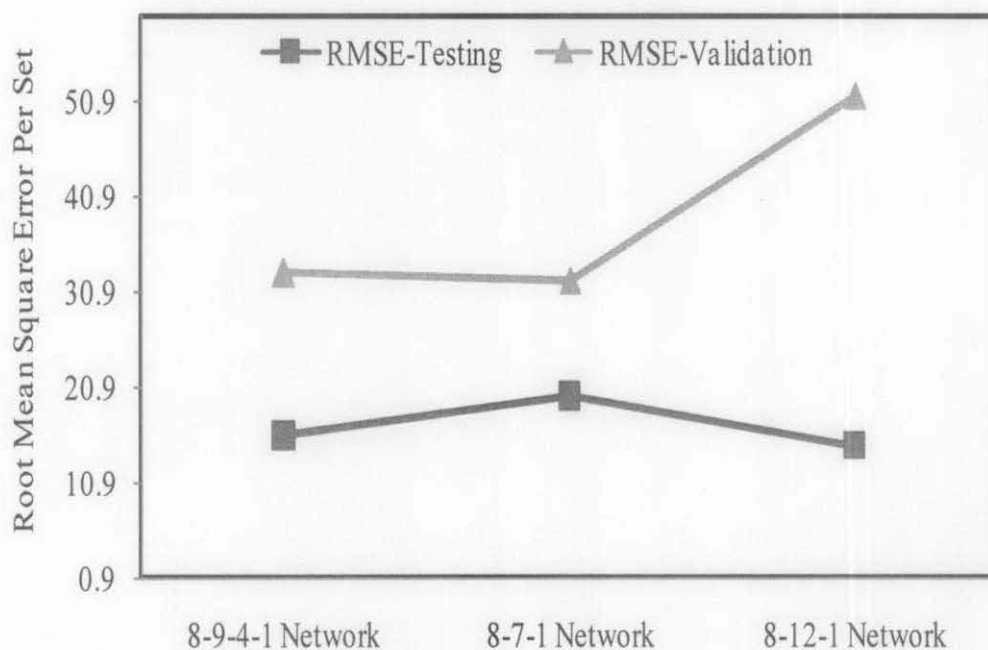


Fig 4.6: Effect of Changing Number of Neurons on Root Mean Square Errors for Testing and Validation sets using Resilient Back-Propagation Training Algorithm.

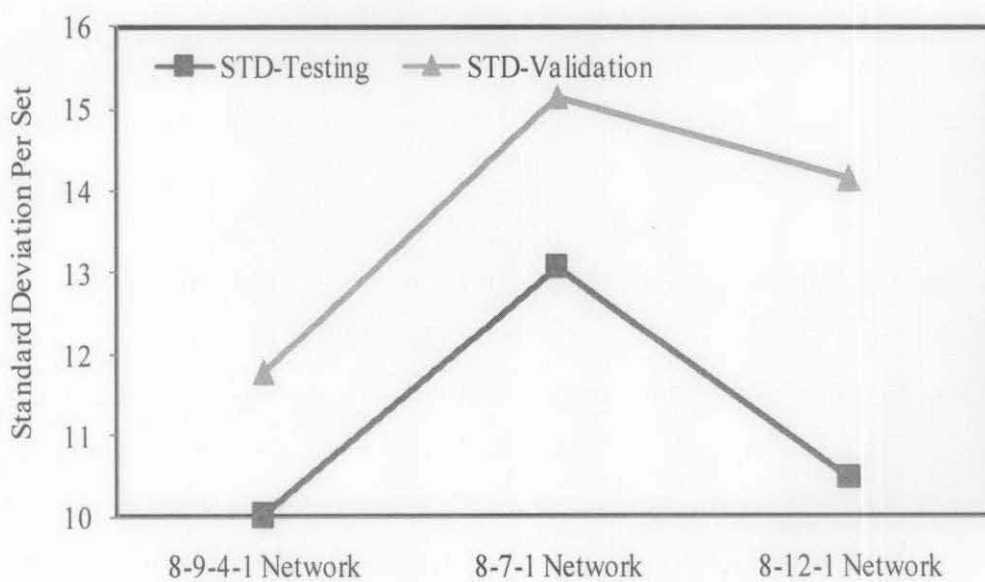


Fig 4.7: Effect of Changing Number of Neurons on Standard Deviation of Errors for Testing and Validation sets using Resilient Back-Propagation Training Algorithm.

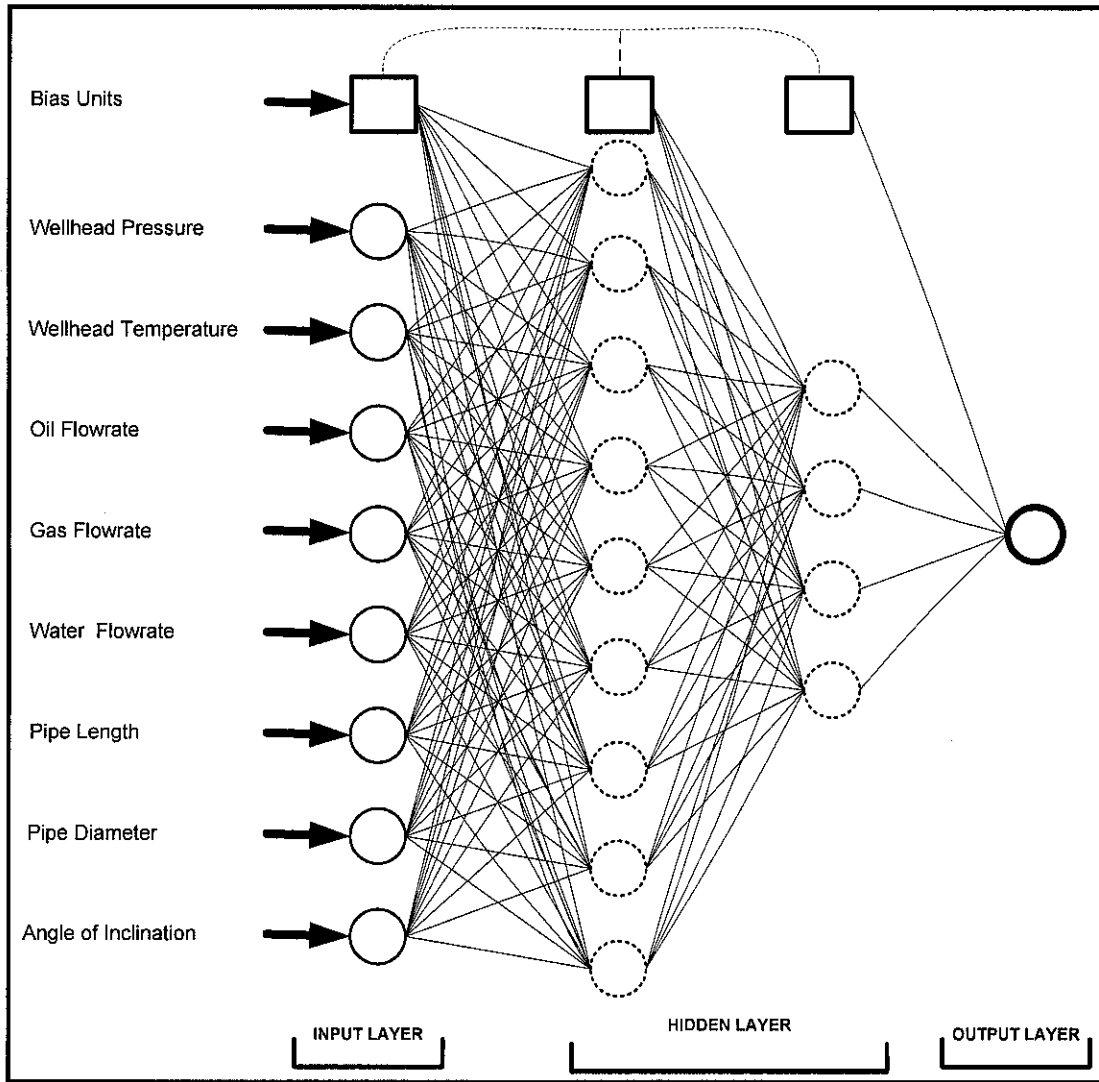


Fig 4.8: Schematic Diagram of the Developed ANN Model.

4.1.3 Objective Function for ANN Model

To train a network and measure how well it performs, an *objective function* (or *cost function*) must be defined to provide an explicit numerical rating of system performance. Selection of an objective function is very important because it represents the design goals and decides what training algorithm can be taken. A few basic functions are commonly used. One of them, which is used in this study, is the sum of the squares of the errors.

$$Error = \frac{1}{NP} \sum_{p=1}^P \sum_{k=1}^N (y_{pk} - o_{pk})^2 \quad (4.1)$$

Where, p refers to patterns in the training set, k refers to output nodes, and o_{pk} and y_{pk} are the target and predicted network output for the k th output unit on the p th pattern, respectively.

4.2 Trend Analysis for the Proposed ANN Model

A trend analysis was carried out to check whether the developed model is physically correct or not. To test the developed model, the effects of gas rate, oil rate, water rate, tubing diameter, angle of deviation and pipe length on pressure drop were determined and plotted on Fig 4.9 through Fig 4.14. The effect of angle of inclination was investigated where each parameter was plotted against pressure for different angles of inclination. This is demonstrated in Fig 4.9, which shows the effect of changing gas rate on pressure drop values. As expected, the developed model produced the correct trend where the pressure drop increases as the gas rate increases. However, a justification is needed when low gas-rate flows at vertical pipe the pressure drop should be higher than for other less valued angles. If the line is not horizontal, an increase in gas velocity will sweep some of the liquid accumulation at the lower sections of the pipe, which might lead to overall decrease in pressure drop, [Beggs, H. Dale, 2003].

This finding is compatible with the physical phenomenon according to the general energy equation, [Beggs, H. Dale, 2003] as stated in the following formula:

$$\frac{dP}{dL} = \frac{g}{g_c} \rho \sin \theta + \frac{f\rho v^2}{2g_c D} + \frac{\rho v dv}{g_c dL} \quad (4.2)$$

If the second and third term of the abovementioned equation is considered, the flow velocity is incorporated in the numerator of each term, which indicates that the pressure drop is directly proportional to the flow velocity and;

$$v = \frac{Q}{A} \quad (4.3)$$

As indicated in equation 4.4 while the cross sectional area is fixed for a given pipe size the velocity term can be used interchangeably with flow-rate. This expression is

valid for oil flow-rate, gas flow-rate, and water flow-rate. The ANN model succeeded in producing the right trend for the three phases (gas, oil, and water) as illustrated in Fig 4.9, Fig 4.10, and Fig 4.11. Another observation was reported where the pressure drop was found to be an increasing function with respect to angle value for all three phases, which is physically sound and follows the normal trend. The pressure drop had been plotted against each phase rate (oil flow-rate, water flow-rate, and gas flow-rate) for different four configurations (horizontal “0°”, vertical “90°”, inclined hilly terrain “44.6°”, and inclined downhill “-20°”).

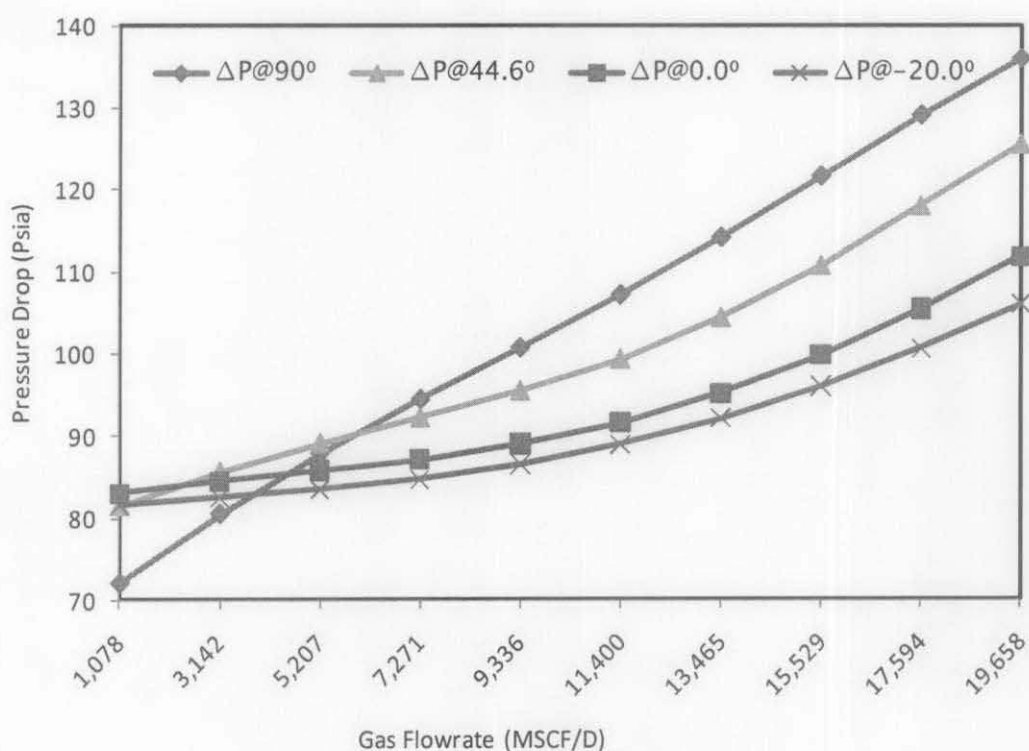


Fig 4.9: Effect of Gas Rate on Pressure Drop at Four Different Angles of Inclination.

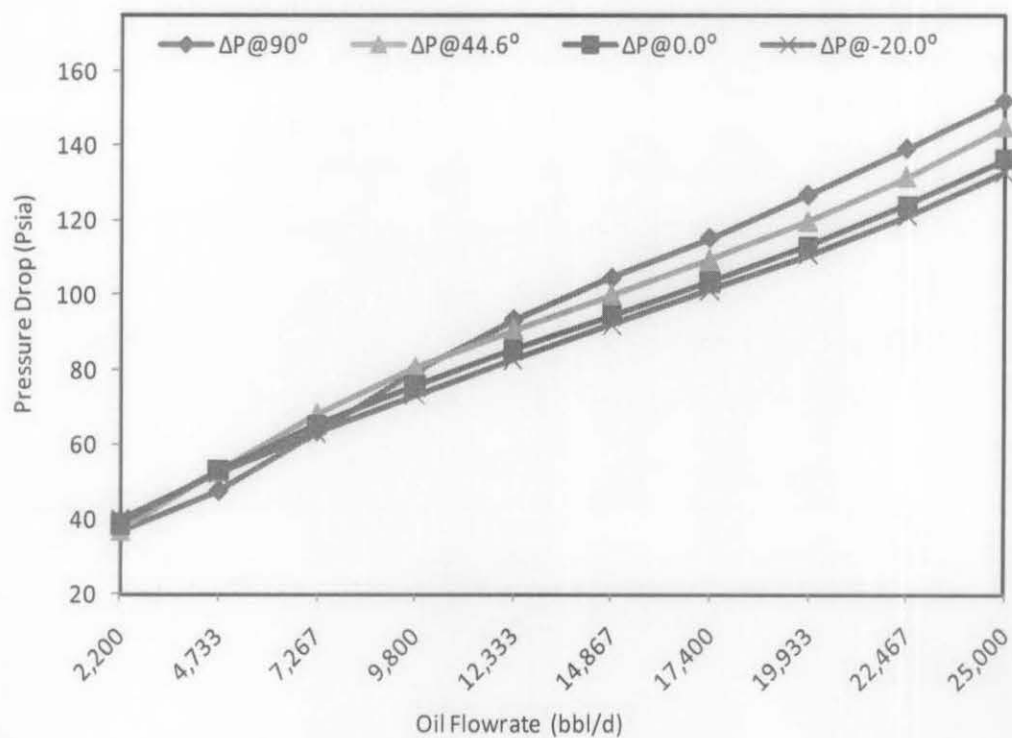


Fig 4.10: Effect of Oil Rate on Pressure Drop at Four Different Angles of Inclination.

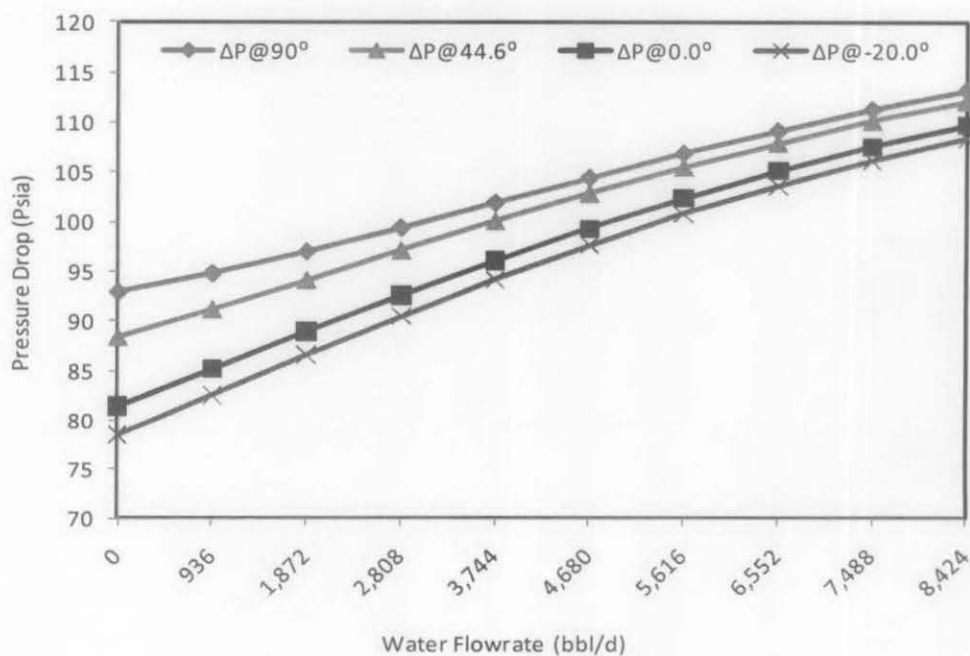


Fig 4.11: Effect of Water Rate on Pressure Drop at Four Different Angles of Inclination.

From equation 4.2 it is clear that pressure drop is inversely proportional to pipe diameter. Fig 4.12 is depicting this relationship for all pipe configurations.

However, the relationship between pressure drop and length of the pipe had been confirmed by the ANN model (pressure drop increases with increasing length of the pipe) as shown in Fig 4.13.

The effect of angle of inclination on the pressure drop had been counted for all range of investigated angles (-52 degrees to 208 degrees). Fig 4.14 shows the trend of angle of inclination with respect to pressure drop for four different pipe diameters. Again, from equation 4.2 (elevation term) sine of the angle is directly proportional to pressure drop and can be extracted as;

$$\frac{dP}{dL} \propto \frac{g}{g_c} \rho \sin \theta \quad (4.4)$$

If this equation is manipulated numerically for the investigated range of angles, it is seen that pressure drop is an increasing function from the range of -52 degrees to 90 degrees and a decreasing function beyond this range till 208 degrees. The ANN model was able to produce the correct physical behavior (according to the logic extracted from equation 4.4).

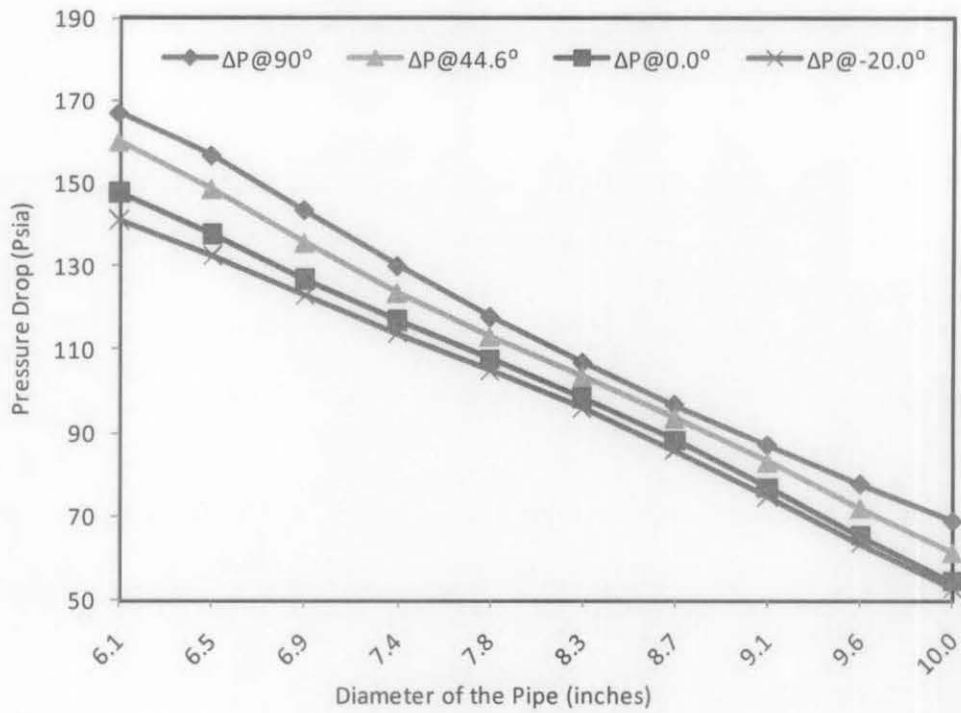


Fig 4.12: Effect of Pipe Diameter on Pressure Drop at four Different Angles of Inclination.

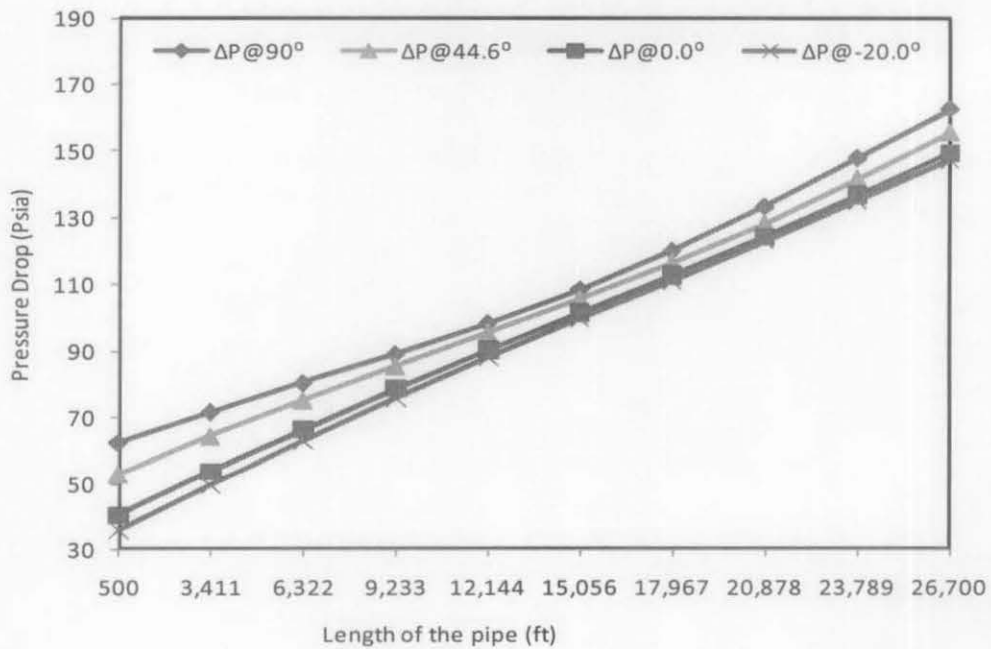


Fig 4.13: Effect of Pipe Length on Pressure Drop at four Different Angles of Inclination.

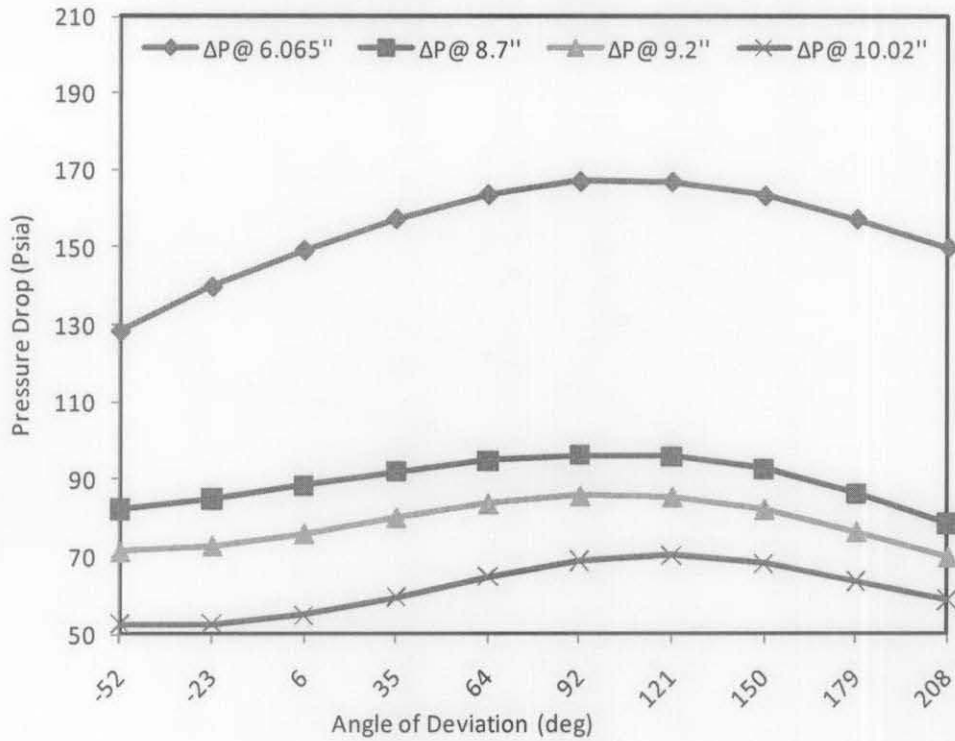


Fig 4.14: Effect of Angle of Inclination on Pressure Drop at Four Different Pipe Diameters.

To further examine the validity of the model, the trend analysis was checked at three different tubing sizes. Fig 4.15 to Fig 4.18 show the trend analysis for oil rate, gas rate, water rate and pipe length respectively.

The effect of diameter on pressure drop had been evaluated through plotting the pressure drop against oil flow-rate at three different diameters. Fig 4.15 shows the effect of changing pipe diameter on pressure drop with respect to oil flow rate. The new proposed ANN was able to produce the right trend where for a given oil-flow rate the pressure drop was found as a decreasing function with increasing pipe diameter.

However, the same procedure was followed at three different diameters with respect to gas flow rate as shown in Fig 4.16. The new proposed ANN produced a sound physical trend where for a given gas-flow-rate the pressure drop decreases as pipe diameter increases.

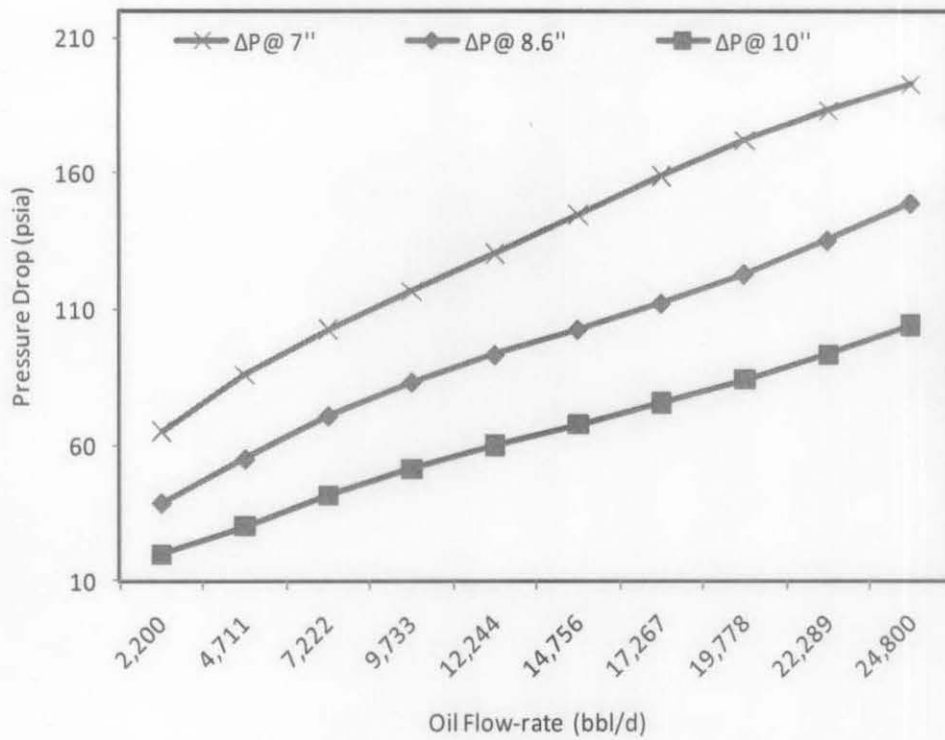


Fig 4.15: Effect of Changing Oil Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.

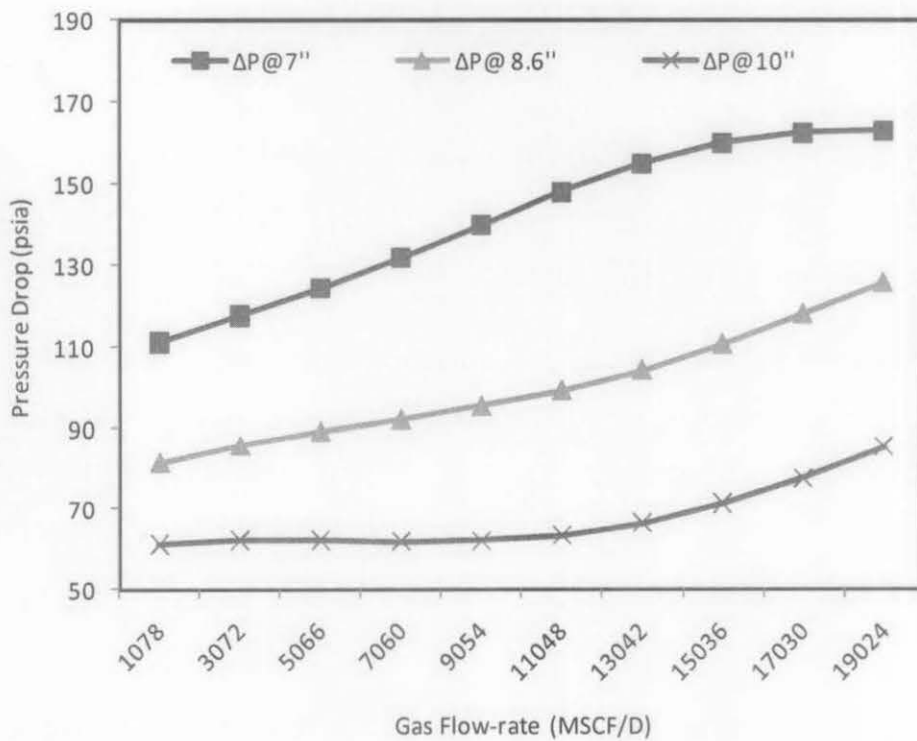


Fig 4.16: Effect of Changing Gas Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.

Additionally, Fig 4.17 draws the pressure drop versus water flow rate at 7, 8.6, and 10 inches pipe diameters. The same analysis can be followed where ANN model was capable in producing the correct trend. At a specific water flow rate the pressure drop was found to be decreasing function with increasing pipe diameter.

Furthermore, the effect of varying pipe lengths on pressure drop at different pipe sizes was clearly investigated as shown in Fig 4.18. As expected, the ANN model succeeded in producing the right trend where smaller pipes were exerting higher pressure drop compared to the biggest ones at a mean angle of 44.6 degrees.

4.3 Group Error Analysis for the Proposed ANN Model against Other Investigated Models

To demonstrate the robustness of the developed model, group error analysis was conducted. Average absolute relative error is a good indicator of the accuracy of all empirical correlations, mechanistic model; as well as the new developed model.

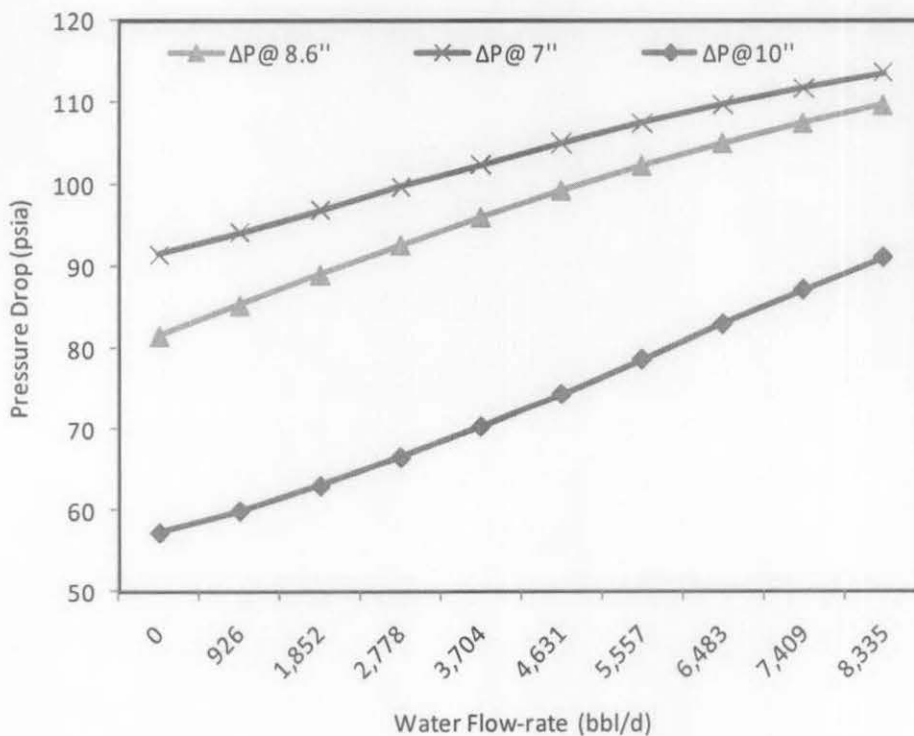


Fig 4.17: Effect of Changing Water Flow-rate for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.

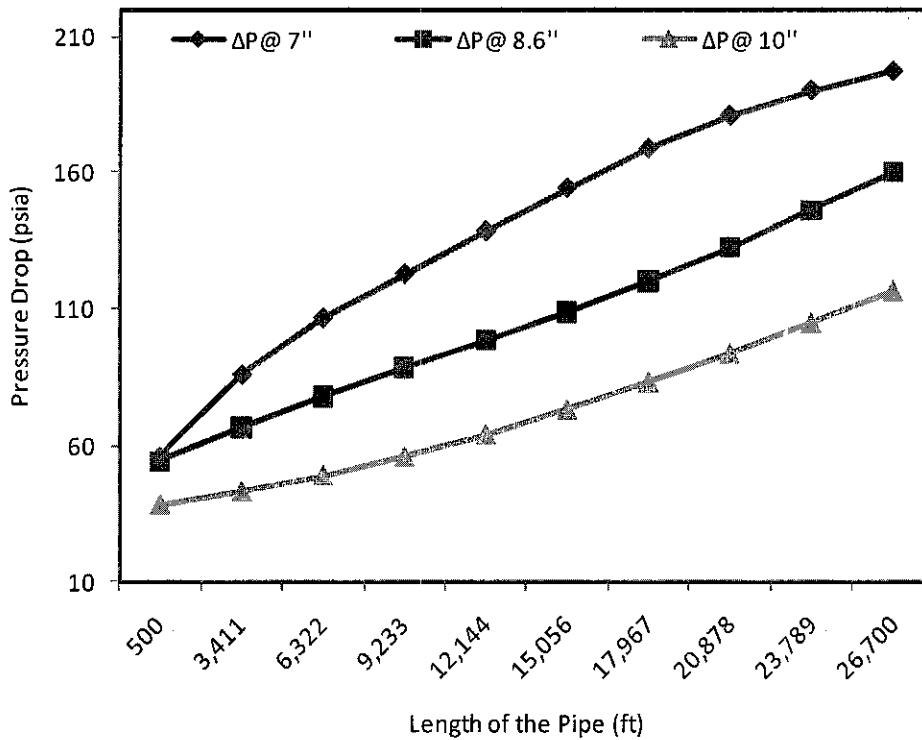


Fig 4.18: Effect of Changing Pipe Length for Three Different Pipe Sizes at a Mean Angle of 44.6 Degrees.

This effective comparison of all investigated correlations and mechanistic models provides a good means of evaluating models performance. Average absolute relative error was utilized in this analysis by grouping input parameter and plotting the corresponding values of average absolute relative error for each set.

Fig 4.19 through Fig 4.24 present the statistical accuracy of pressure drop correlations and models under different groups. Fig 4.19 shows the statistical accuracy of pressure drop grouped by oil rate. The ANN model outperforms the best available correlations and mechanistic models by providing the lowest average absolute relative error for the range of investigated data. Beggs and Brill model outperforms the proposed ANN in the range of oil flow-rate greater than 8001 barrels per day and less than 12600 barrels per day. The reason for that can be attributed to the lower number of cases trained for this range (only eight cases). As shown in Fig 4.20, again ANN model provided the best accuracy when the average absolute relative error plotted against different gas flow-rate groups.

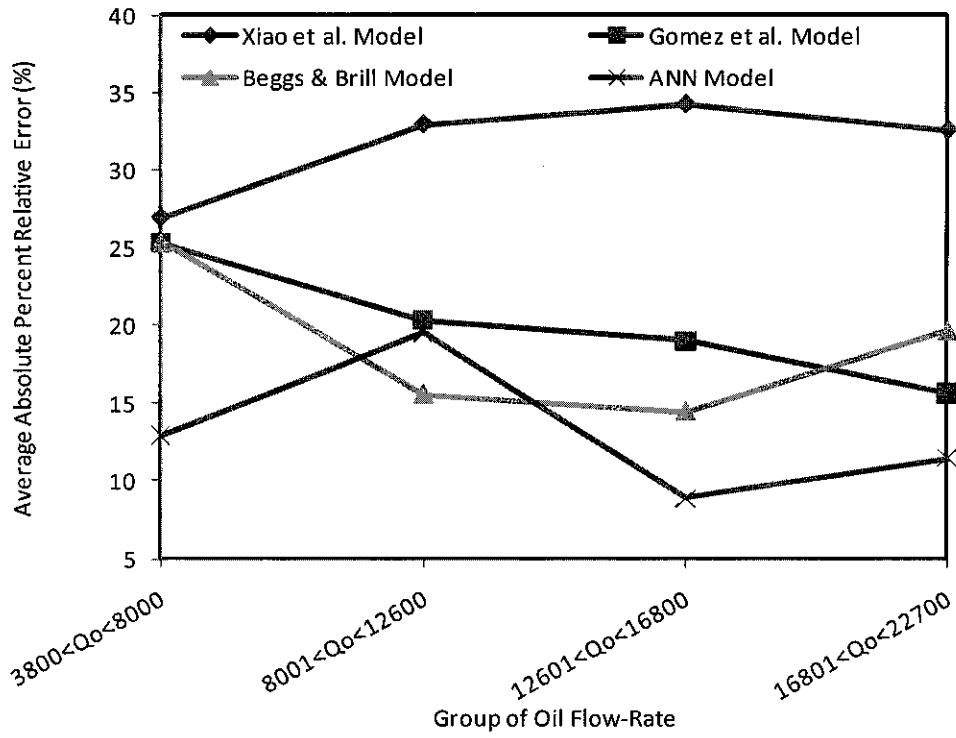


Fig 4.19: Statistical Accuracy of Pressure Drop Grouped by Oil Rate (With Corresponding Data Points).

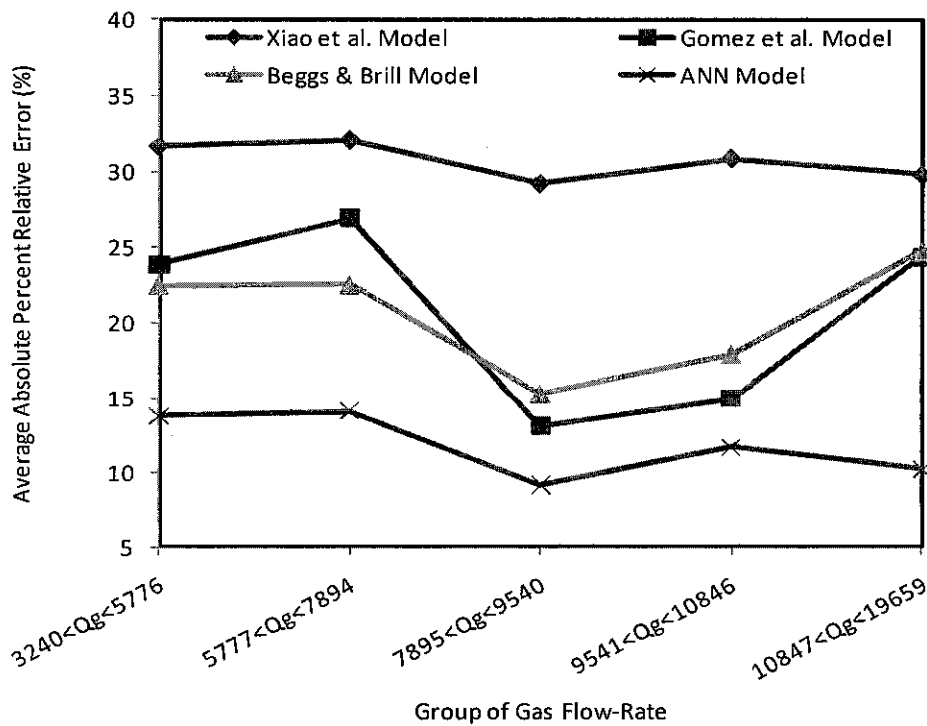


Fig 4.20: Statistical Accuracy of Pressure Drop Grouped by Gas Rate (With corresponding Data Points).

Fig 4.21 shows the statistical accuracy of predicted pressure drop grouped by water rate. The ANN model outperformed other tested methods especially when the system had no water flow-rate and for water flow-rate less than 600 barrels per day. The ANN model showed better results than other tested models when pipe diameter of 6.065 inches was selected, as shown in Fig 4.22. Additionally, the statistical accuracy of pressure drop was also grouped by the pipe length as shown in Fig 4.23. The model also provided the lowest average absolute relative error compared to other tested models. Slight improvement of Beggs and Brill Model for the length interval between 8201 ft and 11900 over the proposed ANN model was witnessed. Fig 4.24 is showing the pressure drop plotted against different ranges of angles of inclination. The proposed artificial neural network was achieving the lowest average absolute percent relative errors (in the range of less than 14%) while Xiao et al. Model was considered the worst with AAPE exceeds 50%.

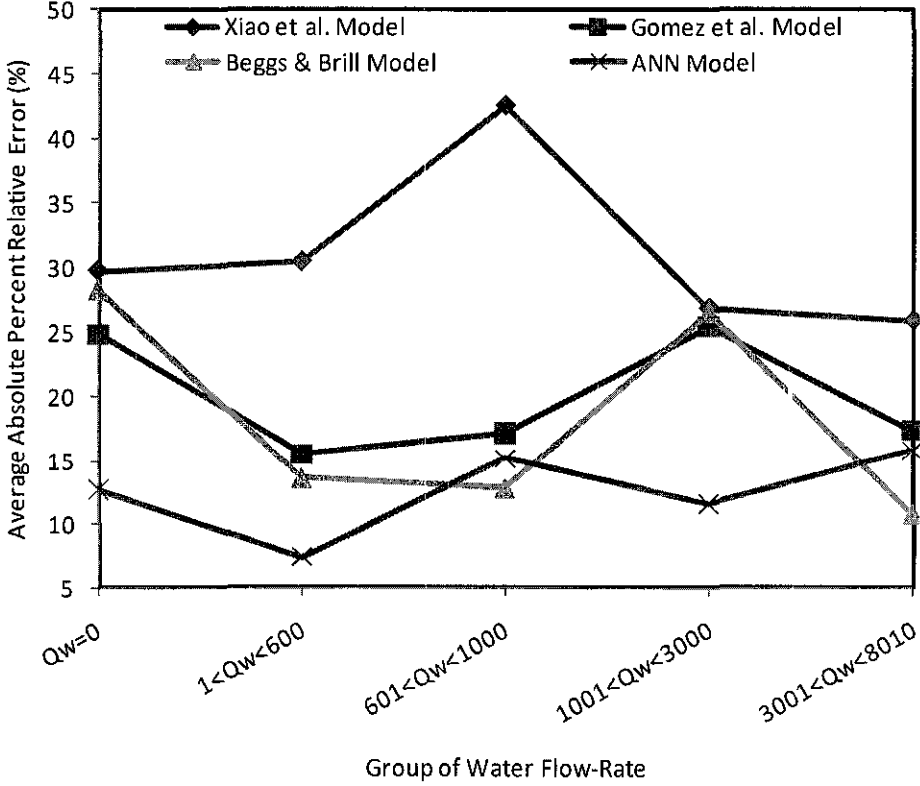


Fig 4.21: Statistical Accuracy of Pressure Drop Grouped by Water Rate (With Corresponding Data Points).

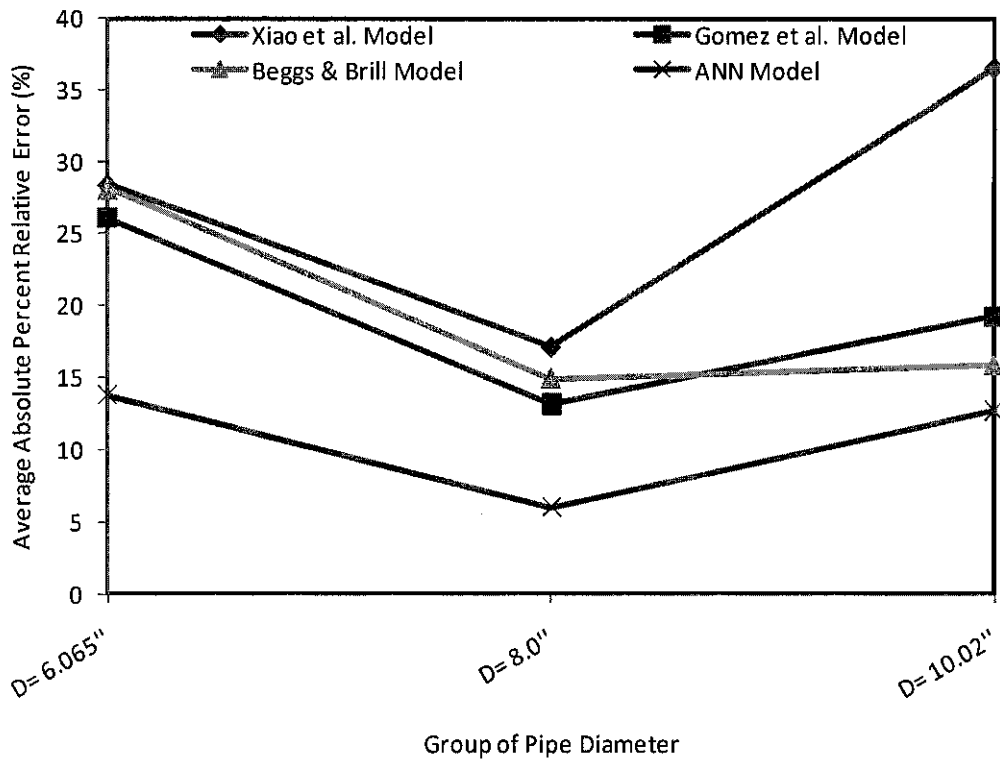


Fig 4.22: Statistical Accuracy of Pressure Drop Grouped by Pipe Diameter (With Corresponding Data Points).

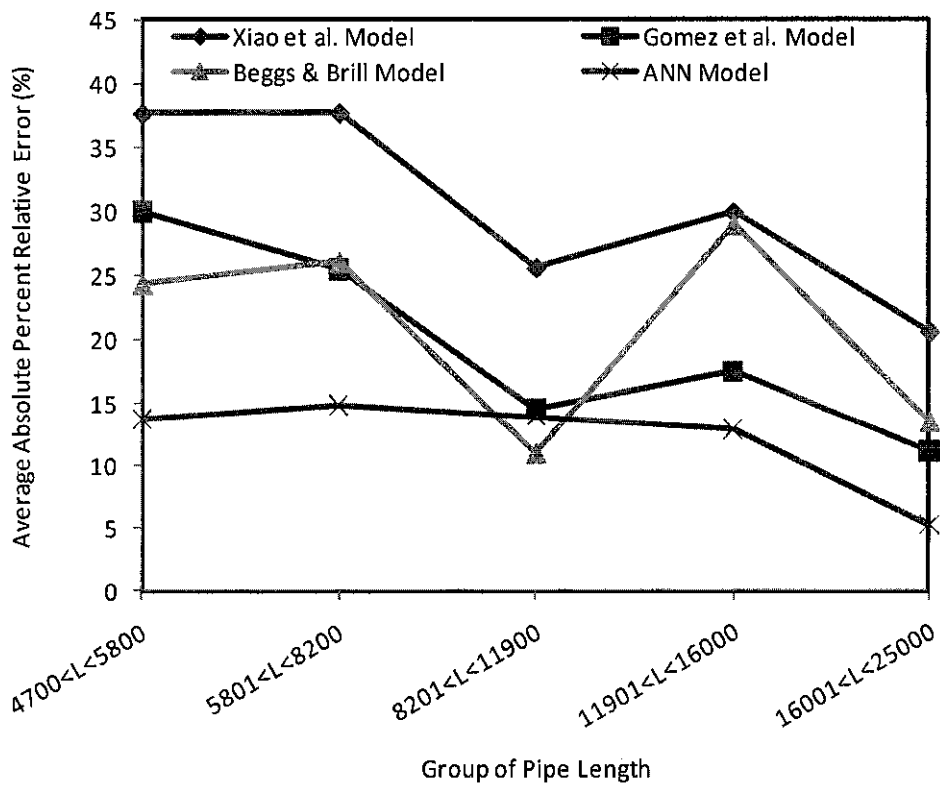


Fig 4.23: Statistical Accuracy of Pressure Drop Grouped by Pipe Length (With Corresponding Data Points).

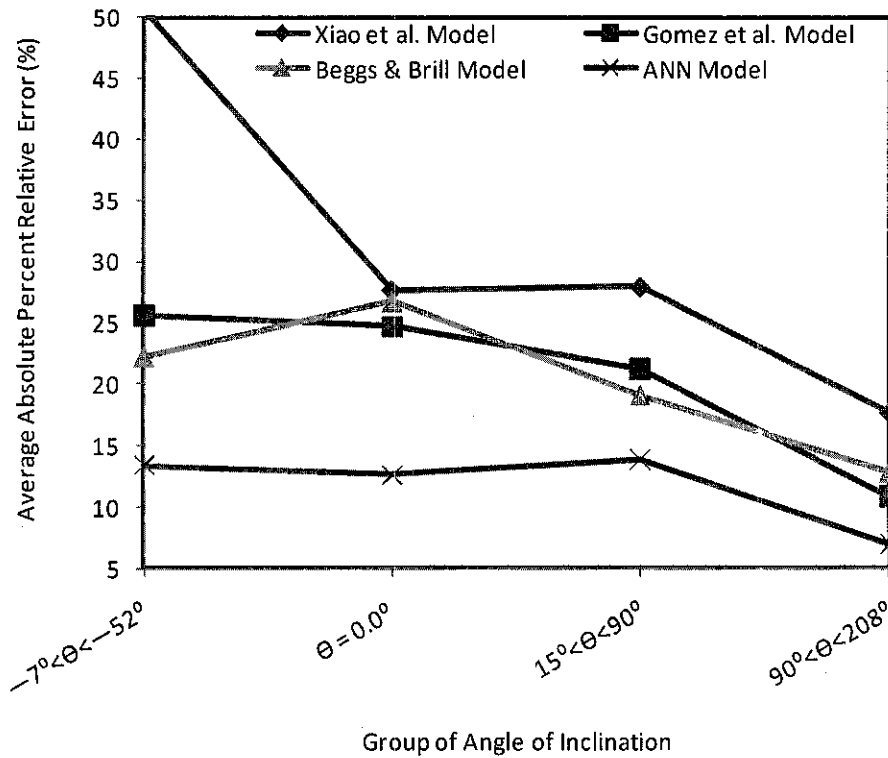


Fig 4.24: Statistical Accuracy of Pressure Drop Grouped by Angle of Inclination (With Corresponding Data Points).

4.4 Statistical and Graphical Comparisons of the Proposed ANN Model against Other Investigated Models

4.4.1 Statistical Error Analysis

As mentioned in methodology chapter (Section 3.6.1), this error analysis was utilized to check the accuracy of all investigated models. The statistical parameters used in the present work are: average percent relative error, average absolute percent relative error, minimum and maximum absolute percent error, root mean square error, standard deviation of error, and the correlation coefficient. Summary of statistical comparisons between all model's sets (training, validation, and testing) is presented in Table 4-4. Robust performance was obtained by the testing set. The main evaluation criterion of the model is AAPE. The ANN model achieved the lowest value among all presented data sets.

Table 4-4: Statistical Analysis Results of the Proposed ANN Model.

Statistical Parameter \ Set Name	Training	Validation	Testing
E_a (Average Absolute Percent Relative Error)	12.3788	17.50147	12.1078
E_r (Average Percent Relative Error)	-4.14	-6.997	1.609
E_{Max} (Maximum Absolute Percent Relative Error)	96.66	165.312	43.996
E_{Min} (Minimum Absolute Percent Relative Error)	0.1657	0.1074	0.2645
RMSE (Root Mean Square Error)	19.504	32.915	15.795
R "fraction" (Correlation Coefficient)	0.9889	0.9670	0.9882
STD (Standard Deviation)	8.398	11.780	10.0158

4.4.2 Graphical Error Analysis of the Proposed ANN Model against Other Investigated Models

Three graphical analysis techniques were employed to visualize the performance of the proposed ANN model and other investigated models; those were cross-plots, error distribution, and residual analysis.

4.4.2.1 Cross-plots of the Proposed ANN Model against Other Investigated Models

Fig 4.25 through Fig 4.30 present cross-plots of predicted pressure drop versus the actual one for the proposed ANN model, and other tested models. Investigation of these figures clearly showed that the proposed ANN model outperformed Beggs and Brill correlation and other two mechanistic models. Fig 4.28 shows the cross-plot between estimated pressure drop values and the actual ones for the training set. As seen from this figure, the ANN reported success in capturing the real relationship between input variables and the output target where higher correlation coefficient was attained by the training set that reached (0.9889). While Fig 4.29 illustrates a cross-plot for the pressure drop values for the validation set. A lower correlation coefficient

(0.967) was attained by the model and this can be attributed to the effect of early stopping technique, which was adopted during training the model.

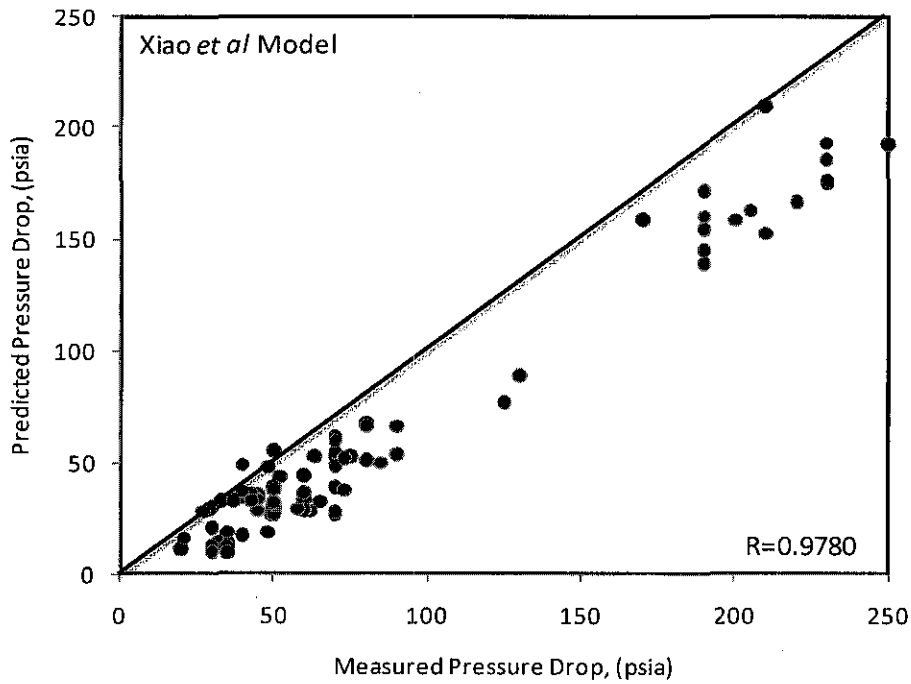


Fig 4.25: Cross-plot of Predicted vs. Measured Pressure Drop for Xiao et al. Model.

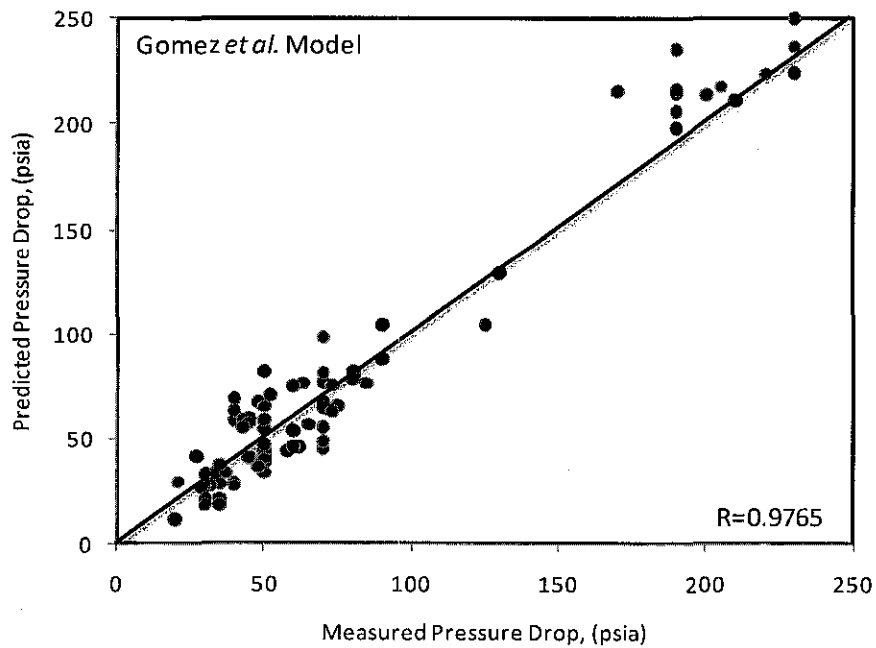


Fig 4.26: Cross-plot of Predicted vs. Measured Pressure Drop for Gomez et al. Model.

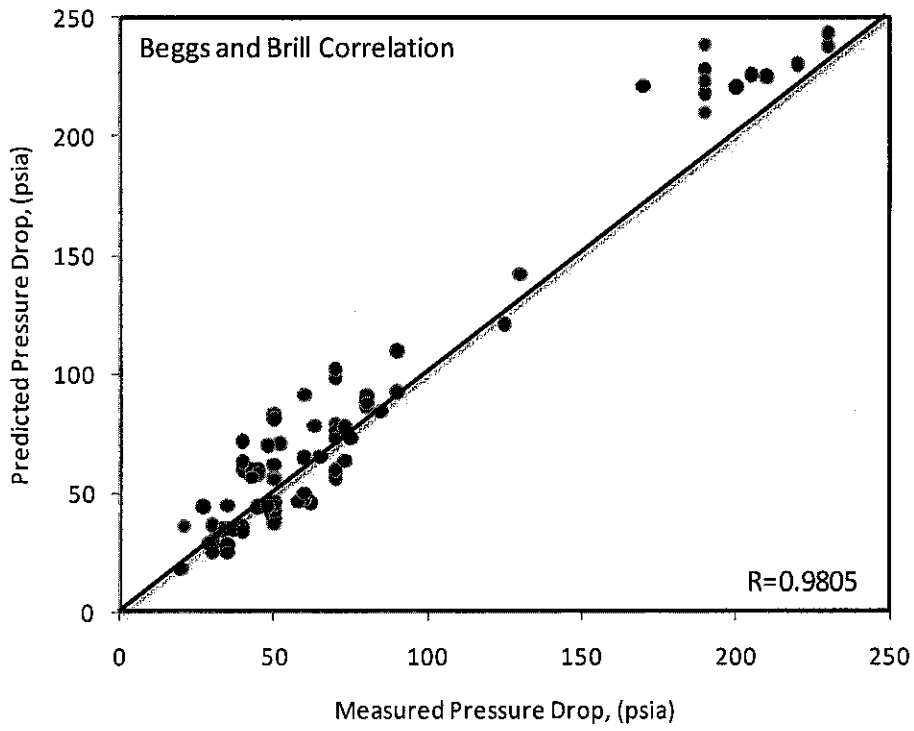


Fig 4.27: Cross-plot of Predicted vs. Measured Pressure Drop for Beggs and Brill Model.

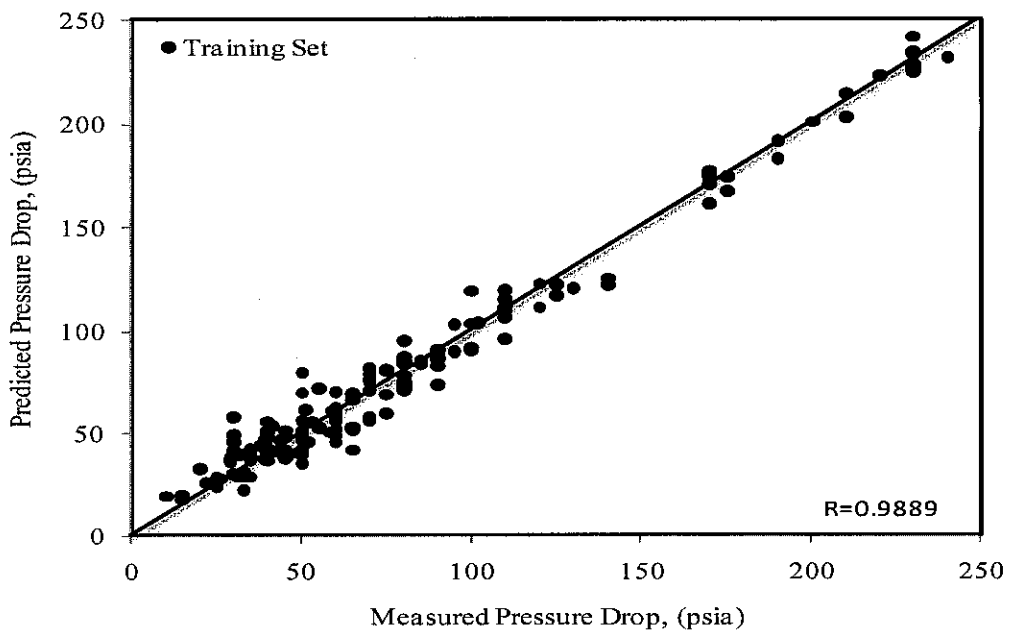


Fig 4.28: Cross-plot of Predicted vs. Measured Pressure Drop for Training Set (Proposed ANN Model).

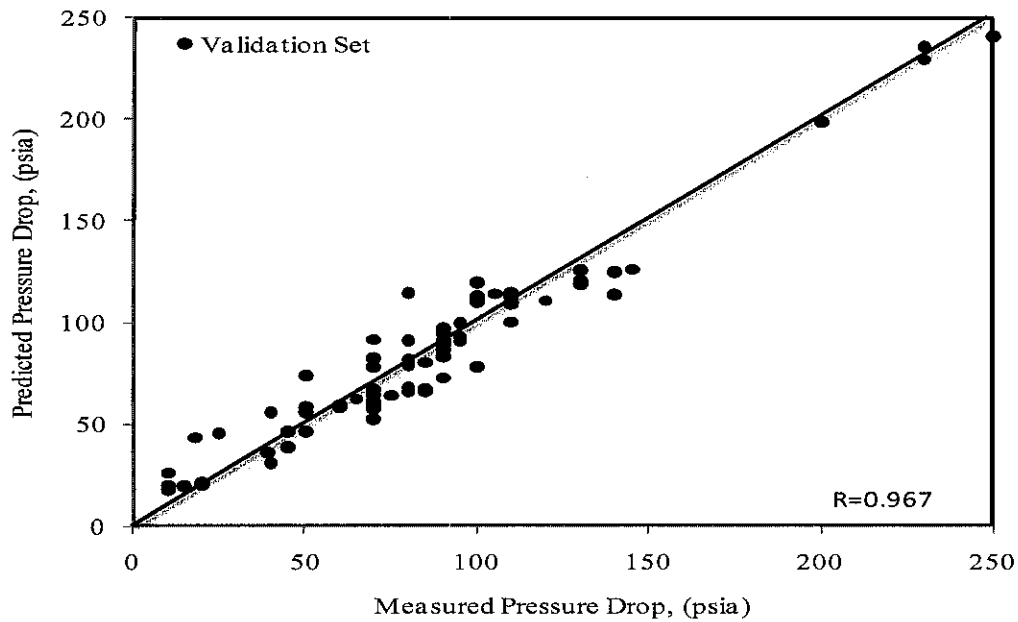


Fig 4.29: Cross-plot of Predicted vs. Measured Pressure Drop for Validation Set (Proposed ANN Model)

Moreover, Fig 4.30 depicts the cross-plot between predicted pressure drops against its real values. The new proposed ANN model achieved the highest correlation coefficient (0.9882) among all models which indicates its superiority.

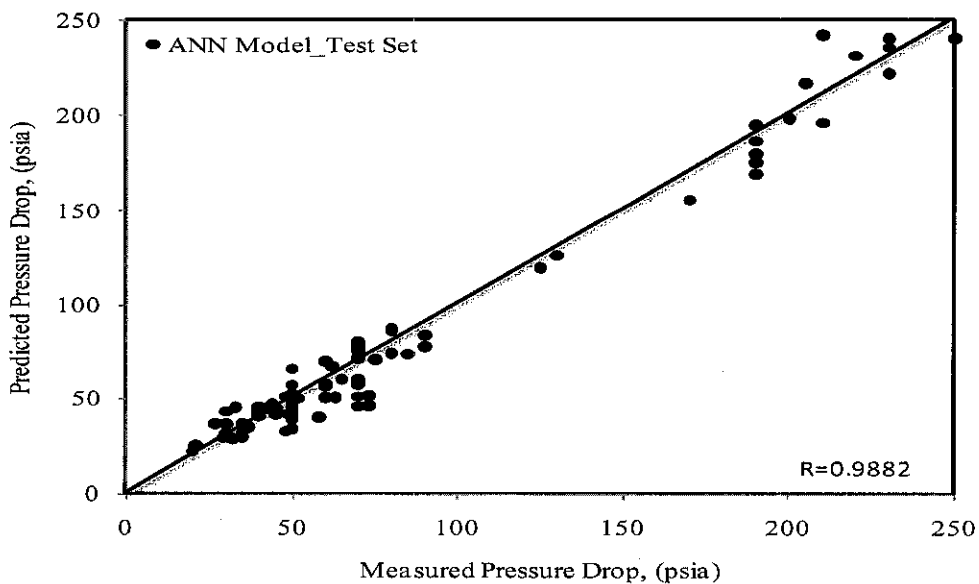


Fig 4.30: Cross-plot of Predicted vs. Measured Pressure Drop for Testing Set (Proposed ANN Model)

Graphical comparison between models was given in Fig 4.31 and Fig 4.32, which showed the correlation coefficients and root mean squared errors of all models. The ANN model achieved the highest correlation coefficient (0.98821), while other correlations indicated higher scattering range compared to the proposed ANN model, where 0.9805 was obtained by Beggs and Brill model; 0.9765 for Gomez et al. model; and 0.9780 for Xiao et al. model. Beggs and Brill correlation achieved the highest correlation coefficient among all other mechanistic models. However, Beggs and Brill model was found to overestimate the pressure drop in the tested range, as presented in Fig 4.27. This finding had coincided with the past conclusion of Hong and Zhou, [Hong and Zhou, 2008]. However, Xiao et al. model tended to underestimate the pressure drop for most of the tested cases as shown in Fig 4.25. In addition, Gomez et al. model had been found to overestimate the pressure drop especially at high pressure drop values as clearly shown in Fig 4.26.

Comparison between the performance of all investigated models plus the new proposed ANN model is provided in Table 4-5. Gomez et al. model achieved the worst correlation coefficient among all investigated models. As seen from the previously described Figures, the margin between the correlation coefficients is insignificant. However, the correlation coefficient serves as a supporting evaluation criterion in which the AAPE is the main evaluation criterion of all models.

As illustrated in Table 4-5, the proposed ANN model achieved the lowest Average Absolute Percent Relative Error (E_a), compared to other tested models (12.11%) while Beggs & Brill model ranked the best among the three tested models with AAPE reached 20.08%. The average absolute percent relative error is a significant sign of the accuracy of the models. Gomez et al. model performed the second best among tested models with AAPE reached 20.85% while Xiao et al. model performed the worst with AAPE of 30.85%. As noticed from the previous discussion that the new proposed ANN model outperformed all investigated models in terms of lower maximum error obtained by the testing set that reached (44%) while other investigated models gave maximum error ranges between (71% to 79%), as shown in Table 4-5. However, model generalization had suffered due to low range of some input parameters and redundancy in others.

To overcome this obstacle, early stopping technique had been adopted during training of the model. It ensures the optimum generality of the trained model. This approach relies on ceasing the training when a signal indicates that prediction's capability of the trained model starts to deteriorate. While training the model with a certain set of training data and at a specific point of time validation and testing sets will be presented to check for the prediction accuracy. Early stopping had been applied when the gap between error curves started to become large. This is can be considered as some sort of checking model's generality when new cases are presented to it after fixing weights.

The developed model also achieved the second lowest minimum error for the range of tested data with approximate values of 0.2645%, directly after Xiao et al model. Root Mean Square Error (RMSE) was used to measure the data dispersion around zero deviation. Again, the proposed ANN model (testing set) attained the lowest RMSE of 15.8% compared to the Beggs & Brill and Gomez et al. models with 26.8% and 26.03%, respectively. Standard Deviation (STD) was used as another confirming feature of model superiority. This statistical feature was utilized to measure the data dispersion. A lower value of standard deviation indicates a smaller degree of scatter. The proposed ANN model obtained the lowest STD of errors (10.02), while Xiao et al. model achieved the lowest STD among other investigated models with a value of 15.7278.

Table 4-5: Statistical Analysis Results of Empirical Correlations, Mechanistic Models, and the Proposed ANN Model.

Statistical Feature	E_a	E_r	E_{Max}	E_{Min}	RMSE	R	STD
Model Name							
Beggs and Brill model (1991)	20.0762	-10.987	79.00	0.3333	26.7578	0.9805	16.9538
Gomez et al. model (1999)	20.802	-2.046	72.65	0.525	26.0388	0.9765	17.7097
Xiao et al. model (1990)	30.845	29.818	71.4286	0.0625	35.4582	0.9780	15.7278
Proposed ANN Model	12.11	1.6087	43.99	0.2644	15.795	0.98821	10.016

Comparison between average absolute percent relative error for all tested models and the new proposed model is provided in Fig 4.33.

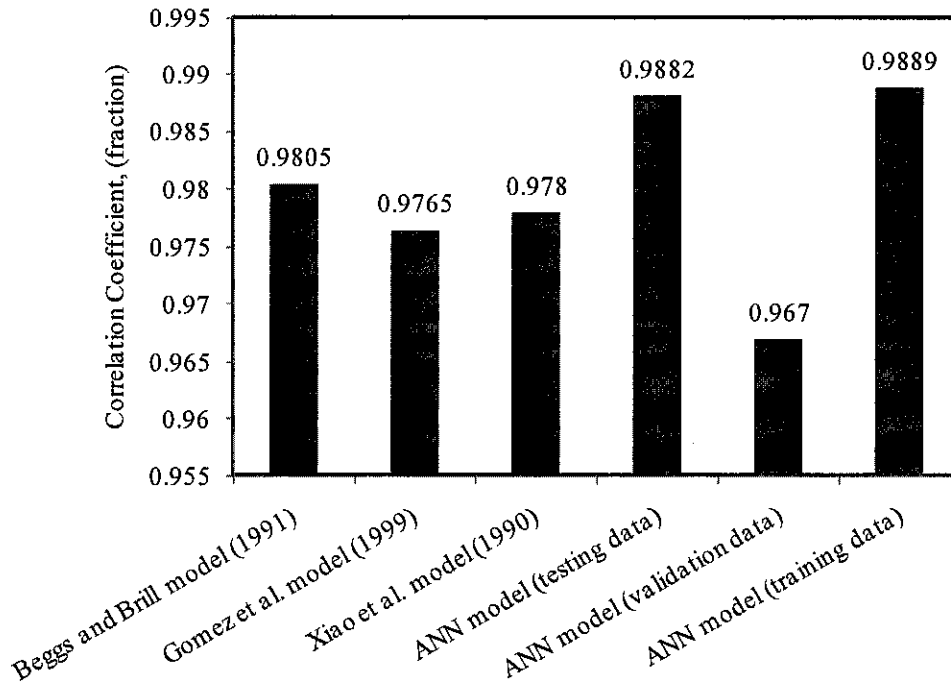


Fig 4.31: Comparison of Correlation Coefficients for the Proposed ANN Model against other Investigated Models.

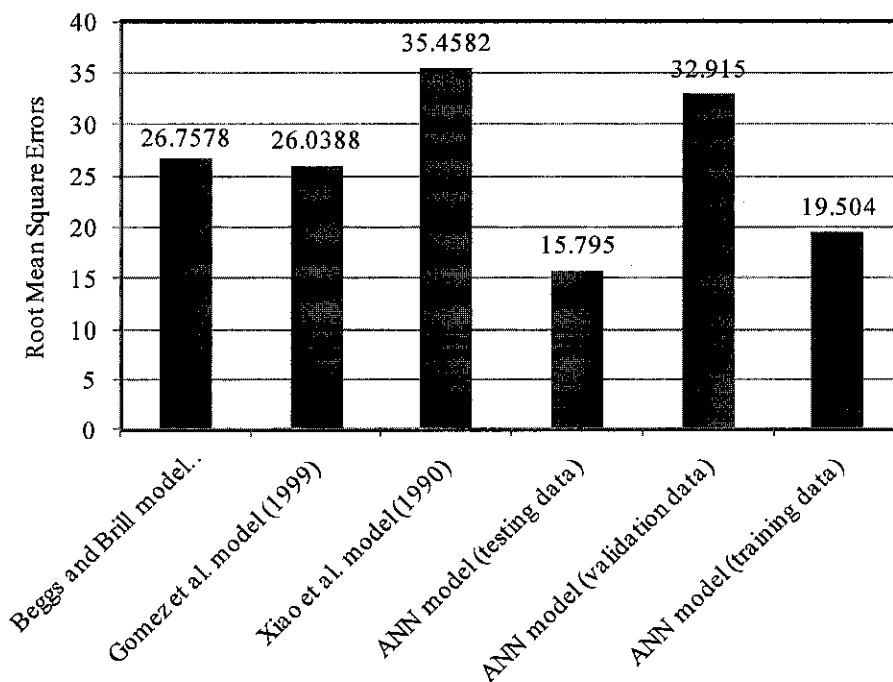


Fig 4.32: Comparison of Root Mean Square Errors for the Proposed ANN Model against other Investigated Models.

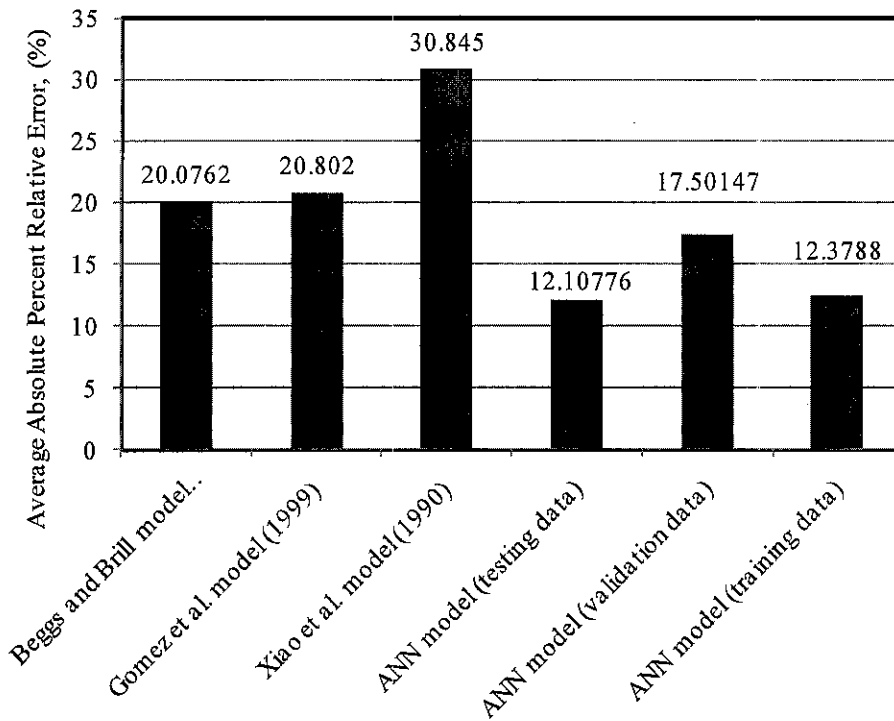


Fig 4.33: Comparison of Average Absolute Percent Relative Errors for the Proposed ANN Model against other Investigated Models.

4.4.2.2 Error Distributions of the Proposed ANN Model against Other Investigated Models

Fig 4.34, Fig 4.35 and Fig 4.36 show the error distribution histograms for the neural network model, (training, validation, and testing sets). Normal distribution curves were fitted to each one of them. The errors are said to be normally distributed with a mean around the 0% and the standard deviation equal to 1.0.

Analyzing the ANN model's error distribution histogram is quite important for the sake of checking model's performance for all data sets. Fig 4.34 shows the error distribution histogram and the normal distribution curve for the training set of the new proposed model. It shows a slight shift of the mean of the errors towards the negative side of the plot (about 4%) indicating that the pressure drop was slightly overestimated. However, as it is seen from the same figure that almost 65.5% of the total error frequencies had laid within the normal distribution curve as indicated by twice the standard deviation (One standard deviation for each side from the mean).

The optimum statistical ratio should be 2/3 of errors lay within the normal distribution curve. It is evident how close this value to the theoretical value (67%).

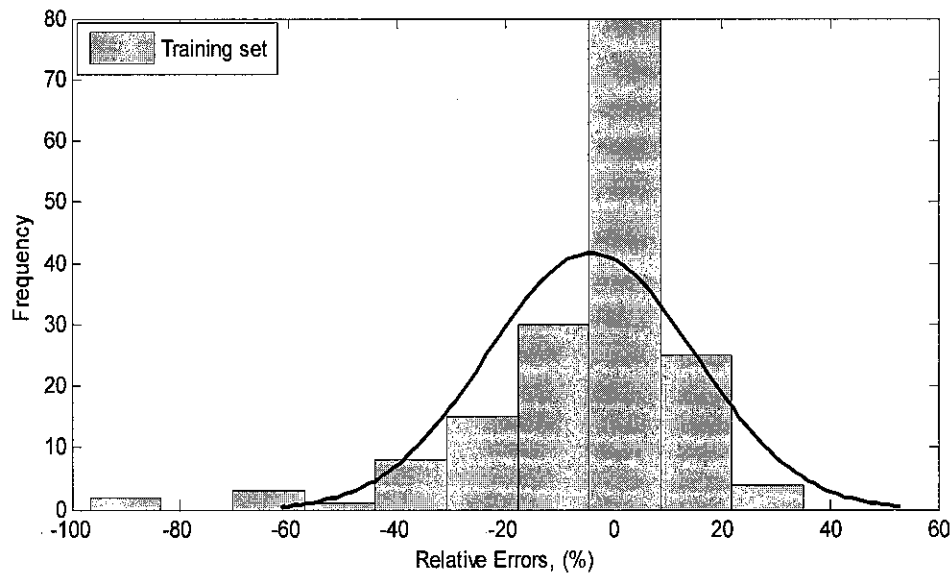


Fig 4.34: Error Distribution for Training Set (Proposed ANN Model).

Fig 4.35 depicts the error distribution histogram and the normal distribution curve for the validation set of the new proposed model. It showed a slight skewing of the mean of the errors towards the negative side of the plot (about 7%) indicating that the pressure drop was overestimated. Following the same approach, it was seen that 46.4% of the total error frequencies had been presented by the shifted normal distribution curve.

Furthermore, Fig 4.36 illustrates the error distribution histogram and the normal distribution curve for the testing set of the new proposed model. The mean of the errors was skewed by 1.6% to the right, which indicates good representation of errors by the normal distribution curve. It indicates that the new proposed ANN model underestimated the pressure drop for the tested region with very minor degree. Almost 61.4% of the total error frequencies lay within the normal distribution curve. This analysis can be adopted for the other investigated models as well.

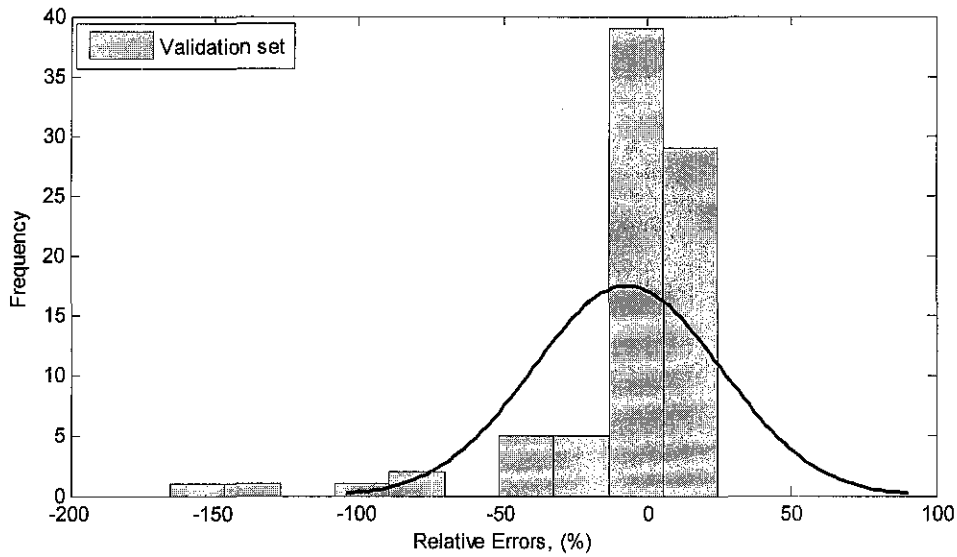


Fig 4.35: Error Distribution for Validation Set (Proposed ANN Model).

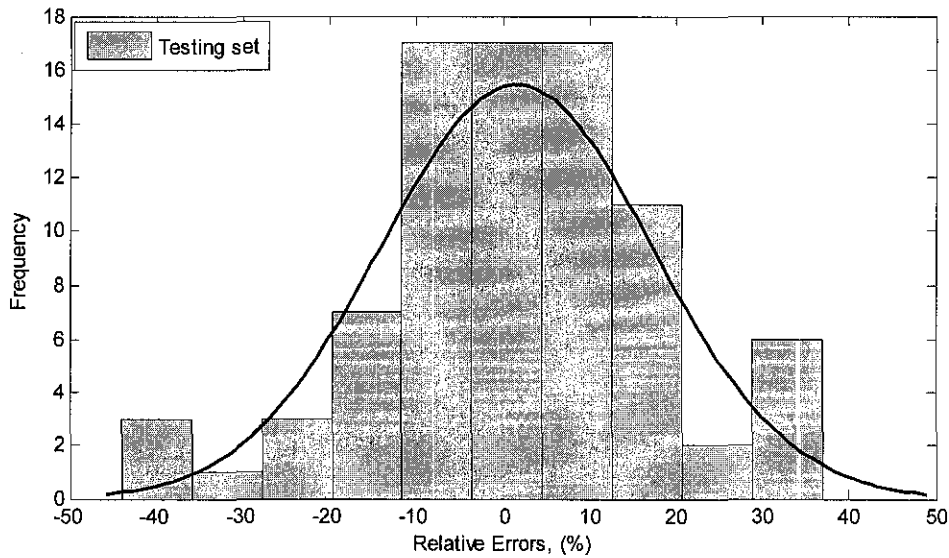


Fig 4.36: Error Distribution for Testing Set (Proposed ANN Model).

Fig 4.37 shows the error distribution histogram and the normal distribution curve for Gomez *et al.* model. It demonstrated a slight shift of the mean of the errors towards the negative side of the plot (about 2%) indicating that the pressure drop was

slightly overestimated. On the other hand, 43 error cases out of 84 tested cases lay within the shifted normal distribution curve.

Fig 4.38 illustrates the error distribution histogram and the normal distribution curve for Beggs & Brill correlation. It demonstrated a great shift of the mean of the errors towards the negative side of the plot (about 11%) indicating that the pressure drop was highly overestimated as confirmed by the cross-plot in Fig 4.27. On the other hand, 40 error cases out of 84 tested cases lay within the shifted normal distribution curve. Xiao *et al.* model's error distribution histogram and the normal distribution curve are presented in Fig 4.39. It is evident that Xiao *et al.* model underestimated the pressure due to the high shift of the normal distribution curve to the right side (29.8%). Xiao *et al.* model showed the worst error distribution curve among all tested models, where it shifted around 29.8% towards the right side indicating underestimation of pressure drop, as illustrated in Fig 4.39, which indicates the inadequacy of the model for predicting pressure drop under the tested range of variables.

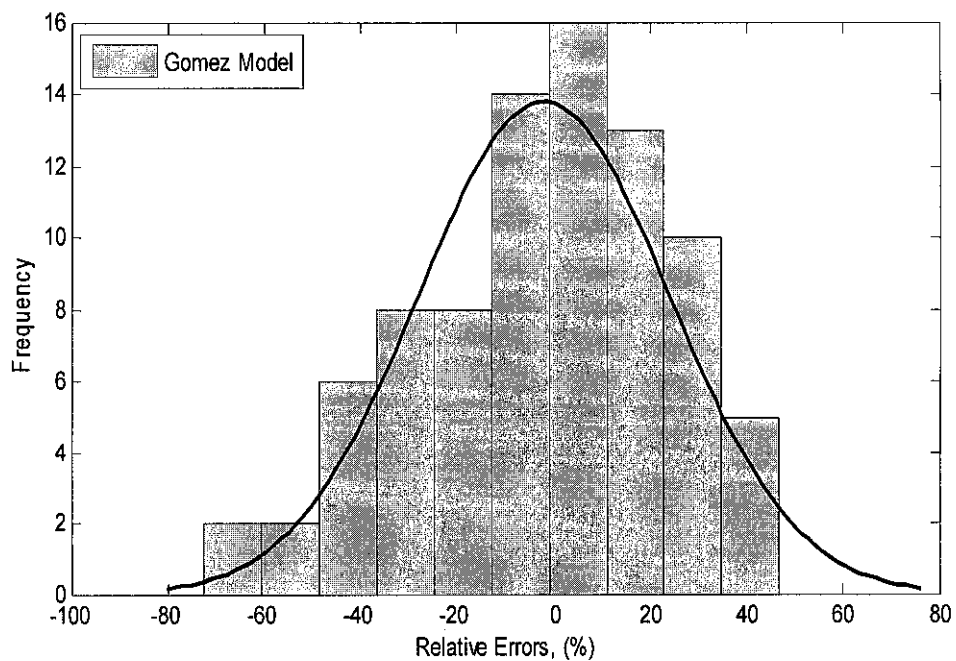


Fig 4.37: Error Distribution for Gomez et al. Model

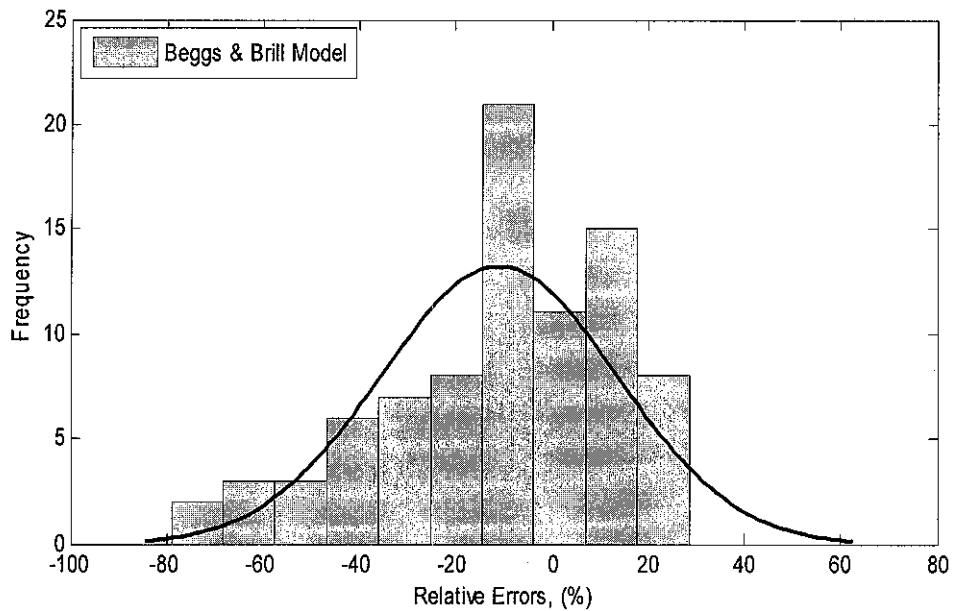


Fig 4.38: Error Distribution for Beggs and Brill Correlation.

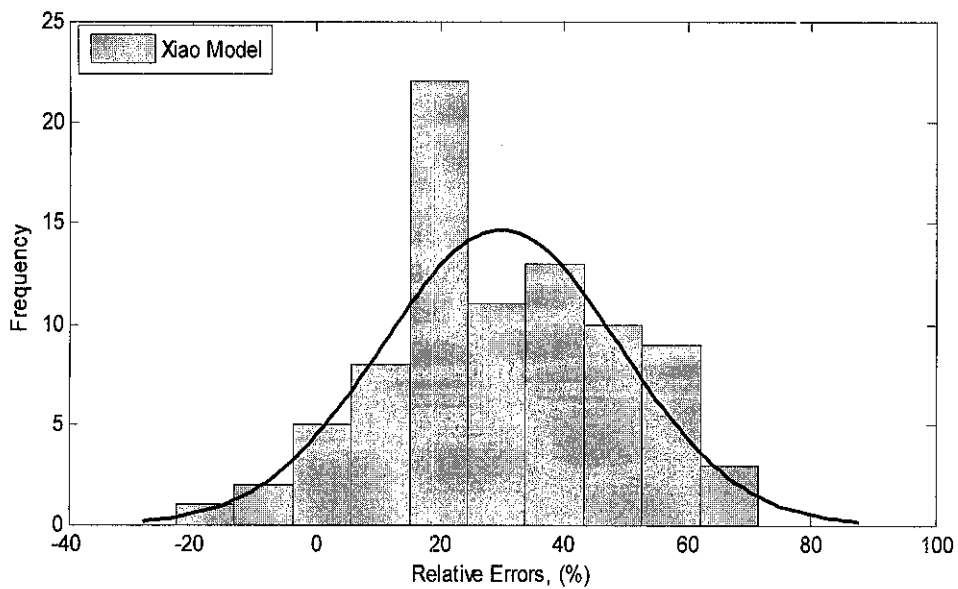


Fig 4.39: Error Distribution for Xiao et al. Model

The range of errors also is an important parameter for detecting the accuracy of each model. This range can be extracted from each histogram figure (from Fig 4.36 through Fig 4.39). A range of -25% to 85% was used for Xiao et al. model as a best model if this feature is considered, whereas an error range of -45% to 50% in pressure

drop was achieved for testing set. This indicates the superiority of the new proposed model over other investigated models.

However, all tested models demonstrated moderate predictability of pressure drop in pipelines with errors normally distributed with a negative or positive mean.

The new proposed model did not suffer from memorizing the pressure drop values as it shows satisfactory degree of consistency when compared to the validation results. The latter had been used as a safeguard against the memorization. If the correlation coefficient is used as a main criterion for selecting the best overall performance, Beggs and Brill correlation could be selected based on this feature. Because standard deviation is one of the measures of scattering tendencies, it is included as a measure of how errors are distributed and scattered. Based on this criterion, Xiao *et al.* model performed the best (15.7) followed by Beggs and Brill correlation (16.95) while Gomez et al model ranked the least accurate with the highest standard deviation of errors of (17.7).

Beggs and Brill correlation showed the lowest average absolute percent error as AAPE and correlation coefficient can be selected as the main criteria for selecting the best model for predicting the pressure drop in pipeline. It was decided to tabulate the values of correlation coefficient and the AAPE for each model in one Table as shown in Table 4-6. For the sake of easing the analysis, the rating of model performance was based on having the lowest average absolute percent relative error and highest correlation coefficient. According to this, the new proposed ANN model showed optimum performance compared to the rest of investigated models. Beggs & Brill model ranked second best followed by Gomez et al. and Xiao et al. models. A close result can be extracted when root mean square errors and the standard deviation of errors of each model had been tabulated in Table 4-7. On the contrary, this time the best model will be judged on having the lowest Root Mean Square of Errors followed by the lowest Standard Deviation of Errors. Again the new proposed ANN model achieved the optimum performance, while the rest of the tested models dropped below it. This indicated superior performance of ANN model compared to other tested models.

Table 4-6: Evaluating Models Performance by Average Absolute Percent Errors and Correlation coefficient

Statistical Feature Correlation / Model	Average Absolute Percent Errors	Correlation Coefficient	Rating
ANN model (testing data)	12.10776	0.9882	1
Beggs and Brill model (1991)	20.0762	0.9805	2
Gomez et al. model (1999)	20.802	0.9765	3
Xiao et al. model (1990)	30.845	0.978	4

Table 4-7: Evaluating Models Performance by Root Mean Square Errors and Standard Deviation of Errors

Statistical Feature Correlation / Model	Root Mean Square Errors	Standard Deviation of Errors	Rating
ANN model (testing data)	15.795	10.0158	1
Beggs and Brill model (1991)	26.0388	17.7097	2
Gomez et al. model (1999)	26.7578	16.9538	3
Xiao et al. model (1990)	35.4582	15.7278	4

4.4.2.3 Residual Analysis Error Distributions of the Proposed ANN Model against Other Investigated Models

As per data partitioning scheme, the test set contains 84 sets of data, which were utilized to perform all statistical and graphical tests. The relative frequency of deviations between estimated and actual values was depicted in Fig 4.40 through Fig 4.45 for the proposed ANN model and other investigated models. These Figures

showed the error distribution around the zero line to verify whether models and correlation contained error trends.

Analysis of residual limits (predicted pressure drop minus the actual pressure drop) is an effective tool to check model deficiencies. Residual limits of investigated model were shown in Table 4-8. Gomez et al. model (refer to Fig 4.43) and Beggs & Brill correlation (refer to Fig 4.44) showed the worst negative error performance with maximum values of -82.61 psia and -79.1 psia, respectively. While Xiao et al. model showed the worst positive error performance (57.91 psia), as appears in Fig 4.45. Additionally,

Fig 4.40, Fig 4.41 and Fig 4.42 showed the residual plots for the new proposed model separately (training, validation, and testing sets). A range of -30 to 23 was reported by the training set as shown in Fig 4.40. Regardless of validation lower performance, the set managed to achieve lower range of residual errors as clearly shown in Fig 4.41. Furthermore, a range between -34 to 26 psia was achieved by the validation set. A maximum value of -32.23 to 26.94 was reported by testing set (refer to Fig 4.42). This is an additional indication that the new proposed model outperformed the investigated models in a sense of error distribution around the zero line.

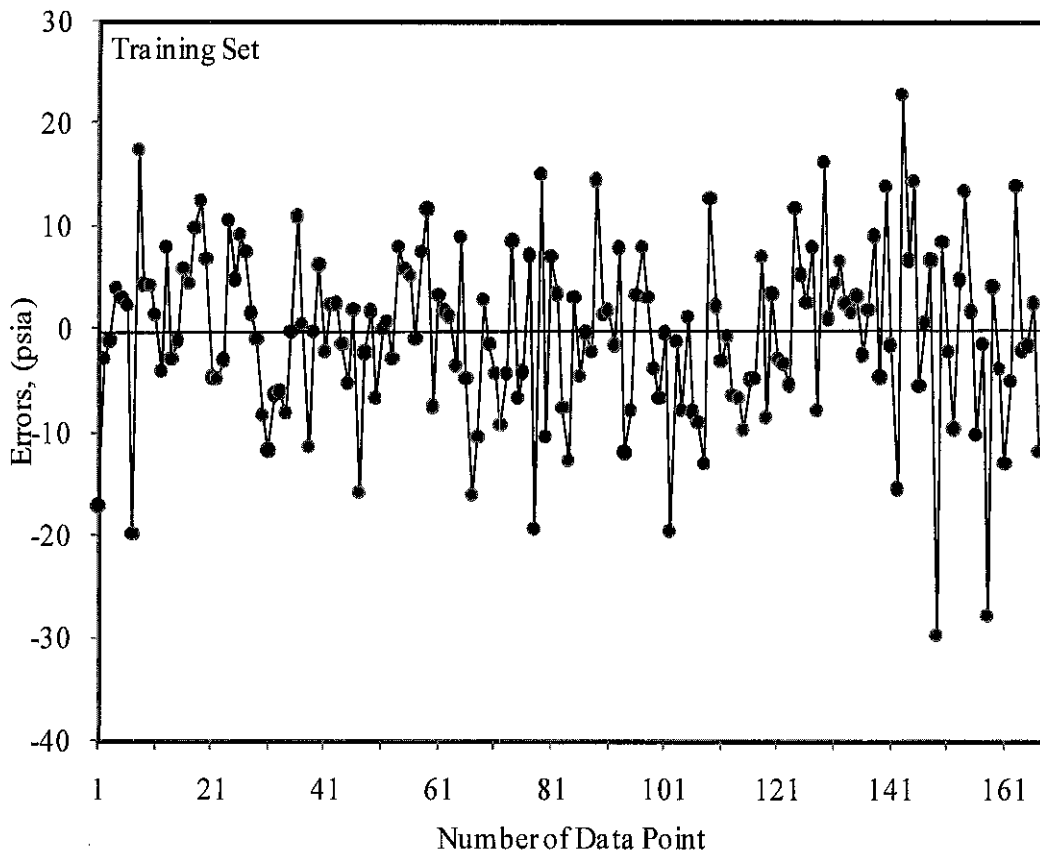


Fig 4.40: Residual Graph for Training Set (Proposed ANN Model).

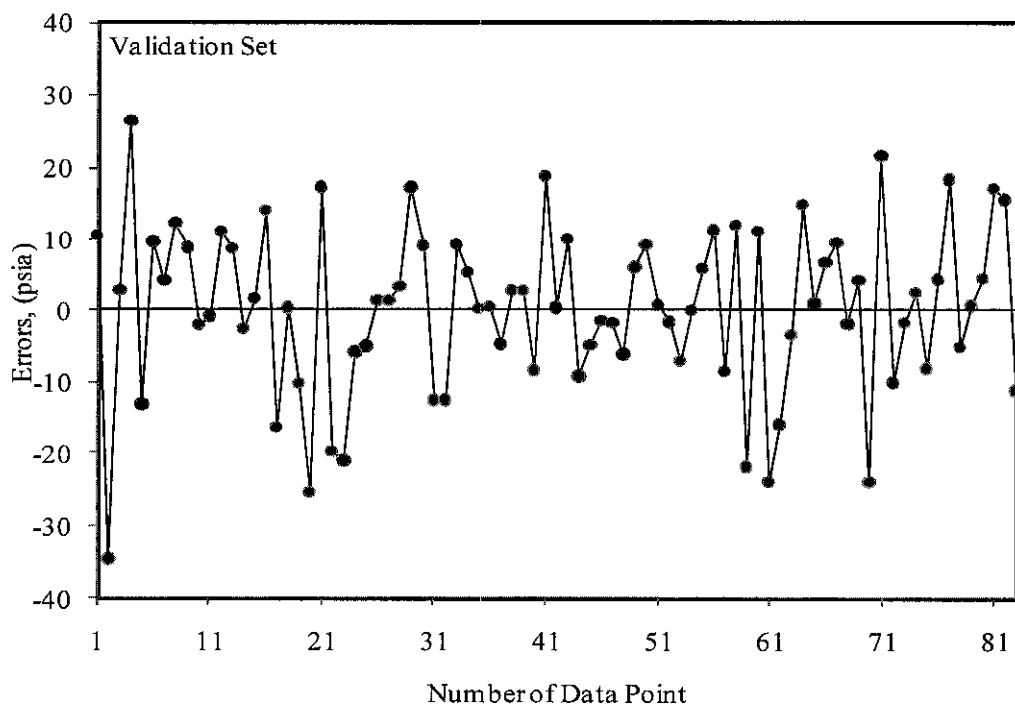


Fig 4.41: Residual Graph for Validation Set (Proposed ANN Model).

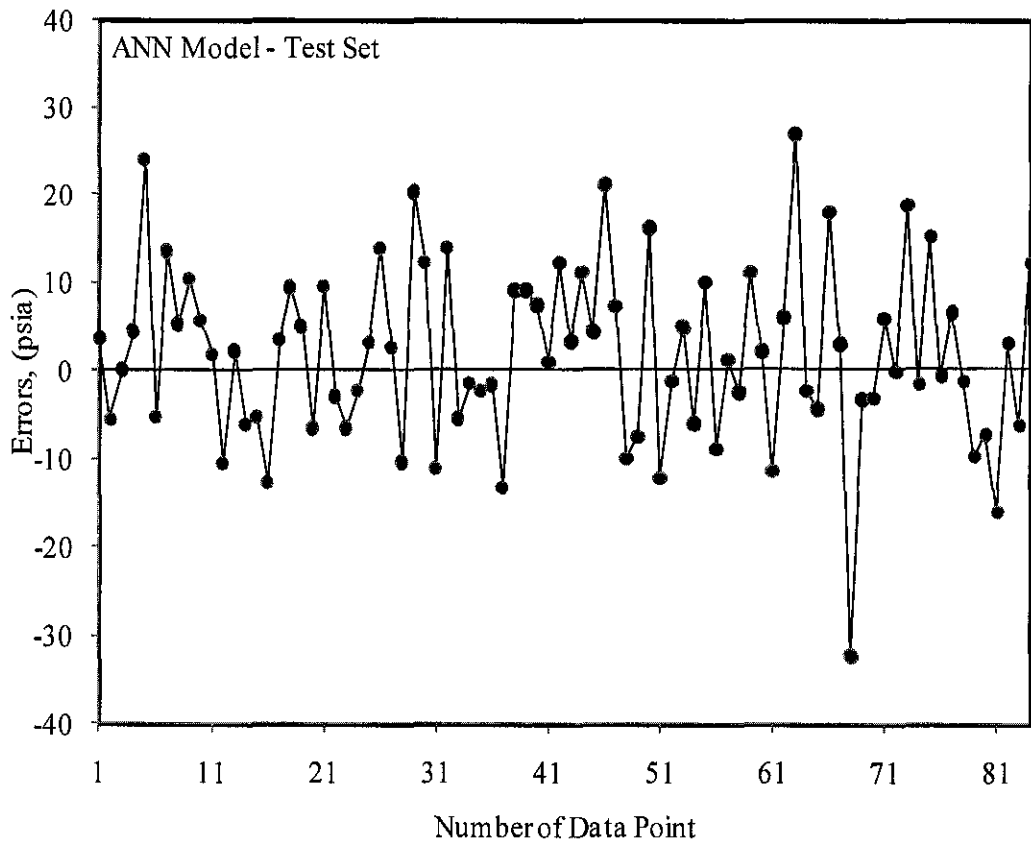


Fig 4.42: Residual Graph for Testing Set (Proposed ANN model).

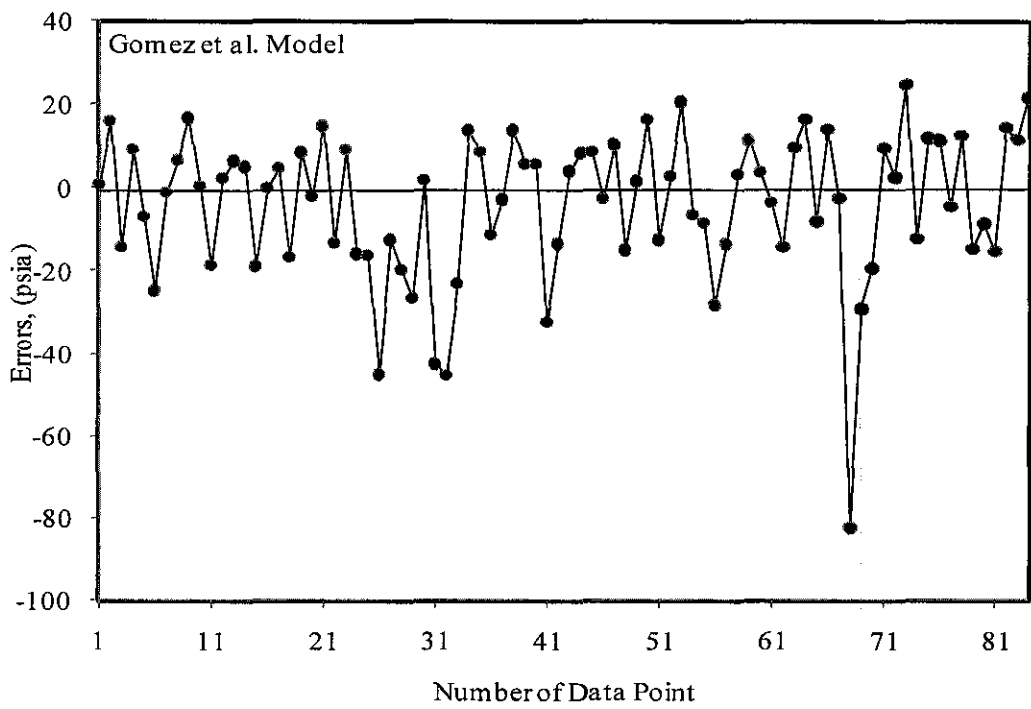


Fig 4.43: Residual Graph for Gomez et al. Model.

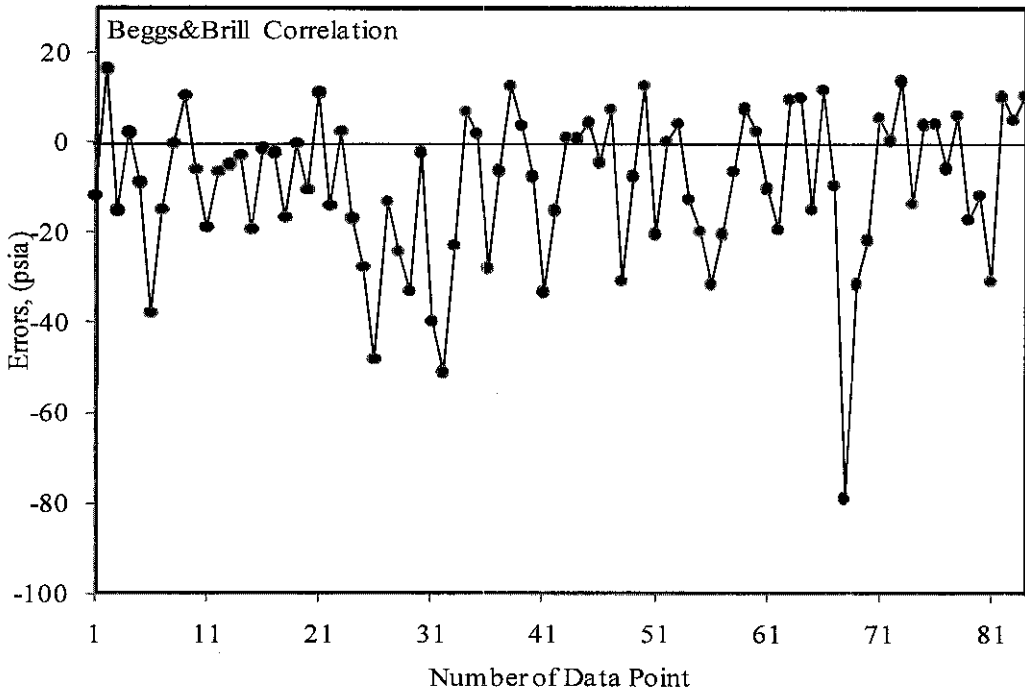


Fig 4.44: Residual Graph for Beggs & Brill Correlation.

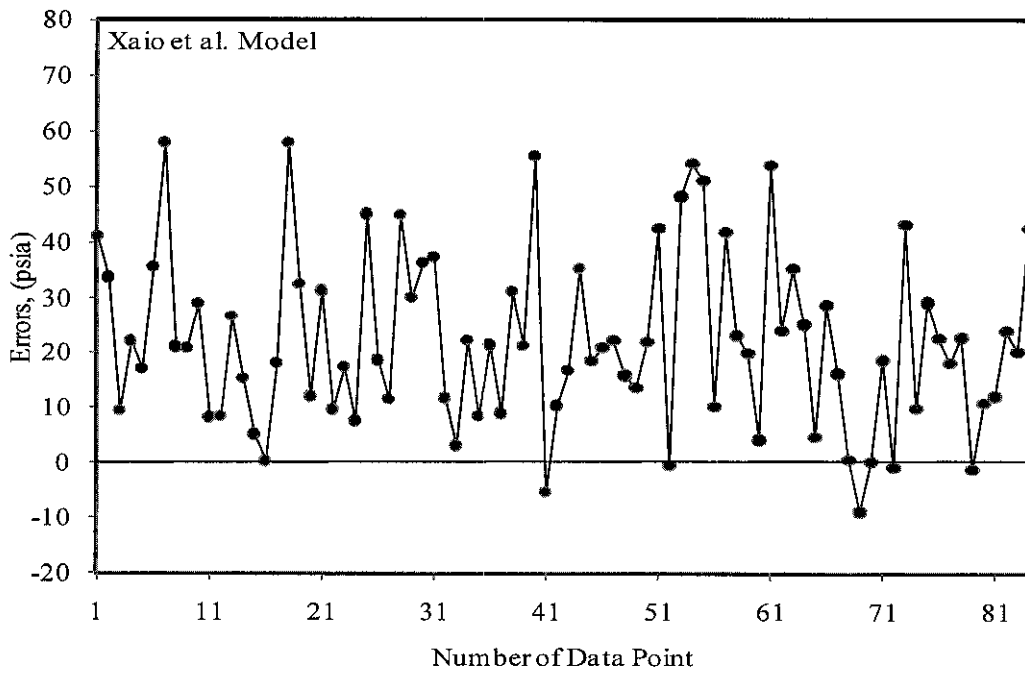


Fig 4.45: Residual Graph for Xiao et al. Model.

Table 4-8: Residual limits of the Proposed ANN Model against the Best Investigated Models.

Statistical Feature Model Name	Maximum	Minimum
ANN Model	26.94	-32.23
Beggs and Brill correlation	16.33	-79.1
Xiao <i>et al.</i> Model	57.91	-9.02
Gomez <i>et al.</i> Model	24.689	-82.61

4.5 Development of AIM Model

4.5.1 Introduction

AIM (Polynomial Group Method of Data Handling technique) is a smart type of regression, which utilizes a series of three steps to reach the final output (representation, selection, and stopping). The technique is capable of producing high degree polynomial in effective predictor. In addition, the process starts with initially simple regression relationship to derive more accurate representation in the next iteration. Polynomial GMDH technique is offering a sound representation of input regime to output through the application of so called “regularity criterion”. Usually this one will be average absolute percentage error. It is implemented to reduce the error between the actual and estimated target in each layer. A threshold level is applied before each layer is added since addition of a new layer and neurons depends on this threshold level.

As described initially in Section 3.10, software was utilized for building the final AIM model, as mentioned in Appendix B. The constructed model consists of two layers. Twenty eight neurons were tried in the first layer, while only two neurons were included at the end of the trial. Only one neuron had been included (by default) for the second layer, which was the pressure drop target.

However three input parameters had shown pronounced effect on the final pressure drop estimate, which were; wellhead pressure, length of the pipe, and angle of inclination. The selection of these three inputs had been conducted automatically without any interference from the user. They were selected based on their mapping influence inside the data set on the pressure drop values.

This topology was achieved after a series of optimization processes by monitoring the performance of the network until the best network structure was accomplished. Fig 4.46 shows the schematic diagram of the proposed AIM topology. Trend analysis has been checked with each model run to make sure the modeling procedure was sound.

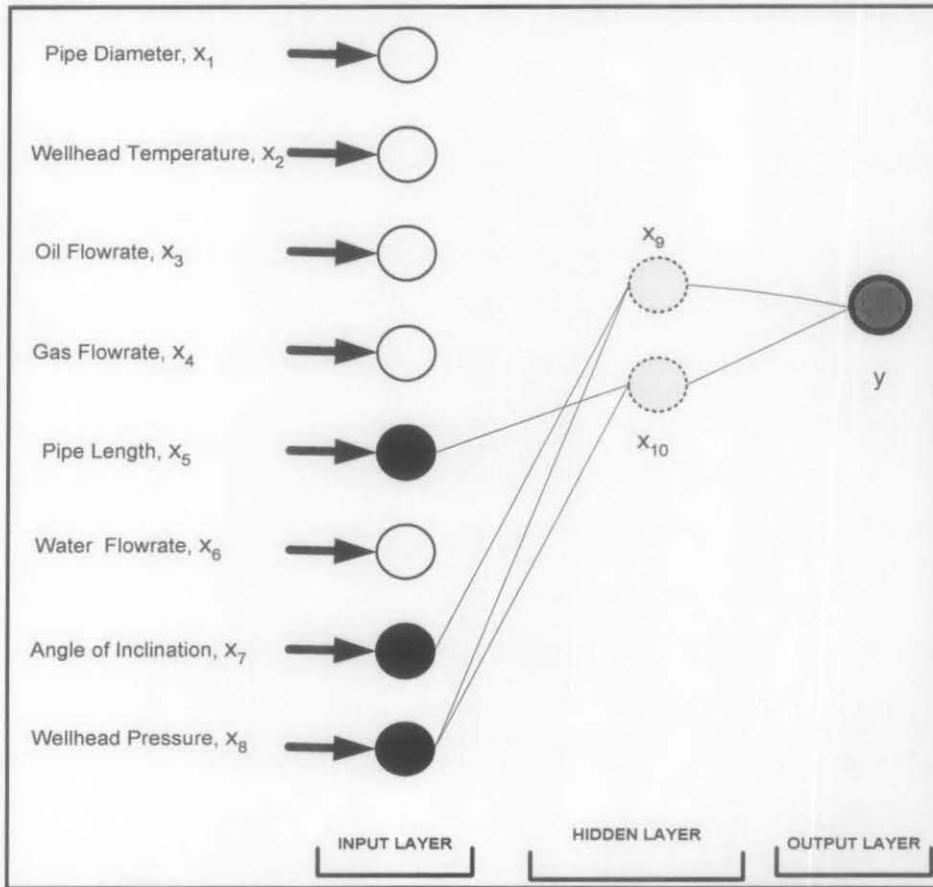


Fig 4.46: Schematic Diagram of the Proposed AIM Topology

4.5.2 Summary of Model's Equation

As described in the previous section the model consists of two layers as follows:

Total Number of layers: 2

Layer #1

Number of neurons: 2 (neurons x_9 and x_{10})

$$x_9 = -428.13059484218 + 3.32804279841806 * x_8 - 0.395894375895042 * x_7 + 0.00219488561608562 * x_7 * x_8 - 0.00470613525745107 * x_8 * x_8 - 0.000813801551583036 * x_7 * x_7$$

$$x_{10} = -404.104040068822 + 3.28280927457335 * x_8 - 0.00560599702533417 * x_5 + 1.7395894539217e-005 * x_5 * x_8 - 0.00474009259349089 * x_8 * x_8 + 3.53811231021166e-008 * x_5 * x_5$$

Layer#2

Number of neurons: 1

$$y = 38.6163548411764 - 0.357238550745703 * x_{10} + 0.349279607055502 * x_9 + 0.0477387718410476 * x_9 * x_{10} - 0.0185457588736114 * x_{10} * x_{10} - 0.0242018021448686 * x_9 * x_9$$

Where;

x_5 = length of the pipe, ft

x_7 = angle of inclination, degrees

x_8 = wellhead pressure, psia

y = simulated pressure drop by AIM Model.

4.6 Trend Analysis for the AIM Model

A trend analysis was conducted for every model's run to check the physical accuracy of the developed model. Depending on the final parameters involved in estimating pressure drop that was obtained automatically by the model; three input variables were found strongly affecting the final output. Those are angle of deviation, length of the pipe, and wellhead pressure.

Only the effect of the first two input parameters will be investigated that should be compatible with the physical phenomenon of the general energy equation (equation 4.2). Fig 4.47 shows the effect of angle of inclination on the pressure drop. The effect of angle of inclination was investigated where all range of angles of inclination was plotted against pressure drop. The model was able to generate the expected trend where pressure drop is known to be an increasing function up to 90 degree and beyond that angle it will become a decreasing function. Additionally, the relationship between the pressure drop and length of the pipe was examined by trend analysis where the length of the pipe was plotted against the simulated pressure drop at four different angles of inclination as shown in Fig 4.48.

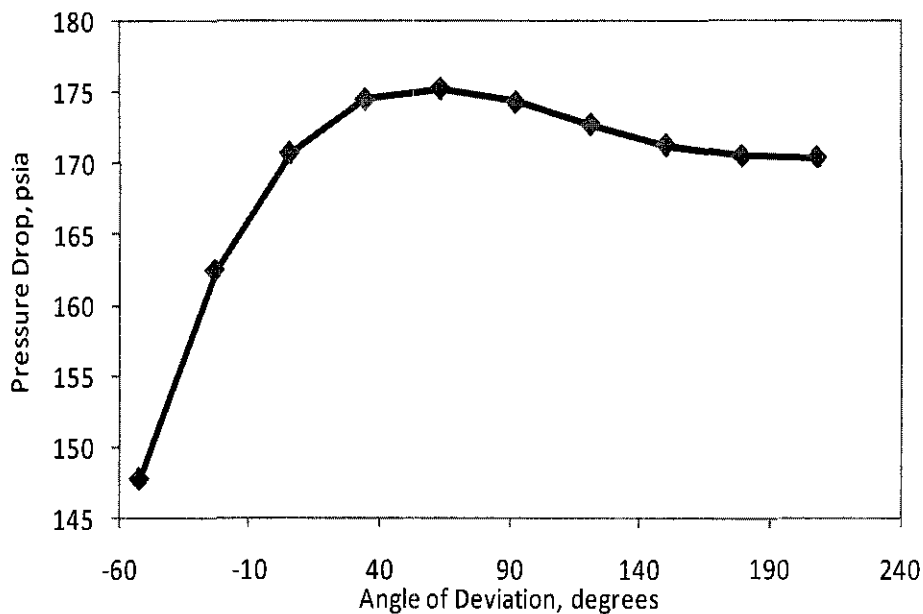


Fig 4.47: Effect of Angle of Inclination on Pressure Drop

Again, and as expected the AIM Model was able to predict the correct phenomenon where the pressure drop is known to be an increasing function with respect to pipe length. Also it is clear that with increasing angle of inclination from downhill to uphill the pressure drop is an increasing function. Again the AIM model was able to produce the right physical trend with respect to angle variation.

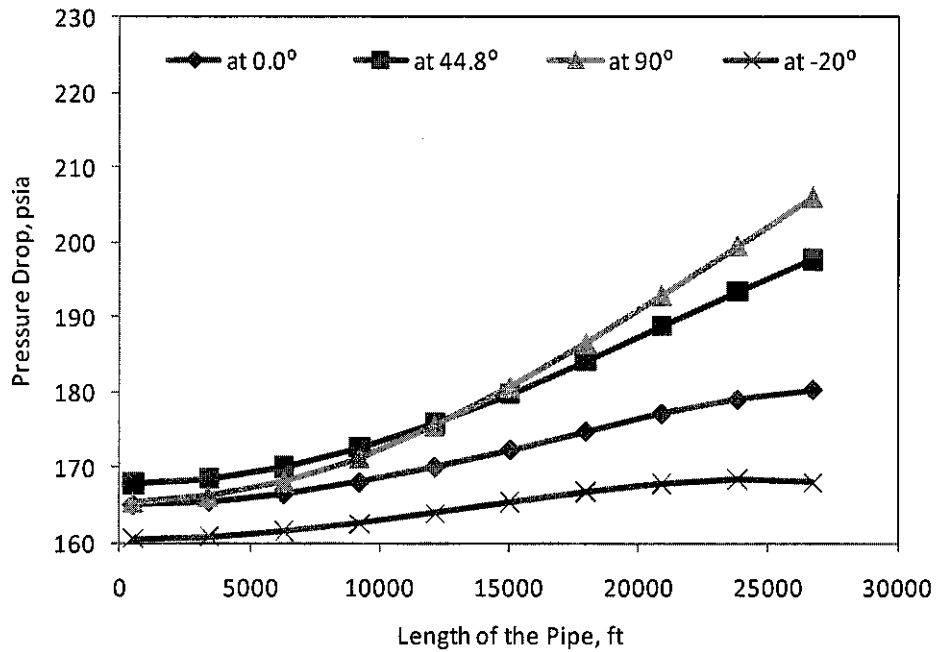


Fig 4.48: Effect of Pipe Length on Pressure Drop at four Different Angles of Inclination

4.7 Group Error Analysis for the AIM Model against Other Investigated Models

To demonstrate the reliability of the developed model, group error analysis was performed. Average absolute relative error is utilized as a powerful tool for evaluating the accuracy of all empirical correlations, mechanistic model, ANN model; as well as the polynomial GMDH model. This effective comparison of all investigated correlations and mechanistic models provides a good means of evaluating models performance since it is used as a main criterion for models evaluation. Average absolute relative error was utilized in this analysis by grouping input parameter and hence plotting the corresponding values of average absolute relative error for each set. Fig 4.49 and Fig 4.50 present the statistical accuracy of pressure drop correlations and models under different groups. Fig 4.49 shows the statistical accuracy of pressure drop grouped by length of the pipe. Length of the pipe had been partitioned into five groups and plotted against the respective average absolute percent relative error for each group.

Polynomial GMDH model was found superior in obtaining the lowest average absolute percent relative error for range of one pipe length groups ($11901 < L < 16000$).

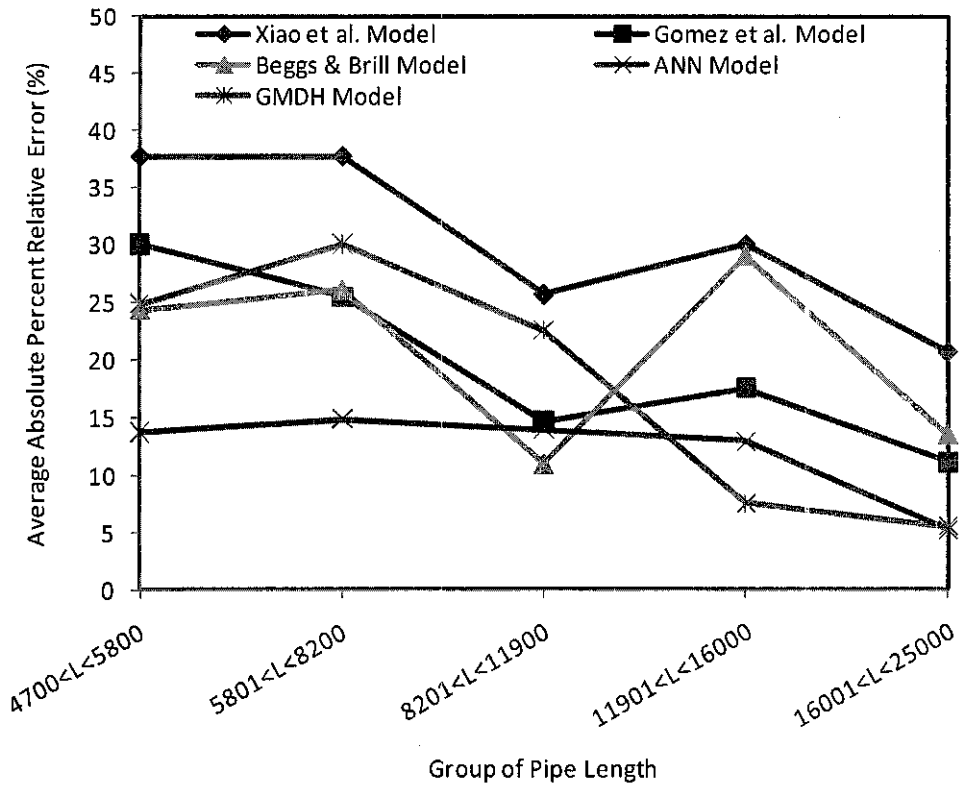


Fig 4.49: Statistical Accuracy of Pressure Drop for the Polynomial GMDH Model and other Investigated Models Grouped by Pipe Length (With Corresponding Data Points)

Furthermore, the statistical accuracy of pressure drop estimation for the polynomial GMDH model against other investigated models grouped by the angle of inclination is plotted in Fig 4.50. Data were partitioned into four categories to include all possible inclination (downhill, horizontal, uphill, and vertical).

As shown in the respective figure, the polynomial GMDH model was found superior only for achieving the lowest average absolute percent relative error for the range of angle of inclination between 90 and 208 (uphill angles only), while ANN model was found optimum in the rest of the tested ranges.

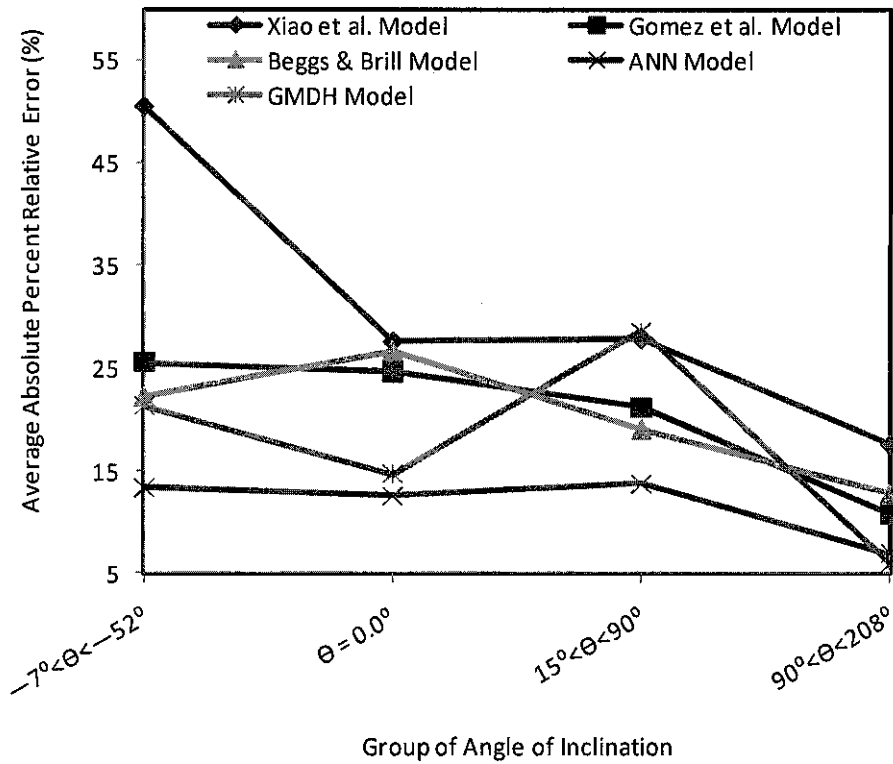


Fig 4.50: Statistical Accuracy of Pressure Drop for the Polynomial GMDH Model and other Investigated Models Grouped by Angle of Inclination (With Corresponding Data Points)

4.8 Statistical and Graphical Comparisons of the Polynomial GMDH Model

4.8.1 Statistical Error Analysis

The same statistical parameters were adopted for comparison for all types of models. Summary of statistical comparisons between all sets (training, validation, and testing) of the polynomial GMDH Model is presented in Table 4-9.

4.8.2 Graphical Error Analysis of the Polynomial GMDH Model

Three graphical analysis techniques were employed to visualize the performance of the Polynomial GMDH Model and other investigated models. Those include cross-plots, error distribution, and residual analysis.

Table 4-9: Statistical Analysis Results of the Polynomial GMDH Model

Data Set Statistical Parameters	Training	Validation	Testing
E_a	18.5282	31.6448	19.5921
E_r	-6.6299	-21.1243	-0.9040
E_{Max}	286.9142	583.0868	130.6760
E_{Min}	0.0862	0.2303	0.0904
RMSE	38.2075	90.9291	33.5273
R "fraction"	0.9771	0.9544	0.9750
STD	12.0291	14.0404	14.3347

4.8.2.1 Cross-plots of the Polynomial GMDH Model

Fig 4.51, Fig 4.52 and Fig 4.53 present cross-plots of predicted pressure drop versus the actual one for Polynomial GMDH Model (training, validation, and testing sets). Fig 4.51 shows a crossplot between predicted and actual pressure drop values for the training set where a correlation coefficient of 0.9771 was obtained by the GMDH model.

The GMDH model showed good agreement between actual and estimated values especially at the middle range (from 70 - 150 psia). However, this measure (correlation coefficient) was not taken as a main criterion for evaluating models performance since it will not give clear insight into the actual error trend while points under the 45° may be recovered by others under the same line. Fig 4.52 is showing another crossplot created by the validation set where predicted pressure drop was plotted against the actual values. A correlation coefficient of 0.9544 is obtained by this model for this data set. Validation set was introduced during training of GMDH model to avoid overtraining. Again, the performance was lower when compared with the proposed ANN validation set where 0.967 is attained.

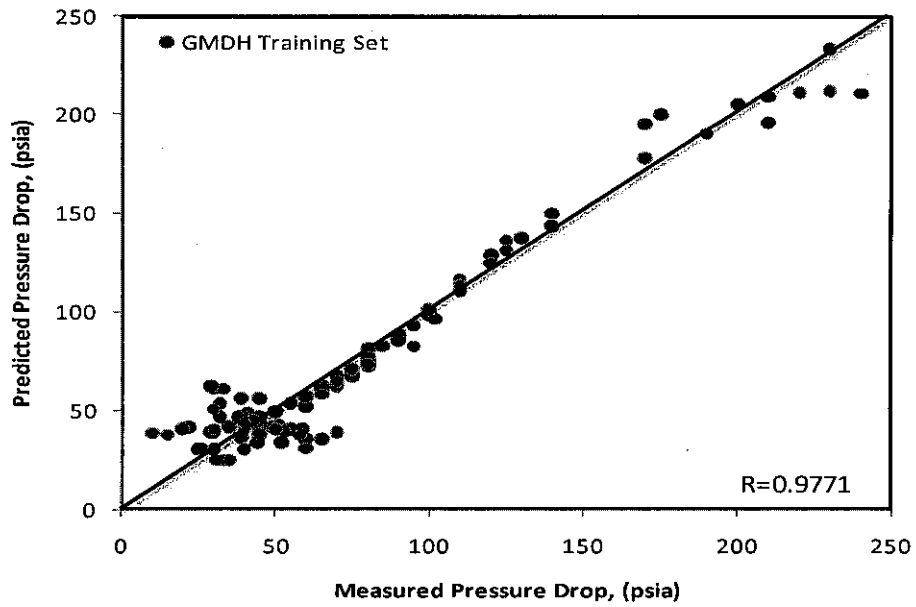


Fig 4.51: Cross-plot of Predicted vs. Measured Pressure Drop for Training Set (Polynomial GMDH Model).

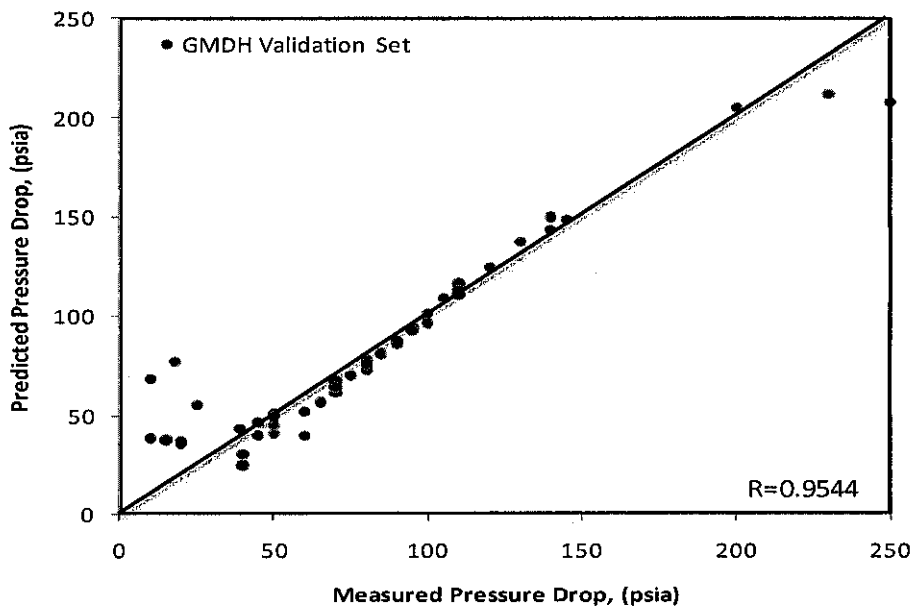


Fig 4.52: Cross-plot of Predicted vs. Measured Pressure Drop for Validation Set (Polynomial GMDH Model)

Fig 4.53 shows a crossplot between estimated and measured pressure drop values for the test set created by the GMDH model. The model achieved reasonable correlation coefficient between estimated and actual values where a value of 0.975 was obtained. Bear in mind that this correlation coefficient was achieved with only

three input parameters; which are angle of inclination, wellhead pressure, and length of the pipe. In addition, the performance of the GMDH may be improved further if more data sets have been introduced with a wide range of tested variables. This may give an indication that most of the input variables used for generating ANN model may serve as noise data.

The main purpose of utilizing this technique is to explore the potential of using GMDH as a tool, for the first time, to predict the pressure drop under wide range of angles of inclination. The exploration includes finding the most influential input parameters in estimating the pressure drop under this wide range of angles of inclination. Fig 4.54 shows a comparison of correlation coefficients for GMDH model against all investigated models; as well as the ANN model. The comparison showed that the ANN model outperformed all investigated models with the highest correlation coefficient.

However, the main criterion for evaluating model's performance, which is the Average Absolute Percent Relative Error, revealed that the GMDH test set achieved the second lowest AAPE after the proposed ANN model with a value of approximately 19.6%, as shown in Fig 4.55. Comparison between the performance of all investigated models plus the polynomial GMDH model is provided in Table 4-10.

Additional criteria for evaluating model's performance are Standard Deviation, Root Mean Square Error (RMSE), Minimum Absolute Percent Relative Error, and Maximum Absolute Percent Relative Error. The GMDH model failed to provide low maximum absolute percent relative error where a value of 130.6% is obtained. On the other hand, the ANN achieved the lowest maximum absolute percent relative error that reaches (44%), as shown in Table 4-10.

If this criterion was selected to evaluate models performance, the GMDH model will be considered as the worst among the rest of investigated models. On contrary, if the minimum absolute percent relative error is considered as the only parameter for evaluating models performance, the GMDH will be ranked second after the Xiao et al. model with a value of 0.0904. Moreover, Fig 4.54 shows a comparison of correlation coefficient for the polynomial GMDH model against all investigated models.

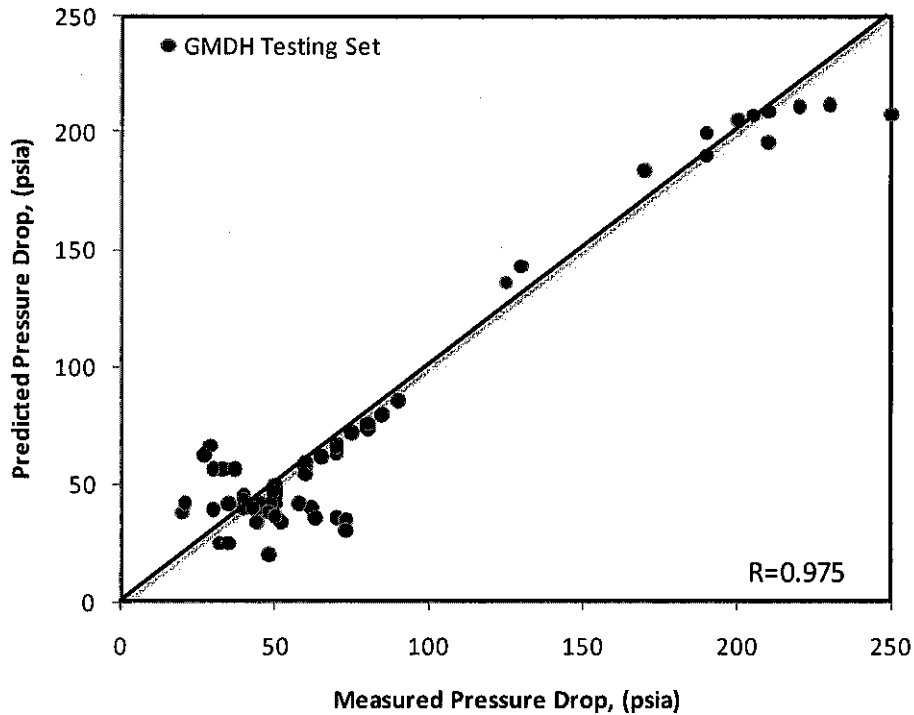


Fig 4.53: Cross-plot of Predicted vs. Measured Pressure Drop for Testing Set (Polynomial GMDH Model)

Root Mean Square Error (RMSE) is used to measure the data dispersion around zero deviation. The lowest RMSE is achieved by ANN model (15.8%) while the GMDH model ranked in the fourth place before the worse model (Xiao et al. model) with a value of 33.53% Fig 4.55 shows a comparison of root mean square errors for the polynomial GMDH model against all investigated models. Comparison between average absolute percent relative error for all tested models; as well as for GMDH model is provided in Fig 4.56. As clearly shown from that figure, the polynomial GMDH model achieved the second best AAPE with a value of 19.6% after the ANN model, which outperforms the rest of the investigated model with a value of 12.11%.

Fig 4.57 shows a comparison of standard deviation for the polynomial GMDH model against the rest of the models. Standard Deviation (STD) was used to measure model advantage. This statistical feature is utilized to measure the data dispersion. A lower value of standard deviation indicates a smaller degree of scatter.

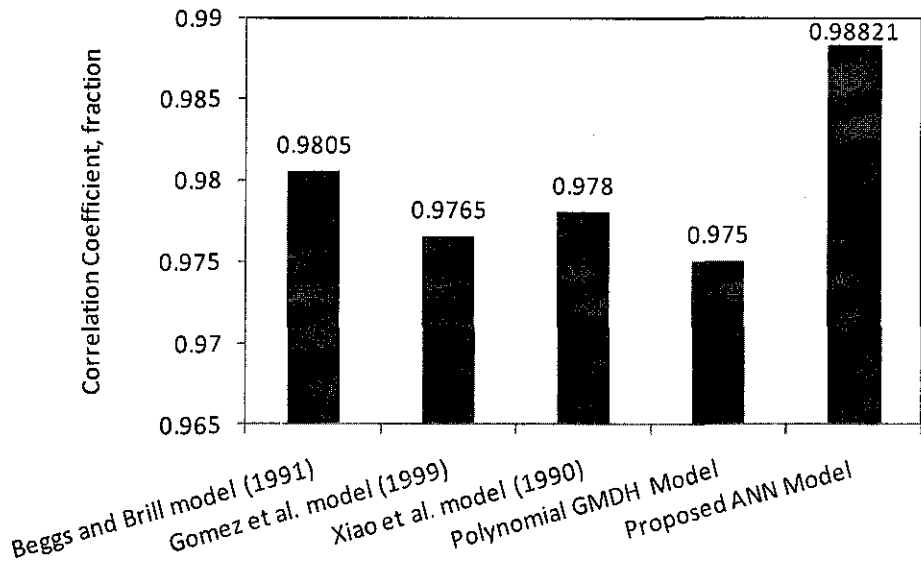


Fig 4.54: Comparison of Correlation Coefficients for the Polynomial GMDH Model against All Investigated Models

Nevertheless, this time the GMDH model came the second best after the ANN model with a value of 14.33, as clearly shown in Table 4-10. Comparison between the performance of all investigated models as well as GMDH model is provided in Table 4-10.

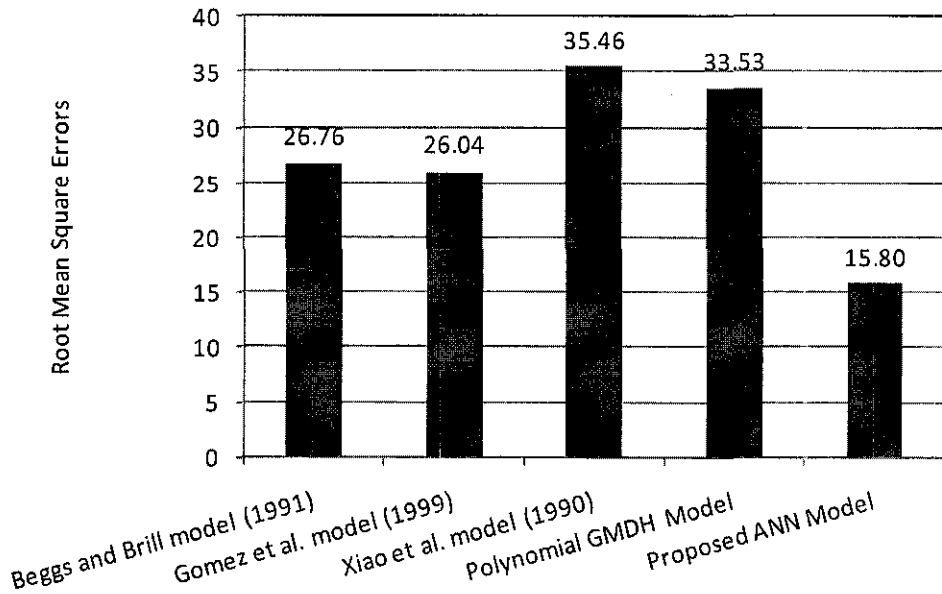


Fig 4.55: Comparison of Root Mean Square Errors for the Polynomial GMDH Model against All Investigated Models

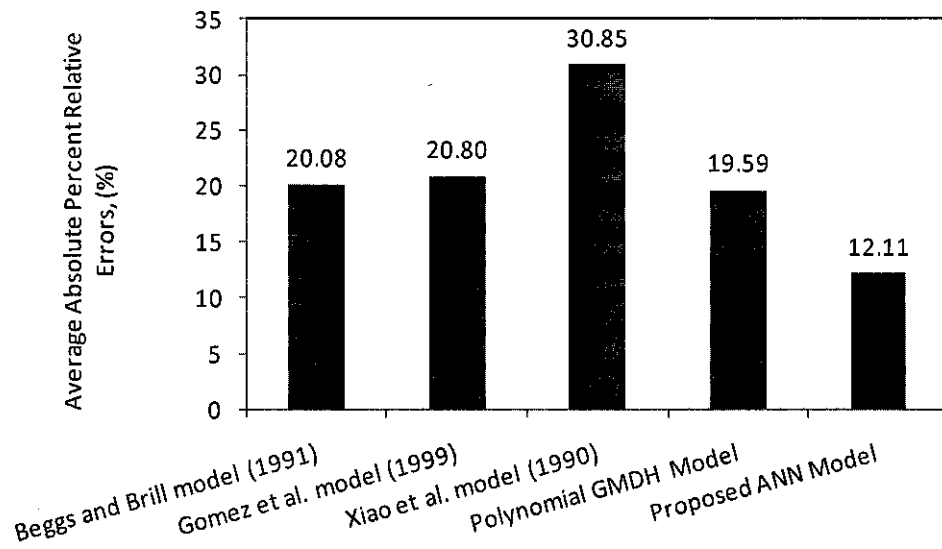


Fig 4.56: Comparison of Average Absolute Percent Relative Errors for the Polynomial GMDH Model against All Investigated Models

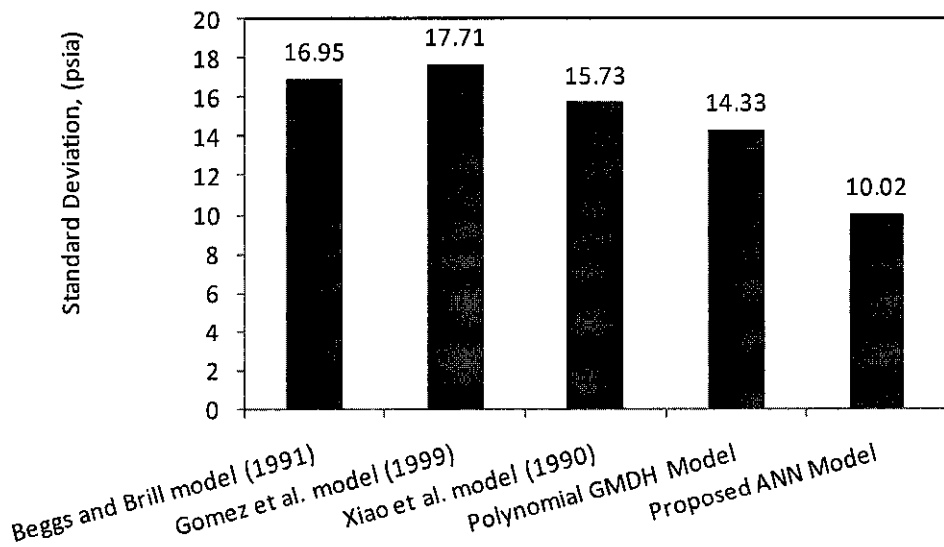


Fig 4.57: Comparison of Standard Deviation for the Polynomial GMDH Model against All Investigated Models

Table 4-10: Statistical Analysis Results of Empirical Correlations, Mechanistic Models, against the Two Developed AIM & ANN models

Statistical Feature	E_a	E_r	E_{Max}	E_{Min}	RMSE	R	STD
Model Name							
Beggs and Brill model	20.076	-10.987	79.00	0.3333	26.7578	0.9805	16.9538
Gomez et al. model	20.802	-2.046	72.65	0.525	26.0388	0.9765	17.7097
Xiao et al. model	30.845	29.818	71.4286	0.0625	35.4582	0.9780	15.7278
Polynomial GMDH Model	19.592	-0.904	130.68	0.0904	33.5273	0.9750	14.3347
Proposed ANN Model	12.11	1.6087	43.99	0.2644	15.795	0.9882	10.016

4.8.2.1 *Error Distributions of the Polynomial GMDH Model against Other Investigated Models*

Fig 4.58, Fig 4.59 and Fig 4.60 show the error distribution histograms for the polynomial GMDH model, (training, validation, and testing sets). Normal distribution curves are fitted to each one of them. The errors are said to be normally distributed with a mean around the 0% and the standard deviation equal to 1.0.

Analyzing the polynomial GMDH model’s error distribution histogram is highly vital for the sake of checking model’s performance for all data sets.

Fig 4.58 shows the error distribution histogram and the normal distribution curve for the training set of the new polynomial GMDH. It showed a slight shift of the mean of the errors towards the negative side of the plot (less than 1%) indicating that the pressure drop was slightly overestimated.

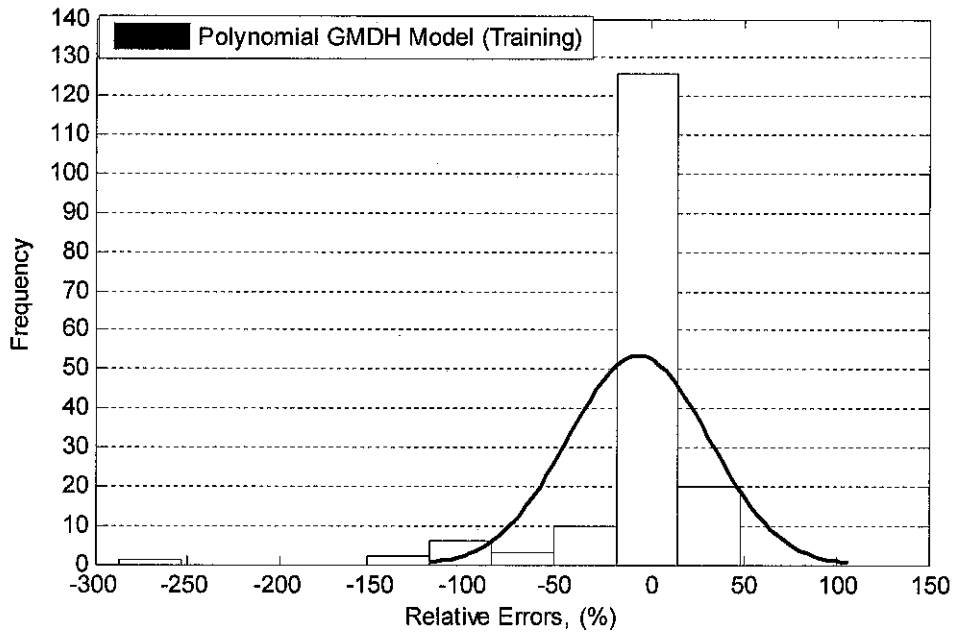


Fig 4.58: Error Distribution for Training Set (Polynomial GMDH Model)

Moreover, as it is seen from the same figure that almost 61.1% of the total error frequencies laid within the normal distribution curve as indicated by twice the standard deviation. The optimum statistical ratio should be $\frac{2}{3}$ of errors lays within the normal distribution curve. It is marked how close this value to the theoretical value (67%).

Fig 4.59 depicts the error distribution histogram and the normal distribution curve for the validation set of the polynomial GMDH model. It shows a considerable skewing of the mean of the errors towards the negative side of the plot (about 21.1%) indicating that the pressure drop is highly overestimated by the model for this set. Following the same approach, it is seen that only 22.6% of the total error frequencies have been presented by the shifted normal distribution curve. This means only 19 cases out of 84 tested cases are presented by the shifted normal distribution curve.

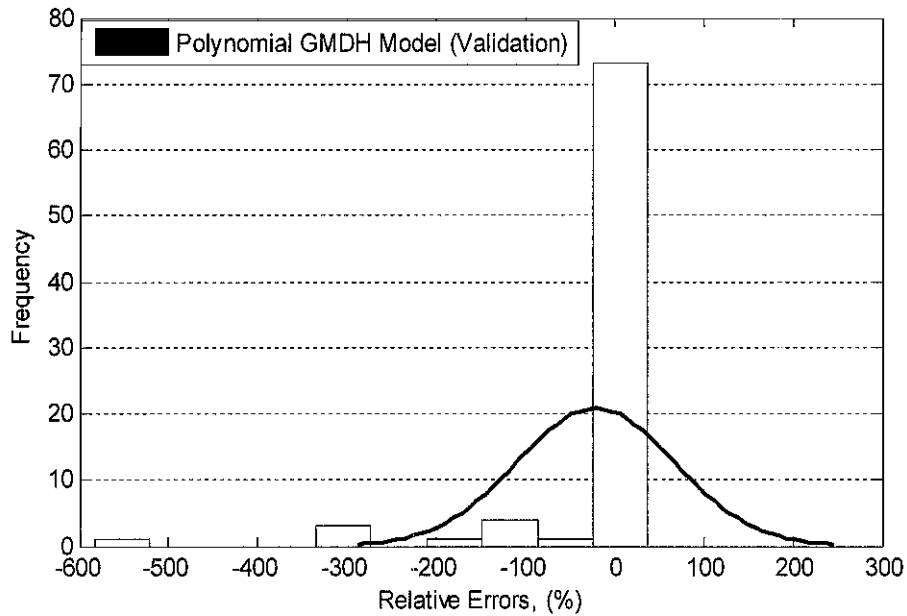


Fig 4.59: Error Distribution for Validation Set (Polynomial GMDH Model)

Additionally, Fig 4.60 illustrates the error distribution histogram and the normal distribution curve for the testing set of the polynomial GMDH model. The mean of the errors is skewed by less than 1% to the left, which indicates excellent representation of errors by the normal distribution curve. It indicates the new proposed ANN model overestimates the pressure drop for the tested region with very minor degree.

Almost 57.4% of the total error frequencies lay within the normal distribution curve. The closest value to the theoretical value of 67%, the better the model is in representing the error trend. This can be replaced by 48 cases presented by the normal distribution curve out of 84 tested cases.

The range of errors also is another essential parameter for detecting the accuracy of models. This range can be extracted from each histogram figures (from Fig 4.58, Fig 4.59 and Fig 4.60). A range of -100% to 100% was used for polynomial GMDH model (training and validation sets) as the best sets of the generated GMDH model when compared to the validation set, which showed drastic error range between -300% to 250% as shown in Fig 4.59.

Needless to mention that the polynomial GMDH model had been created with aid of validation set to prevent over-prediction of pressure drop values. The generated model can be considered the best (based on the represented data) where the regularity criterion (MSE) was used to safeguard the model from being over-trained.

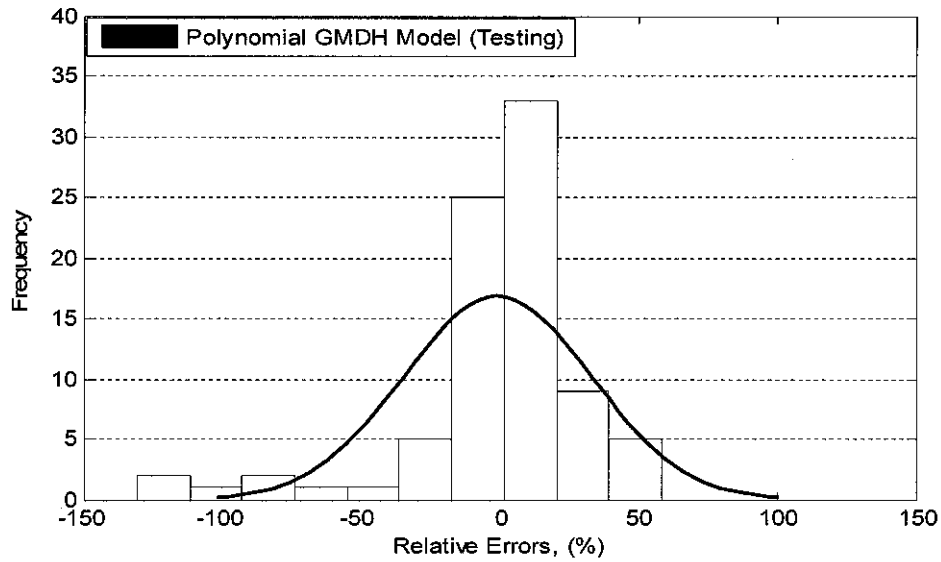


Fig 4.60: Error Distribution for Testing Set (Polynomial GMDH Model)

The Correlation Coefficient and the Average Absolute Percent Errors for each model were put in a tabulated form for easiness of comparing models performance. The rating of model performance was based on having the lowest average absolute percent relative error and highest correlation coefficient. As shown in Table 4-11, the ANN model shows optimum performance compared to the rest of investigated models including the polynomial GMDH model. Polynomial GMDH model ranked second while Beggs & Brill model ranked third. This one followed by Gomez et al. and Xiao et al model. A close result can be extracted when Root Mean Square Errors and the Standard Deviation of errors of each model had been tabulated in Table 4-12. On the contrary, this time the best model will be judged on having the lowest Root Mean Square of Errors followed by the lowest Standard Deviation of Errors. Again the new proposed ANN model achieved the optimum performance, while the rest of the tested models dropped below it. This indicates better-quality performance of ANN model when compared to other tested models.

Table 4-11: Evaluating Models Performance by Average Absolute Percent Errors and Correlation coefficient (Including GMDH Model)

Statistical Feature Correlation / Model	Average Absolute Percent Errors	Correlation Coefficient	Rating
ANN Model (testing data)	12.11	0.9882	1
Polynomial GMDH Model	19.592	0.975	2
Beggs and Brill model	20.076	0.9805	3
Gomez et al. model	20.802	0.9765	4
Xiao et al. model	30.845	0.978	5

Table 4-12: Evaluating Models Performance by Root Mean Square Errors and Standard Deviation of Errors (Including GMDH Model)

Statistical Feature Correlation / Model	Root Mean Square Errors	Standard Deviation of Errors	Rating
ANN Model (testing data)	15.795	10.016	1
Gomez et al. model	26.0388	17.7097	2
Beggs and Brill model	26.7578	16.9538	3
Polynomial GMDH Model	33.5273	14.3347	4
Xiao et al. model	35.4582	15.7278	5

4.8.2.2 Residual Analysis Error Distributions of the Polynomial GMDH Model against all Investigated Models

Residual analysis was utilized to check models consistency. The relative frequency of deviations between estimated and actual values is depicted in Fig 4.61, Fig 4.62 and Fig 4.63 for the Polynomial GMDH model (training, validation, and testing sets). The

purpose of conducting such analysis was to show the error distribution around the zero line to verify whether models have error trends.

Analysis of residual (predicted pressure drop minus the actual pressure drop) is an effective tool to check model deficiencies. Residual limits of investigated model are shown in Table 4-13. Analysis of model's values revealed that Gomez et al. model (refer to Fig 4.43) and Beggs & Brill correlation (refer to Fig 4.44) show the worst negative error performance with maximum values of -82.61 psia and -79.1 psia, respectively. While Xiao et al. model showed the worst positive error performance (57.91 psia), as appears in Fig 4.45.

Analysis of the polynomial GMDH model data sets appear in Fig 4.62, and Fig 4.63. A range of -30 to 33 psia was reported by the training set as shown in Fig 4.61. Regardless of validation lower performance, the set managed to achieve lower range of residual errors as clearly shown in Fig 4.62. Furthermore, a range between -42 to 59 psia was achieved by the validation set. A maximum value of -42 to 37 psia was reported by testing set (refer to Fig 4.63). It is an encouraging indication that GMDH technique can be used successfully to estimate pressure drop values with wide range of angle of inclination. Further room of improvement can be obtained if data with wide range of variables and additional sets can be used. However, the main purpose of using this technique was to explore its potential in predicting pressure drop values and reducing the curse of dimensionality by minimizing the number of input parameters used in prediction without sacrificing the modeling accuracy. It is thought that this objective has been fulfilled successfully.

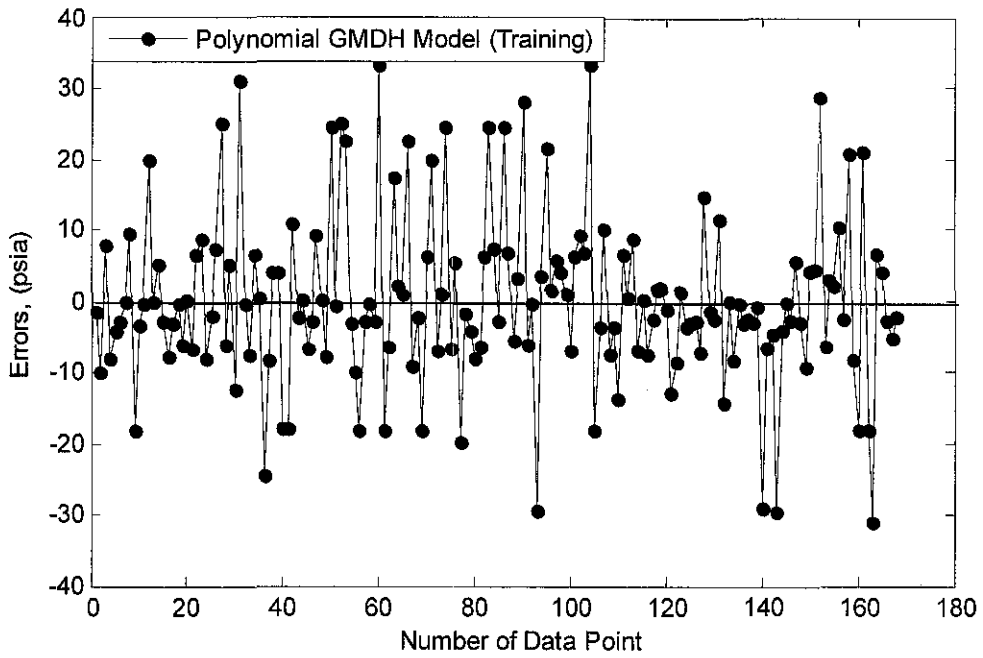


Fig 4.61: Residual Graph for Training Set (Polynomial GMDH Model)

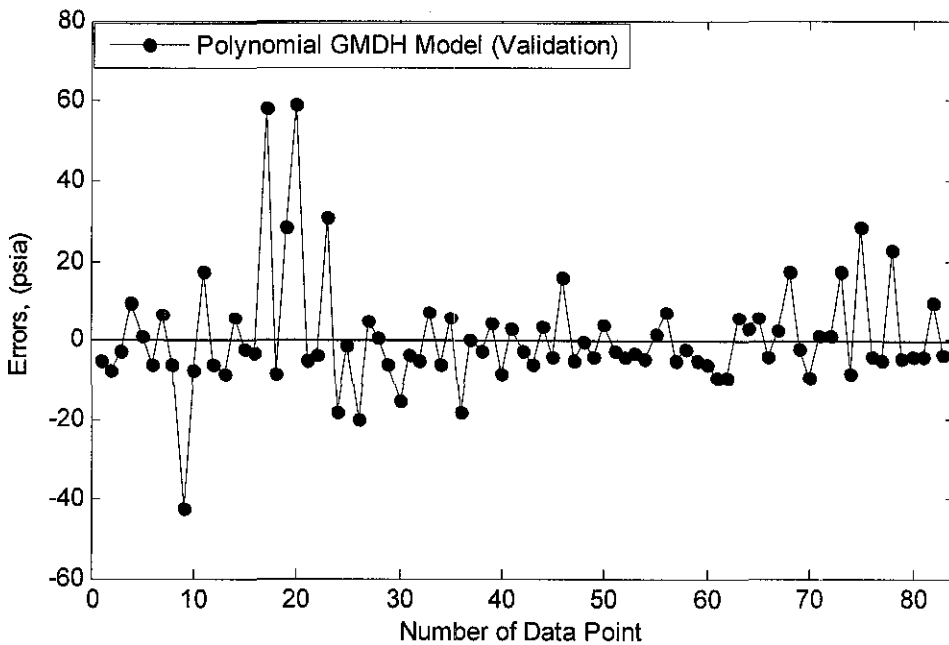


Fig 4.62: Residual Graph for Validation Set (Polynomial GMDH Model)

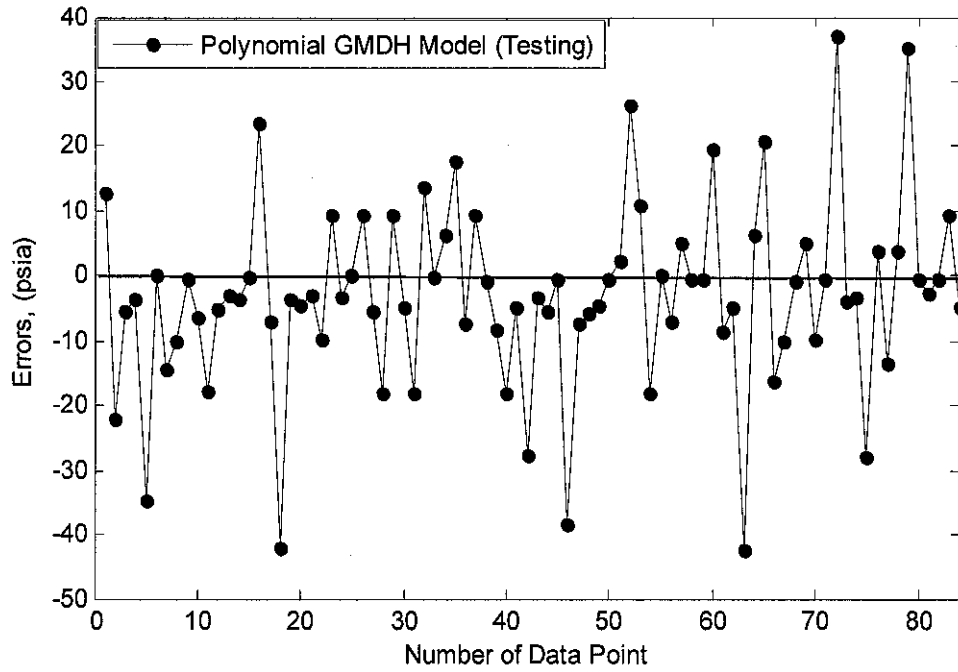


Fig 4.63: Residual Graph for Testing Set (Polynomial GMDH Model)

Table 4-13: Residual limits of the Polynomial GMDH &ANN Models against the Best Investigated Models

Statistical Feature	Maximum	Minimum
ANN Model	26.94	-32.23
Polynomial GMDH Model	37.08	-42.55
Beggs and Brill correlation	16.33	-79.1
Xiao <i>et al.</i> Model	57.91	-9.02
Gomez <i>et al.</i> Model	24.689	-82.61

4.9 Uncertainty Study

The uncertainty of each input parameters has been calculated and tabulated in Table 4-14. It is worthy to mention that these calculations were based on testing data only. The calculation was based on the definition of relative standard deviation around the mean.

Table 4-14 : Uncertainty of Input Variables Used in Testing Models

Input	Gas rate	Water rate	Oil rate	Length	Angle	Diameter	Wellhead pressure	Wellhead Temperature
Uncertainty (%)	±32%	±150%	±42%	±49%	±147%	±21%	±40%	±14%

By using equation 3.13, it is found that the ANN model obtained the expanded measured uncertainty of ±19% at the confidence interval of 95% (default) and at the degree of freedom equal to 83 according to the following formula:

$$[L_\alpha, U_\alpha] = \left[\bar{X} - t_{\alpha, N} \frac{STD}{\sqrt{N}}, \bar{X} + t_{\alpha, N} \frac{STD}{\sqrt{N}} \right] \quad (4.5)$$

where N is the sample size, X is the sample mean, STD is the sample standard deviation, $t_{\alpha, N}$ is the critical value from the Student's t distribution associated with the desired confidence level and the given sample size. α is defined as a confidence risk. Moreover, degree of freedom can be defined as the subtraction of sample size minus one, which is equal to $84-1 = 83$.

Testing set of ANN model with 95% confidence interval has been shown in Fig 4.68 as a representative sample of plotting type. The principle behind confidence intervals was formulated to provide an answer to the question raised in statistical inference of how to deal with the uncertainty inherent in results derived from data. However, these data are themselves only a randomly selected subset of an entire statistical population of possible datasets, [wikipedia, 2011].

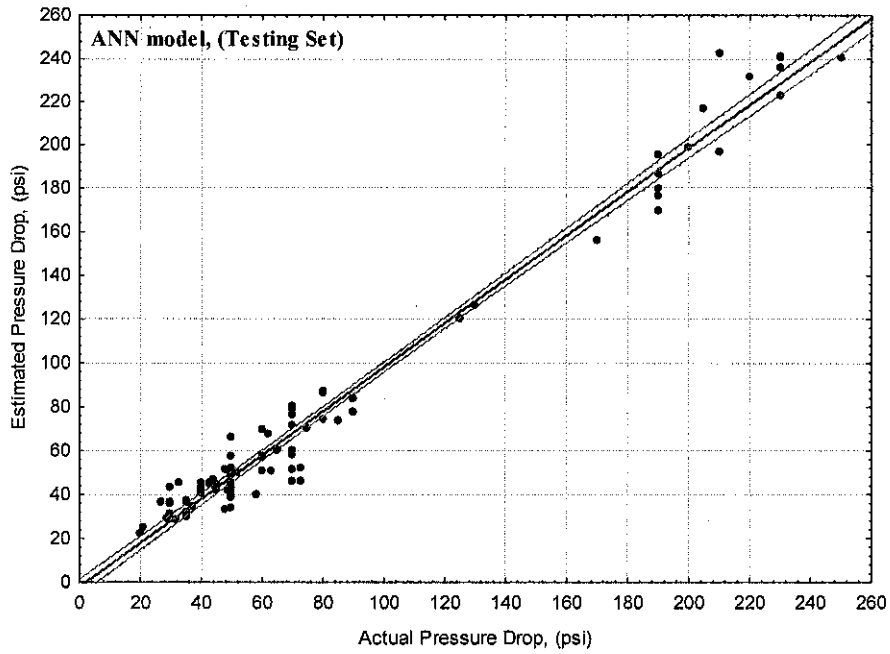


Fig 4.64: Estimated Pressure Drop against Actual Pressure Drop for the Testing Set of ANN Model with 95% Confidence Interval

Table 4-15 shows the uncertainty associated with each model at 95% confidence interval based on the abovementioned methodology.

Table 4-15: Uncertainty Values of All Models

Model	ANN	Beggs & Brill	Gomez et al.	AIM (GMDH)	Xiao et al.
Uncertainty (%)	±19%	±26%	±30%	±34%	±54%
Rating	1	2	3	4	5

The model with less uncertainty value is the most confident and vice versa. Based on the obtained results, ANN model achieved the best results followed by Beggs & Brill correlation. While Gomez et al model ranked third. AIM (GMDH) and Xiao et al models ranked the least certain ones according to this classification.

4.10 Sensitivity Analysis

As described in Section 3.12, it was found that wellhead pressure had the biggest impact on the prediction of pressure drop while the rest of the variables contributed minimally to the total output as shown in Table 4-16. Since the randomized set used for extracting this relative importance factor had a mean square error of 17.4, the single contribution of each input parameter was less than for the original set.

Pressure drop is a direct function of pressure difference between separator pressure and wellhead pressure. It can interpret why wellhead pressure had the greatest share in pressure drop estimation. These values of relative importance are direct indication of data distribution within testing data set. They imply only relative variations from statistical point of view.

This sensitivity analysis can quantify and rate the importance of each input parameter in estimation of pressure drop in pipelines under wide range of angles of inclination. Additionally, some insignificant input parameters can be omitted (depending on the cut-off-trade) if further modeling effort is performed, which might save time and data analysis.

Table 4-16: Relative Importance of Input Variable on ANN Model

Input	Relative Importance (%)	Rating
Wellhead Pressure	34.9	1
Wellhead Temperature	13.3	2
Water Rate	12.7	3
Diameter	11.6	4
Oil Rate	9.6	5
Angle	8.9	6
Gas Rate	6.8	7
Length	2.2	8

4.11 Summary

This Chapter included comprehensive analysis of the results obtained from the current research. The Chapter presented, firstly, the detailed process of ANN model optimization. Trend analysis was achieved successfully by the ANN and GMDH models and checked for their main input parameters. Secondly, group error analysis was conducted to show models performances grouped at certain input parameters and their respective ranges. The bottomline is that statistical and graphical analyses revealed the superiority of the developed ANN model over the investigated correlations and mechanistic models. Average Absolute Percent Error (AAPE) has been chosen as a main statistical criterion for evaluating models' performances. ANN obtained the lowest AAPE of 12.1% while the developed GMDH model obtained 19.6%. A powerful Graphical User Interface (GUI) has been built to aid in applying the results obtained by the ANN model (detailed description of GUI is provided in Appendix C). The GMDH model managed to discover the most relevant and influential input parameters involved in estimating pressure drop with a reasonable degree of accuracy. This can improve the modelling procedure. Finally, the potential of using ANN and AIM techniques in this new area has been investigated successfully.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The following conclusions can be drawn from the current study:

1. The ANN model achieved optimum performance when compared to the Polynomial GMDH Model and to the best available models adopted by industry for estimating pressure drop in pipelines with an outstanding correlation coefficient of 98.82%.
2. Statistical analysis revealed that the ANN model achieved the lowest average absolute percent error, lowest standard deviation, lowest maximum error, and lowest root mean square error.
3. Average Absolute Percent Error, which has been utilized as a main statistical feature for comparing models performances, showed that ANN model is obtained 12.1% while the GMDH model obtained 19.6%.
4. A useful Graphical User Interface tool (GUI) has been built to implement the ANN results through using Visual Basic programming environment.
5. Accurate results can be obtained if wider range of data is used for generating ANN & AIM models. Both two Models can be applied confidently within the range of trained data. Extrapolating data beyond that range might produce erroneous results.
6. Uncertainty analysis revealed that the ANN model was the less uncertain one, followed by Beggs and Brill model.
7. Polynomial GMDH model helps in reducing the problem of dimensionality that lowers the performance of ANN modeling efficiency.

8. No single model had been found reliable for estimating the pressure drop among the investigated old models (Beggs and Brill (1973), Xiao et al. (1990), and Gomez et al. (1999)).

5.2 Recommendations

The following recommendations may be forwarded for future work:

1. Another model can be built using ANN, in which important input parameters such as wellhead pressure, angle of inclination and length of the pipe will be extracted from the generated AIM model.
2. A wide range of data that can be collected from different fields with additional input variables such as oil viscosity, oil density and specific gravity of gas and water phases can be used to construct more robust models using ANN and AIM techniques.
3. Other different vigorous training algorithms such as Polak-Ribiere conjugate gradient can be tried to generate ANN models where the effect of each input parameter can be verified exactly.
4. A double-verification of the current models results can be assessed through using a smart simulator such as OLGA.
5. Again, trend analysis and group error analysis should be conducted to check whether the final proposed model simulates the real behavior.

REFERENCES

- [1] Abduvayt, P., Manabe, R. and Arihara, N., (2003). Effects of Pressure and Pipe Diameter on Gas-Liquid Two-Phase Flow Behavior in Pipelines. SPE Annual Technical Conference and Exhibition. Denver, Colorado, U.S.A, Paper SPE 84229.
- [2] Aggour, M. A., Al-Yousef, H.Y. and Al-Muraikhi, A. J., (1994). Vertical Multiphase Flow Correlations for High Production Rates and Large Tubulars. SPE Annual Technical Conference & Exhibition. New Orleans , LA, Paper SPE 28465.
- [3] Ahmed, O, Abdel-Aal, R and AlMustafa, H, (2010). "Reservoir property prediction using abductive networks." *Geophysics* 75: P1.
- [4] Al-Ne'aim, S., Aggour, M. and Al-Yousef, H., (1995). "Horizontal Multiphase Flow Correlations for Large Diameter Pipes and High Flow Rates." Middle East Oil Show.
- [5] Aleksander, I. and Morton, H., (1990). *An Introduction to Neural Computing*. London, Chapman & Hall.
- [6] American National Research Council., (1980). *Estimating Population and Income for Small Places*. Washington, D.C, National Academy Press.
- [7] Andritsos, N., Williams, L. and Hanratty, J., (1989). "Effect of Liquid Viscosity on the Stratified-Slug Transition in Horizontal Pipe Flow." *International Journal of Multiphase Flow* 15: 877.
- [8] Ansari, M., Sylvester, N., Sarica, C., Shoham, O. and Brill, J., (1994). "A Comprehensive Mechanistic Model for Upward Two-Phase Flow in Wellbores." *SPEPF Journal*: 217-226.

- [9] Arirachakaran, S., Jefferson, L., Brill, J. and Shoham, O., (1991). Intelligent Utilization of a Unified Flow Pattern Prediction Model in Production System Optimization. 66th Annual Technical Conference and Exhibition of the Society of Petroleum Engineers. Dallas, Texas, Paper SPE 22869.
- [10] Arya, A. and Thomas, L., (1981). Comparison of Two Phase Liquid Holdup and Pressure Drop Correlations across Flow Regime Boundaries for Horizontal and Inclined Pipes. 56th Annual Fall Technical Conference and Exhibition of the Society of Petroleum Engineers of AIME. San Antonio, USA, Paper SPE 10169.
- [11] Ashby, W. R., (1952). Design for A Brain. New York, Wiley.
- [12] Ayoub, M. A., (2004). Development and Testing of an Artificial Neural Network Model for Predicting Bottom-hole Pressure in Vertical Multiphase Flow. Dhahran, Saudi Arabia, King Fahd University of Petroleum and Minerals. Msc Dissertation.
- [13] Aziz, K. and Petalas, N., (1994). New PC-based Software for Multiphase Flow Calculations. SPE Petroleum Computer Conference. Dallas, TX, Paper SPE 28249.
- [14] Baker, A., Nielson, K. and Gabb, A., (1988). "Pressure Loss, Liquid-Holdup Calculations Developed." Oil Gas J.:(United States) 86(11).
- [15] Baker, O., (1954). "Simultaneous Flow of Oil and Gas." Oil & Gas Journal: 185-195.
- [16] Bankes, S., (1993). "Exploratory Modeling for Policy Analysis." Operations Research 41(3): 435-449.
- [17] Barnea, D., (1987). "A Unified Model for Predicting Flow Pattern Transitions for the Whole Range of Pipe Inclinations." International Journal of Multiphase Flow 13(No. 1): 1-12.

- [18] Barnett, V., (1978). "The Study of Outliers: Purpose and Models." *Applied Statistics* 27: 242-250.
- [19] Beggs, H. D., (1972). *An Experimental Study of Two-Phase Flow in Inclined Pipes*. Tulsa, OK, University of Tulsa. PhD Dissertation.
- [20] Beggs, H. D. and Brill, J. P., (1973). "A Study of Two-Phase Flow in Inclined Pipes." *Journal of Petroleum Technology* 25: 607-617.
- [21] Beggs, H. Dale, (2003). *Production Optimization Using Nodal Analysis*, OGCi and Petroskills publication.
- [22] Bendiksen, K. H., Maines, D. , Moe, R. and Nuland, S., (1991). "The Dynamic Two-Fluid Model OLGA: Theory and Application." *SPE Production Engineering*: 171-180.
- [23] Berry, M.J and Linoff, G, (1997). *Data Mining Techniques*, NY: John Wiley & Sons.
- [24] Bevington, P. R. and Robinson, D. K., (2003). *Data Reduction and Error Analysis for the Physical Sciences*. New York, McGraw-Hill-New York.
- [25] Bharath, R., (1998). *Multiphase Flow Models Range of Applicability*. CTES (Coiled Tubing Engineering Services) Publication. 18.
- [26] Bilgesu, H. I., Altmis, U., Ameri, Mohaghegh, S. and Aminian, S. K., (1998). *A new Approach to Predict Bit Life Based on Tooth or Bearing Failures*. SPE East Regional Conference. Pittsburgh, Paper SPE 51082.
- [27] Bilgesu, H. and Ternyik, J., (1994). *A New Multiphase Flow Model for Horizontal, Inclined, and Vertical Pipes*. Eastern Reigonal Conference & Exhibition. Charleston, WV, USA., Paper SPE 29166: 87-94.
- [28] Brill, J., (1987). "Multiphase Flow in Wells." *Journal of Petroleum Technology(SPE 16242)*: 15-21.
- [29] Brill, J. and Arirachakaran, S., (1992). "State of The Art in Multiphase Flow." *Journal of Petroleum Technology*: 538-541.

- [30] Brill, J. and Mukherjee, H., (1999). Multiphase Flow in Wellbores, SPE Monograph.
- [31] Brookbank, M. and Fagiano, P., (1975). A Review of the Theory and Practice of Two-Phase Flow in Gas-Liquid Transmission Systems.
- [32] Cheremisinoff, N. and Davis, E., (1979). "Stratified Turbulent-Turbulent Gas-Liquid Flow." *AIChE Journal* 25(No. 1): 48-56.
- [33] Corteville, J., Duchet-Suchaux, P. and Lopez, D., (1991). Comparison of Methods for Predicting Pressure Loss in Oil and Gas Wells. BHRG's 6th International conference on Multiphase Production. Cannes, France.
- [34] Dhulesia, H. and Lopez, D., (1996). Critical Evaluation of Mechanistic Two-Phase Flow Pipeline and Well Simulation Models. SPE Annual Technical Conference and Exhibition, Denver, Colorado, USA.
- [35] Dukler, A. E., Wickes, M. and Cleveland, R.G., (1964). "Frictional Pressure Drop in Two-Phase Flow: B. an Approach through Similarity Analysis." *AIChE J* 10(No. 1): 44-51.
- [36] Duns, H. J. and Ros, N. C., (1963). Vertical Flow of Gas and Liquid Mixtures in Wells. The 6th World Petroleum Congress.
- [37] Eaton, B., Knowles, C. and Silberberg, I., (1967). "The Prediction of Flow Patterns, Liquid Holdup and Pressure Losses Occurring During Continuous Two-Phase Flow in Horizontal Pipelines." *Journal of Petroleum Technology* 19(6): 815-828.
- [38] El-Sebakhy, E., Sheltami, T., Al-Bokhitan, S., Shaaban, Y., Raharja, P. and Khaeruzzaman, Y., (2007). Support Vector Machines Framework for Predicting the PVT Properties of Crude Oil Systems.
- [39] Engelbrecht, A.P., (2007). *Computational intelligence: An Introduction*, Wiley.

- [40] Faille, I. and Heintze, E., (1996). Rough Finite volume schemes for modeling two-phase flow in a pipeline. Proceedings of the CEA, EDFINRIA course "Méthodes numériques pour les écoulements diphasiques, Rocquencourt, France.
- [41] Felizola, H. and Shoham, O., (1995). "A Unified Model for Slug Flow in Upward Inclined Pipes." ASME Journal of Energy Resources Technology 117: 1-6.
- [42] Flanigan, O., (1958). "Effect of Uphill Flow on Pressure Drop in Design of Two-Phase Gathering Systems." Oil and Gas Journal 56(132).
- [43] Gabor, D., (1954). "Communication Theory and Cybernetics." RE Transaction on Circuit Theory ICT-1: 19-31.
- [44] Gomez, L. E., Shoham, O., Schmidt, Z., Chokshi, R. N., Brown, A. and Northug, T., (1999). A Unified Mechanistic Model for Steady-State Two-Phase Flow in Wellbores and Pipelines. SPE Annual Technical Conferences and Exhibition. Houston, Texas, Paper SPE 56520.
- [45] Gori, M. and Tesi, A., (1992). "On the Problem of Local Minima in Back-Propagation." IEEE Transactions on Pattern Analysis and Machine Intelligence 14: 76-86.
- [46] Hagedorn, A. R. and Brown, K.E., (1965). "Experimental Study of Pressure Gradient Occurring During Continuous Two-Phase Flow in Small Diameter Vertical Conduits." Journal of Petroleum Technology: 475-484.
- [47] Hall, A., (1992). Multiphase Flow of Oil, Water and Gas in Horizontal Pipes. London, Imperial College. PhD Dissertation.
- [48] Hamby, D. M., (1994). "A review of techniques for parameter sensitivity analysis of environmental models." Environmental Monitoring and Assessment 32(2): 135-154.
- [49] Hasan, A. and Kabir, S., (1988). "A Study of Multiphase Flow Behavior in Vertical Wells." SPE Production Engineering, AIME 285: 263-272.

- [50] Hasan, R. and Kabir, S., (2005). A Simple Model for Annular Two-Phase Flow in Wellbores. SPE Annual Technical Conference and Exhibition. Dallas, Texas, U.S.A, Paper SPE 95523.
- [51] Haykin, S., (1994). Neural Network: A Comprehensive Foundation. NJ, Macmillan Publishing Company.
- [52] Haykin, S., (1999). Neural Network: A Comprehensive Foundation, Upper Saddle River, NJ, Prentice Hall
- [53] Hebb, D. O., (1949). The Organization of Behavior: A Neuropsychological Theory. New York, Wiley.
- [54] Hoaglin, D., Mosteller, F. and Tukey, J. , Eds. (1983). Introduction to More Refined Estimator. New York:, John Wiley. .
- [55] Holditch, SA., Xiong, H., Rueda, J. and Rahim, Z., (1993). Using an Expert System To Select the Optimal Fracturing Fluid and Treatment Volume. the SPE Gas Technology Symposium. Calgary, Alberta, Canada, Paper SPE 26188.
- [56] Hong, Y. and Zhou, D., (2008). Evaluation of Two-Phase-Flow Correlations and Mechanistic Models for Small-Diameter Pipelines at Slightly Inclined Downward Flow. SPE Eastern Regional/AAPG Eastern Section Joint Meeting, Pittsburgh, Pennsylvania.
- [57] Ilobi, M. and Ikoku, C., (1981). Minimum Gas Flow Rate for Continuous Liquid Removal in Gas Wells. SPE Annual Technical Conference and Exhibition, San Antonio, Texas.
- [58] Ivakhnenko, A.G. and Yurachkovsky, Y., (1986). "Modeling of Complex Systems after Experimental Data." Moscow: Radio i Svyaz: .118.
- [59] Izmiran. (2010). "<http://matlab.izmiran.ru/help/toolbox/nnet/backpr58.html>." Retrieved December 21, 2010.

- [60] James, A. Freeman. and David, M. Skapura., (1991). *Neural Networks: Algorithms, Applications, and Programming Techniques*, Addison-Wesley Publishing Company.
- [61] Jekabsons, G. (2010). "GMDH-Type Polynomial Neural Networks for Matlab." Retrieved September 20, 2010, from <http://www.cs.rtu.lv/jekabsons/>.
- [62] Kemp, S.J., Zaradic, P. and Hansen, F., (2007). "An Approach for Determining Relative Input Parameter Importance and Significance in Artificial Neural Networks." *Ecological Modelling* 204(3-4): 326-334.
- [63] Khor, S., Mendes-Tatsis, M. and Hewitt, G., (1997). "One-Dimensional Modeling of Phase Holdups in Three-Phase Stratified Flow." *International Journal of Multiphase Flow* 23(5): 885-897.
- [64] Kordyban, E. and Ranov, S., (1970). "Mechanism of Slug Formation in Horizontal Two-Phase Flow." *Journal of Basic Engineering* 92: 857-864.
- [65] Kramer, A. H. and Sangiovanni, V. A., (1989). *Efficient Parallel Learning Algorithms for Neural Networks*. *Advances in Neural Information Processing Systems 1* (D. S. Touretzky, Edition), San Mateo, CA, Morgan Kaufmann.
- [66] Lagiere, M., Miniscloux, C. and Roux, A, (1984). "Computer Two Phase Flow Model Predicts Pipeline Pressure and Temperature Profiles." *Oil and Gas J.*: 82-92.
- [67] Laurinat, J.E., Hanratty, T.J. and Jepson, W.P, (1985). "Film Thickness Distribution for Gas-Liquid Annular Flow in a Horizontal Pipe." *International Journal of Multiphase Flow* 6(1/2): 179-195.
- [68] Lee, B. Y., Liu, H. S. and Tarng, Y. S. , (1995). *Abductive Network for Predicting Tool Life in Drilling*. *Conference on Industrial Automation and Control: Emerging Technology Applications*, Taipei, Taiwan, IEEE

transactions on Industry Applications by the Industrial Automation and Control Committee of the IEEE Industry Applications Society.

- [69] Lee, B. Y., Liu, H. S. and Tarng, Y. S. , (1998). "Modeling and Optimization of Drilling Process." *Journal of Materials Processing Technology* 74(1-3): 149-157.
- [70] Levy, P. and Lemeshow, S., Eds. (1991). *Sampling of Populations: Methods and Applications*. New York, John Wiley.
- [71] Lim, JS., Park, HJ. and Kim, JW., (2006). A new Neural Network Approach to Reservoir Permeability Estimation from Well Logs. SPE Asia Pacific Oil & Gas Conference and Exhibition. Adelaide, Australia, Paper SPE 100989.
- [72] Lockhart, RW and Martinelli, RC, (1949). "Proposed Correlation of Data for Isothermal Two-Phase, Two-Component Flow in Pipes." *Chem. Eng. Prog* 45(1): 39-48.
- [73] Madala, H.R. and Ivakhnenko, A.G., (1994). *Inductive Learning Algorithms for Complex System Modeling*. Florida, USA, CRC Press, inc.
- [74] Manabe, R. and Arihara, N., (1996). Experimental and Modeling Studies in Two-Phases Flow in Pipelines. SPE Asia Pacific Oil and Gas Conference. Adelaide, Australia, Paper SPE 37017.
- [75] Manabe, R., Zhang, H. Q., Delle, E., Brill, J. P. and Mukherjee, H., (2001). Crude Oil-Natural Gas Two Phase Flow Pattern Transition Boundaries at High Pressure Conditions. SPE Annual Technical Conference and exhibition. New Orleans, SPE Paper 71563.
- [76] Mandhane, JM., Gregory, GA. and Aziz, K., (1974). "A Flow Pattern Map for Gas-Liquid Flow in Horizontal Pipes." *International Journal of Multiphase Flow* 1(4): 537-553.
- [77] Masella, J. M. , Tran, Q. H. , Ferre, D. and Pauchon, C. , (1998). "Transient Simulation of Two Phase Flows in Pipes." *International Journal of Multiphase Flow* 24.

- [78] MATLAB, "Mathwork, Neural Network Toolbox Tutorial. (2009)." (http://www.mathworks.net/MATLAB/Neural_Networks/index.html)."
Retrieved February 12, 2009.
- [79] McCulloch, W. S. and Pitts, W., (1943). "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bulletin of Mathematical Biophysics* 5: 115-133.
- [80] Memmi, D., (1989). *Connectionism and Artificial Intelligence*. Neuro-Nimes, International Workshop on Neural Network and Their Applications. Nimes, France: 17-34.
- [81] Minsky, M. and Papert, S., (1969). *Perceptrons*. Cambridge, MA, MIT Press.
- [82] Minsky, M. and Papert, S., (1988). *Perceptrons: Expanded Edition*, MIT Press, Cambridge, MA, .
- [83] Mohaghegh, S. and Ameri, S., (1995). *Artificial Neural Network as A Valuable Tool for Petroleum Engineers*. an Unsolicited Paper for Society of Petroleum Engineers, Paper SPE 29220.
- [84] Mohaghegh, S., Arefi, R., Belgesu, L., Ameri, S. and Rose, D., (1995). *Design and Development of Artificial Neural Network for Estimation of Formation Permeability*. SPE Production Computer Conference. Dallas, Paper SPE 28237.
- [85] Mukherjee, H., (1979). *An Experimental Study of Inclined Two-Phase Flow*. Oklahoma, University of Tulsa. PhD Dissertation.
- [86] Mukherjee, H. and Brill, J.P., (1985). "Pressure Drop Correlations for Inclined Two-Phase Flow." *Journal of Energy Resources Technology trans., ASME*: 549-554.
- [87] Novikoff, A., (1963). *On Convergence Proofs for Perceptrons*. Defense Technical Information Center, Stanford Research Inst. Menlo Park California.

- [88] Osman, E. A., Abdel-Wahab, O. A. and Al-Marhoun, M. A., (2001). Prediction of Oil PVT Properties Using Neural Networks. 12th MEOS (Middle East Oil and Gas Show and Conference). Bahrain, Paper SPE 68233.
- [89] Osman, E.A. and Abdel-Aal, R.E., (2002). Abductive Networks: A New Modeling Tool for the Oil and Gas Industry. SPE Asia Pacific Oil & Gas Conference and Exhibition. Melbourne, Australia, Paper SPE 77882.
- [90] Osman, S. A., (2001). Artificial Neural Networks Models for Identifying Flow Regimes and Predicting Liquid Holdup in Horizontal Multiphase Flow. SPE Middle East Oil and Gas Show and Conference. Bahrain, Paper SPE 68219.
- [91] Osman, S. and Aggour, M., (2002). "Artificial Neural Network Model for Accurate Prediction of Pressure Drop in Horizontal-Multiphase Flow." *Petroleum Science and Technology* 20 (No.1 and 2): 1-15.
- [92] Oyeneyin, M. B. and Faga, A. T., (1999). Formation-Grain-Size Prediction Whilst Drilling: A Key Factor in Intelligent Sand Control Completion. SPE Annual Technical Conference & Exhibition. Huston, Texas, Paper SPE 56626.
- [93] Ozon, P.M., Ferschneider, G. and Chwetzof, A., (1987). A New Multiphase Flow Model Predicts Pressure and Temperature Profiles. European Offshore Conference. Aberdeen, Paper SPE 16535.
- [94] Park, HJ. and Kang, JM., (2006). "Polynomial Neural Network Approach for Prediction of Liquid Holdup in Horizontal Two-Phase Flow." *Energy Sources, Part A: Recovery, Utilization, and Environmental Effects* 28(9): 845-853.
- [95] Pauchon, C., Dhulesia, H., Lopez, D. and Fabre, J., (1993). TACITE: A Comprehensive Mechanistic Model for Two-Phase Flow. 6th BHRG Multiphase International Conference. Cannes, France.

- [96] Payne, G. A., Palmer, C. M., Brill, J. P. and Beggs, H. D., (1979). "Evaluation of Inclined-Pipe Two-Phase Liquid Holdup and Pressure Loss Correlations using Experimental Data." *Journal of Petroleum Technology AIME*, 267: 1198-1208.
- [97] Paz, R.J. and Shoham, O., (1994). Film Thickness Distribution for Annular Flow in Directional Wells: Horizontal to Vertical. 69th Annual Technical Conference and Exhibition of the Society of Petroleum Engineering. New Orleans, LA, Paper SPE 28541
- [98] Petalas, N. and Aziz, K., (2000). "A Mechanistic Model for Multiphase Flow in Pipes." *Journal of Canadian Petroleum Technology* 39(6).
- [99] Pucknell, J. K., Mason, J. N. E. and Vervest, E.G., (1993). An Evaluation of Recent Mechanistic Models of Multiphase Flow for Predicting Pressure Drops in Oil and Gas Wells. Offshore European Conference. Aberdeen, Paper SPE 26682.
- [100] Riedmiller, M. and Braun, H., (1994). RPROP-Description and Implementation Details, Citeseer.
- [101] Ros, N. C., (1961). "Simultaneous Flow of Gas and Liquid as Encountered in Well Tubing." *Journal of Petroleum Technology*: 1037-1049.
- [102] Rosenblatt, F., (1958). "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review* 65: 386-408.
- [103] Rosenblatt, F., (1962). Principles of neurodynamics: Perceptrons and the theory of brain mechanisms, Spartan books Washington, DC.
- [104] Roux, A., Corteville, J. and Bernicot, M., (1988). WELLSIM and PEPITE: Accurate Models of Multiphase Flow in Oil Wells and Risers. SPE International Meeting on Petroleum Engineering. Tianjin, China, Paper SPE 17576.

- [105] Rumelhart, D. E., Hinton, G. E. and Williams, R. J., (1986). "Learning Representations by Back-Propagation Errors." *Nature*, London 323: 533-536.
- [106] Sage, A. P., (1990). *Concise Encyclopedia of Information Processing in Systems and Organizations*. New York, Pergamon.
- [107] Saltelli, A. and ebrary, Inc., (2008). *Global Sensitivity Analysis, the Primer*, John Wiley & Sons.
- [108] Sawaragi, Y., Soeda, T., Tamura, H., Yushimura, T., Ohe, S., Chujo, Y. and Ishihara, H., (1979). "Statistical Prediction of Air Pollution Levels Using Non-Physical Models." *Automatica (IFAC)* 15(4): 441-452.
- [109] Semenov, A., Oshmarin, R., Driller, A. and Butakova, A., (2010). Application of Group Method of Data Handling for Geological Modeling of Vankor Field. SPE North Africa Technical Conference and Exhibition, Cairo, Egypt, Paper SPE 128517.
- [110] Sevigny, R., (1962). *An Investigation of Isothermal, Concurrent, Two-Fluid Two-Phase Flow in an Inclined Tube*. Rochester, N, Y, U. of Rochester. PhD Dissertation.
- [111] Shepherd, G. M. and Koch, C., (1990). *Introduction to Synaptic Circuits, in the Synaptic Organization of the Brain*. New York, Oxford University Press: 3-31.
- [112] Shoham, O. and Taitel, Y., (1984). "Stratified Turbulent-Turbulent Gas Liquid Flow in Horizontal and Inclined Pipes." *AIChE Journal* 30: 377-385.
- [113] Simmons, M. and Hanratty, T., (2001). "Transition from Stratified to Intermittent Flows in Small Angle Upflows." *International Journal of Multiphase Flow* 27(4): 599-616.

- [114] Stoitsits, R., Crawford, K., MacAllister, D., McCormack, M., Lawal, A. and Ogbe, D., (1999). Production Optimization at the Kuparuk River Field Utilizing Neural Networks and Genetic Algorithms. Mid-Continent Operations Symposium. Oklahoma City, Oklahoma, SPE 52177.
- [115] Sylvester, N. D., (1987). "A Mechanistic Model for Two-Phase Vertical Slug Flowing Pipes." ASME JERT 109: 206-213.
- [116] Tackacs, G., (2001). Considerations on the Selection of an Optimum Vertical Multiphase Pressure Drop Prediction Model for Oil Wells. Production and Operations Symposium. Oklahoma, Paper SPE 68361.
- [117] Taitel, Y., Barnea, D. and Brill, J.P., (1995). "Stratified Three Phase Flow in Pipes." International Journal of Multiphase Flow 21 (1): 53.
- [118] Taitel, Y., Barnea, D. and Dukler, A. E., (1980). "Modeling Flow Pattern Transition for Steady Upward Gas-Liquid Flow in Vertical Tubes." AIChE Journal 26(No. 3): 345-354.
- [119] Taitel, Y. and Duckler, A. E., (1976). "A Model for Predicting Flow Regime Transitions in Horizontal and Near Horizontal Gas-Liquid Flow." AIChE Journal 22(No. 1): 47-55.
- [120] Tamraparni, D and Johnson, T., (2003). Exploratory Data Mining and Data Cleaning. New Jersey, John Wiley & Sons, Inc., Hoboken.
- [121] Ternyik, J., Bilgesu, H. I., Mohaghegh, S. and Rose, D.M., (1995(a)). Virtual Measurement in Pipes: Part 1-Flowing Bottom-Hole Pressure under Multi-Phase Flow and Inclined wellbore Conditions. SPE Eastern Regional Conference and Exhibition. Morgantown, WV-USA, Paper SPE 30975.
- [122] Ternyik, J., Bilgesu, H. and Mohaghegh, S., (1995(b)). Virtual Measurements in Pipes: Part 2-liquid Holdup and Flow Pattern Correlations. SPE Eastern Regional Conference and Exhibition. Morgantown, WV-USA, Paper SPE 30976.

- [123] Verleysen, M. and François, D., (2005). "The Curse of Dimensionality in Data Mining and Time Series Prediction." *Computational Intelligence and Bioinspired Systems*: 758-770.
- [124] Wallis, G., (1969). *One-Dimensional Two Phase Flow*. New York, McGraw-Hill Book Co.
- [125] Wallis, G. and Dobson, J., (1973). "The Onset of Slugging in Horizontal Stratified Air-Water Flow." *International Journal of Multiphase Flow* 1: 173-193.
- [126] Widrow, B., (1962). "Generalization and Information Storage in Networks of Adaline'Neurons'." *Self-Organizing Systems*: 435-461.
- [127] Widrow, B. and Hoff, M., (1960). *Adaptive Switching Circuits*. IRE WESCON Conv. Rev. 4:96-104.
- [128] wikipedia. (2011). "http://en.wikipedia.org/wiki/Confidence_limits." Retrieved January 2 2011.
- [129] Xiao, J., Shoham, O. and Brill, J. P., (1990). *A Comprehensive Mechanistic Model for Two-Phase Flow in Pipelines*. 65th Annual Technical Conference and Exhibition of the Society of Petroleum Engineering. New Orleans, LA, Paper SPE 20631.
- [130] Yuan, H. and Zhou, D., (2008). *Evaluation of Two-Phase Flow Correlations and Mechanistic Models for Pipelines at Inclined Downward Flow*. Eastern Regional/AAPG Eastern Section Joint Meeting. Pittsburgh, Pennsylvania, USA, Paper SPE 117395.
- [131] Zhang, H. Q., Wang, Q., Sarica, C. and Brill, J., (2003(a)). "A unified mechanistic model for slug liquid holdup and transition between slug and dispersed bubble flows." *International Journal of Multiphase Flow* 29: 97-107.

- [132] Zhang, Q., Wang, Q., Sarica, C. and Brill, J., (2003(b)). "Unified Model for Gas-Liquid Pipe Flow via Slug Dynamics - Part 1: Model Development." ASME Journal of Energy Resources Technology 125(266).

LIST OF PUBLICATIONS

Conference Papers

- 1) Ayoub, M., Raja, D. M., and Al-Marhoun. M. A., "Evaluation of Below Bubble Point Viscosity Correlations & Construction of a New Neural Network Model" SPE 108439 – Presented at the 2007 Asia Pacific Oil & Gas Conference and Exhibition held in Jakarta, Indonesia, 30 October-1 November 2007.
- 2) Ayoub, M., and Birol M. D., "The Use of Artificial Neural Networks and Genetic Algorithms for Effectively Optimizing Production from Multiphase Flow Wells" paper, Presented at the 1st National Postgraduate Conference on Engineering Science and Technology, held in UTP, 31 March 2008.
- 3) Ayoub, M., and Birol M. D., "Development of a Universal Artificial Neural Network Model for Pressure Loss Estimation in Pipeline Systems; A comparative Study" paper – Presented at the 1st ICIPEG (International Conference on Integrated Petroleum Engineering and Geosciences) held in Convention Center, Kula Lumpur, Malaysia, 15-17 June 2010.

APPENDIX A

THEORY OF ARTIFICIAL NEURAL NETWORKS AND ABDUCTIVE NETWORKS

A.1 Fundamentals

In this Section, artificial neural network basics will be presented, along with the close relationship between the technology and the biological nervous system. A full mathematical notation of the developed model and the network topology are also provided.

A.1.1 Network Learning

Usually ANN model can be developed using one of three learning paradigms. These are supervised learning, unsupervised learning, and reinforcement learning. During the course of this dissertation the supervised learning will be followed. The network is trained using supervised learning “providing the network with inputs and desired outputs”. The difference between the real outputs and the desired outputs is used by the algorithm to adapt the weights in the network. Fig A.1 illustrates the supervised learning diagram. The net output is calculated and compared with the actual one, if the error between the desired and actual output is within the desired proximity, there will be no weights' changes; otherwise, the error will be back-propagated to adjust the weights between connections (feed backward cycle). After the weights are fixed the feed forward cycle will be utilized for the test set.

The second learning scheme is the unsupervised one where there is no feedback from the environment to indicate if the outputs of the network are correct. The network must discover features, rules, correlations, or classes in the input data by itself. As a matter of fact, for most kinds of unsupervised learning, the targets are the same as inputs.

In other words, unsupervised learning usually performs the same task as an auto-associative network, compressing the information from the input

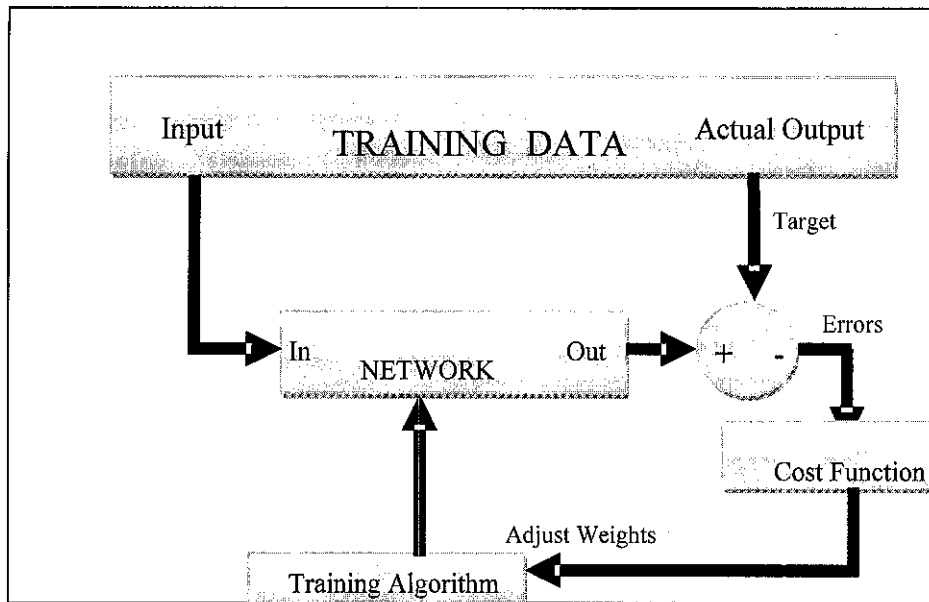


Fig A1: Supervised Learning Model

The third type of network learning is through the reinforcement technique. In this learning paradigm, the purpose is to reward the neuron or parts of a network for good performance, and to penalize the neuron or parts of a network for bad performance.

A.1.2 Network Architecture

Network topology (architecture) is an important feature in designing a successful network. Typically, neurons are arranged in layers, each layer is responsible for performing a certain task.

Based on how interconnections between neurons and layers are; neural network can be divided into two main categories (feed forward and recurrent).

A.1.2.1 Feed forward networks

In these networks the input data sweep directly through hidden layers and finally to the output layer. Hence, it does not allow an internal feedback of information.

The essence of connectivity is primarily related to the fact that every node (neuron) in each layer of the network is connected to every other node in the adjacent forward layer.

The number of neurons in the input layer should be equivalent to the number of input parameters being presented to the network as input. The same thing is correct for output layer, while the function of hidden layer is to intervene between the external input and the network output. Fig A2 is a schematic diagram of a fully connected network with two hidden layer and output layer. The overall response of the network is achieved through the final layer, [Haykin, 1994].

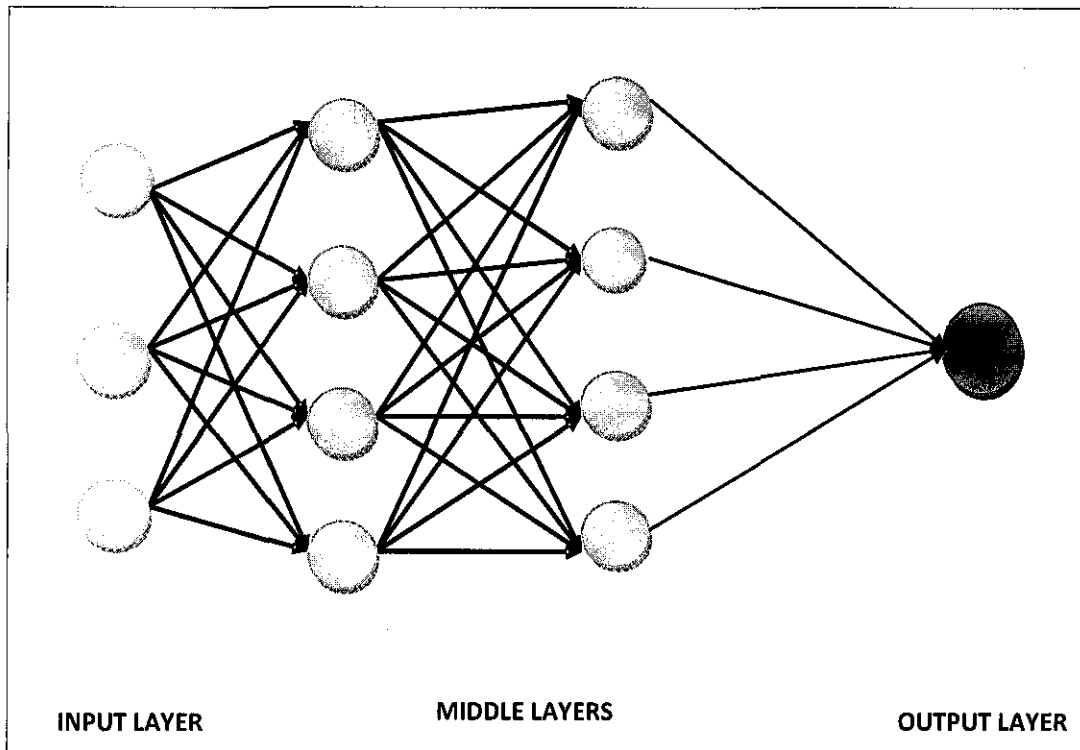


Fig A2: Fully Connected Network with Two Hidden Layers and Output Layer

A.2.2.2 Recurrent networks

Feed-forward networks can be only used for dynamic relationship between input and output variable by including lagged values of input and output variables in the input layer. However, Recurrent Neural Network (RNN) allows for an internal feedback in the system. Internal feedback is a more successful way to account for dynamics in the model. It contains the entire history of inputs as well as outputs, [Haykin, 1994]. Two types of recurrent neural networks are presented here as examples; Jordan Recurrent Neural Network, (JRNN) and Elman Recurrent Neural Network, (ERNN), [Haykin, 1994] and [James and David, 1991].

In JRNN, the output feeds back into the hidden layer with a time delay. The output of the previous periods becomes input in the current period as illustrated in Fig A3. Thus, the current period output carries the history of past outputs, which in turn contains past values of inputs.

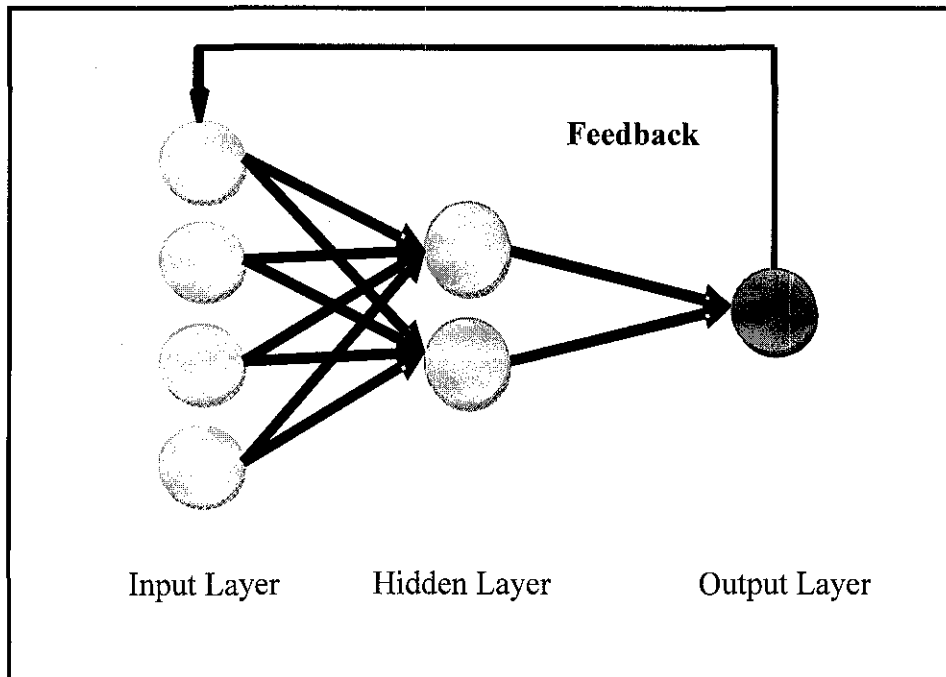


Fig A3: Jordan Recurrent Network

While a two-layer Elman Recurrent Neural Network (ERNN) is depicted in Fig A4. The ERNN accounts for internal feedback in such a way that the hidden layer output feeds back in itself with a time delay before sending signals to the output layer.

RNN, however, requires complex computational processes that can only be performed by more powerful software. The back-propagation algorithm is used during the training process in the computation of estimates of parameters.

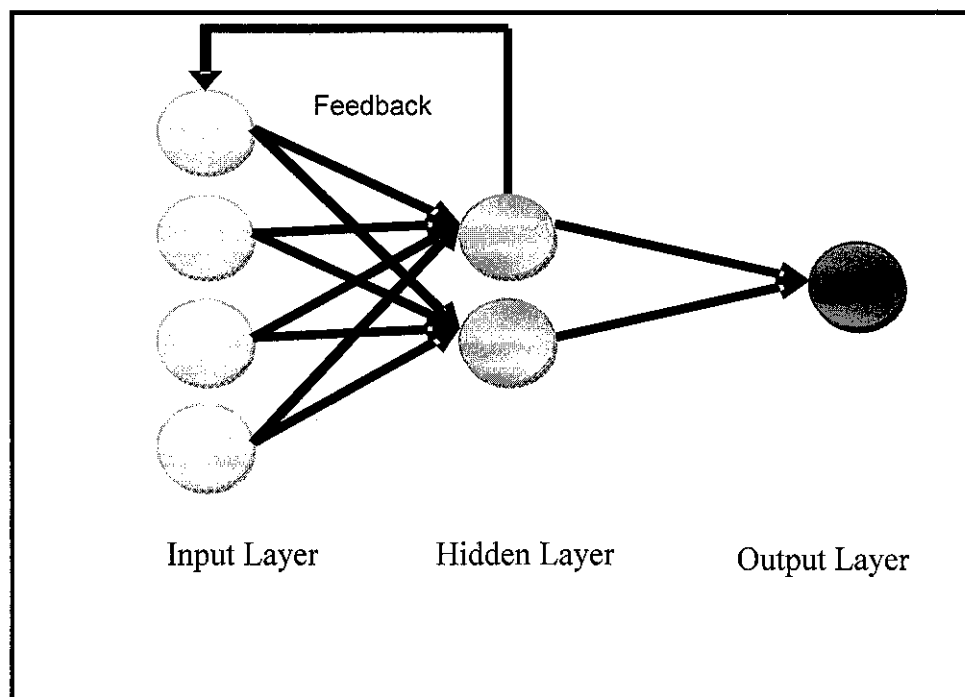


Fig A4: Elman Recurrent Network

A.3 General Network Optimization

Any network should be well optimized in different senses in order to simulate the true physical behavior of the property under study. Certain parameters can be well optimized and rigorously manipulated such as selection of training algorithm, stages, and weight estimation. An unsatisfactory performance of the network can be directly related to an inadequacy of the selected network configuration or when the training algorithm traps in a local minimum or an unsuitable learning set.

In designing network configuration, the main concern is the number of hidden layers and neurons in each layer. Unfortunately, there is no sharp rule defining this feature and how it can be estimated. Trial and error procedure remains the available way to do so, while starting with small number of neurons and hidden layers “and monitoring the performance” may help to resolve this problem efficiently.

Regarding the training algorithms, many algorithms are subjected to trapping in local minima where they stuck on it unless certain design criteria are modified.

The existence of local minima is due to the fact that the error function is the superposition of nonlinear activation functions that may have minima at different points, which sometimes results in a non-convex error function. Using randomly initialized weight and inversion of the algorithm may become a solution for this problem.

The two most frequent problems that often encountered in network designing are the bad or unrepresentative learning set and overtraining. Therefore, selecting global ratios of data division may resolve it through using 2:1:1 or 3:1:1 data set configuration or even 4:1:1 as suggested by some researchers, [Haykin, 1994]. Overtraining refers to the phenomenon when the network starts to model the noise associated with the training data. This phenomenon affects the generalization of network (network is able to accurately generalize when new cases that have not been seen during training are submitted to it). For this reason, cross-validation data are kept aside during training to provide an independent check on the progress of training algorithm. Besides, more confidence is gained where cross-validation data can minimize the error function as training progresses.

A.4 Activation Functions

As described earlier, the four basic elements of the neural network model are; synapses (that may receive a signal), adder (for summing up the input signals, weighted by respective synapses), an activation function, and an externally applied threshold. An activation function that limits (the amplitude of) the output of a neuron within a normalized value in a closed interval, say, between [0, 1] or [-1, 1], (see Fig A5). The activation function *squashes* the output signal in a 'permissible' (amplitude) range. When a neuron updates it passes the sum of the incoming signals through an activation function, or transfer function (linear or nonlinear). A particular transfer function is chosen to satisfy some specification of the problem that the neuron is attempting to solve. In mathematical terms, a neuron j has two equations that can be written as, [Haykin, 1994]:

$$NET_{pj} = \sum_{i=1}^N w_{ji} x_{pi} \tag{A.1}$$

and;

$$y_{pj} = \varphi(NET - \phi_{pj}) \quad (A.2)$$

Where; $x_{p1}, x_{p2}, \dots, x_{pN}$ are the input signals; $w_{j1}, w_{j2}, \dots, w_{jk}$ are the synaptic weights of neuron j ; NET_{pj} is the linear combiner output, ϕ_{pj} is the threshold, φ is the activation function; and y_{pj} is the output signal of the neuron.

Four types of activation functions are identified based on their internal features. A simple threshold function is illustrated by the form;

$$y_{pj} = k(NE_{T_{pj}}) \quad (A.3)$$

Where k is a constant threshold function, i.e.:

$$y_{pj} = 1 \text{ if } (NET_{pj}) > T$$

$$y_{pj} = 0 \text{ otherwise.}$$

T is a constant threshold value, or a function that more accurately simulates the nonlinear transfer characteristics of the biological neuron and permits more general network functions as proposed by their model, [McCulloch and Pitts, 1943]. However, this function is not widely used because it is not differentiable.

The second type of these transfer functions is the Gaussian function, which can be represented by;

$$y_{pj} = ce^{\left(\frac{-NET_{pj}^2}{\sigma^2} \right)} \quad (A.4)$$

Where:

σ is the standard deviation of the function.

The third type is the Sigmoid Function, which is being tried in the present study for its performance as shown by equation 3.5. It applies a certain form of squashing or compressing the range of $(NET)_{pj}$ to a limit that is never exceeded by y_{pj} this function can be represented mathematically by:

$$y_{pj} = \frac{1}{\left(1 + e^{-a \times NET_{pj}}\right)} \quad (A.5)$$

Where;

a is the slope parameter of the sigmoid function.

By varying the slope parameter, different sigmoid function slopes are obtained. Equation 3.6 shows another commonly used activation function, which is the hyperbolic function:

$$y_{pj} = \tanh(x) = \left(\frac{1 - e^{-NET_{pj}}}{1 + e^{-NET_{pj}}} \right) \quad (A.6)$$

This function is symmetrically shaped about the origin and looks like the sigmoid function in shape. However, this function produced good performance when compared to sigmoid function. Hence, it is used as an activation function for the present model. Other functions are presented in Fig A5.

A.5 Back-Propagation Training Algorithm

Is probably the best known, and most widely used learning algorithm for neural networks. It is a gradient based optimization procedure. In this scheme, the network learns a predefined set of input-output sample pairs by using a two-phase propagate-adapt cycle.

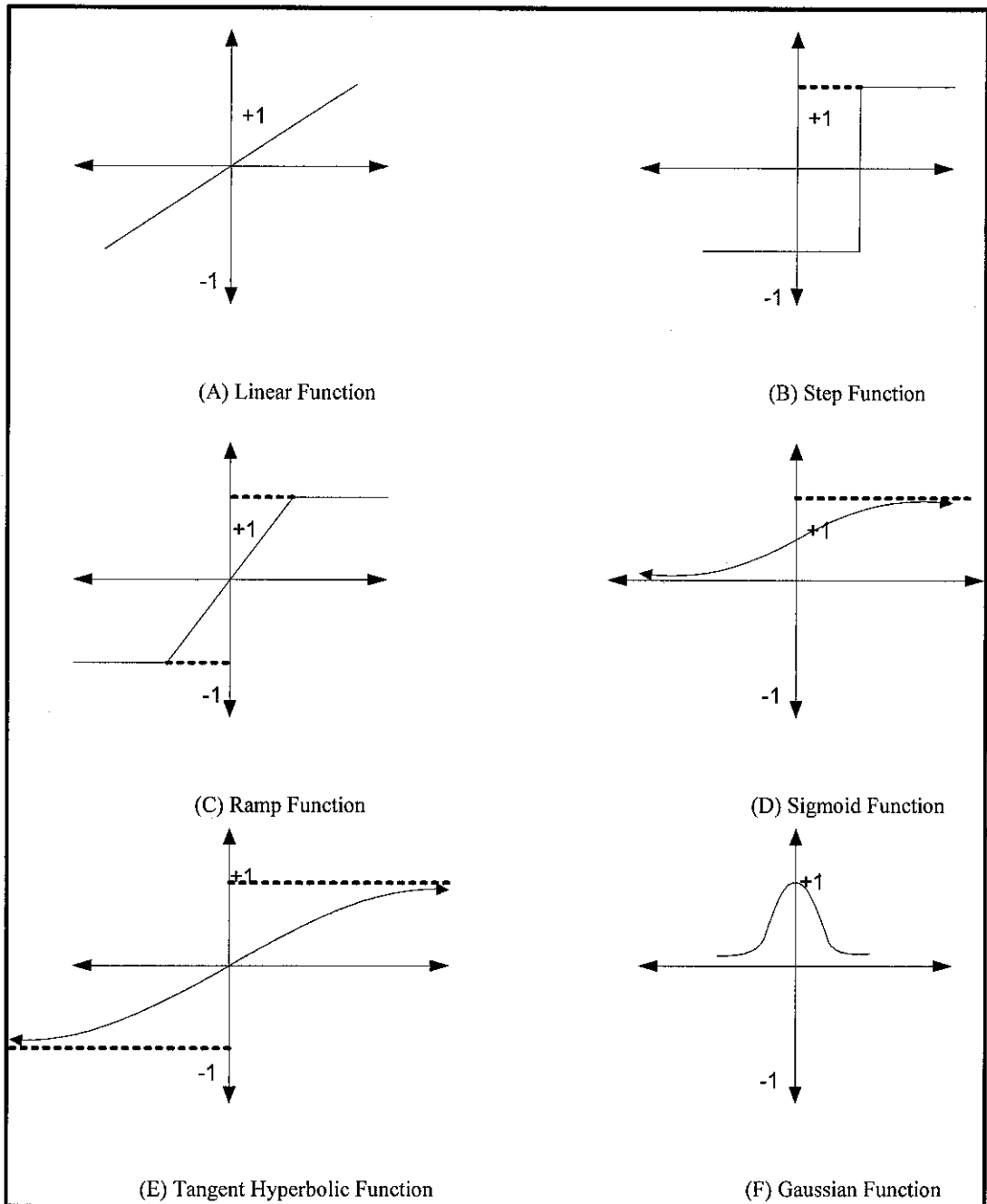


Fig A5: Common Types of Activation Functions, reprinted with permission [Engelbrecht, 2007]

After the input data are provided as stimulus to the first layer of network unit, it is propagated through each upper layer until an output is generated. The latter, is then compared to the desired output, and an error signal is computed for each output unit. Furthermore, the error signals are transmitted backward from the output layer to each node in the hidden layer that mainly contributes directly to the output.

However, each unit in the hidden layer receives only a portion of the total error signal, based roughly on the relative contribution the unit made to the original output. This process repeats layer by layer, until each node in the network has received an error signal that describes its relative contribution to the total error. Based on the error signal received, connection weights are then updated by each unit to cause the network to converge toward a state that allows all the training set to be prearranged. After training, different nodes learn how to recognize different features within the input space. The way of updating the weights connections is done through the generalized delta rule "GDR". A full mathematical notion is presented in the next subsection.

A.6 Generalized Delta Rule

This Section deals with the formal mathematical expression of Back-Propagation Network operation. The learning algorithm, or generalized delta rule, and its derivation will be discussed in details. This derivation is valid for any number of hidden layers.

Suppose the network has an input layer that contains an input vector x_p as shown by;

$$x_p = (x_{p1}, x_{p2}, x_{p3}, \dots, x_{pN}) \quad (\text{A.7})$$

The input units distribute the values to the hidden layer units. The net output to the j^{th} hidden unit is described by:

$$NET_{pj}^h = \sum_{i=1}^N w_{ji}^h x_{pi} + \theta_j^h \quad (\text{A.8})$$

Where;

w_{ji}^h is the weight of the connection from the i^{th} input unit, and

θ_j^h is the bias term

h is a subscript refers to the quantities on the hidden layer.

Assuming that the activation of this node is equal to the net input; then the output of this node is represented by;

$$i_{pj} = f_j^h(NET_{pj}^h) \quad (A.9)$$

The mathematical forms of the output nodes are presented by;

$$NET_{pk}^o = \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o \quad (A.10)$$

$$o_{pk} = f_k^o(NET_{pk}^o) \quad (A.11)$$

Where:

o superscript refers to quantities of the output layer unit.

The basic procedure for training the network is embodied in the following description:

- 1) Apply an input vector to the network and calculate the corresponding output values.
- 2) Compare the actual outputs with the correct outputs and determine a measure of the error.
- 3) Determine in which direction (+ or -) to change each weight in order to reduce the error.
- 4) Determine the amount by which to change each weight.
- 5) Apply the correction to the weights.
- 6) Repeat steps 1 to 5 with all the training vectors until the error for all vectors in the training set is reduced to an acceptable tolerance.

A.6.1 Update of Output-Layer Weights

The general error for the k^{th} input vector can be defined by;

$$\varepsilon_k = (d_k - y_k) \quad (A.12)$$

Where:

d_k = desired output

y_k = actual output

Because the network consists of multiple units in a layer; the error at a single output unit can be defined mathematically by;

$$\delta_{pk} = (y_{pk} - o_{pk}) \quad (\text{A.13})$$

Where;

p subscript refers to the p^{th} training vector

k subscript refers to the k^{th} output unit

So,

y_{pk} = desired output value from the k^{th} unit.

o_{pk} = actual output value from the k^{th} unit.

The error that is minimized by the GDR is the sum of the squares of the errors for all output units as simply shown by;

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (\text{A.14})$$

To determine the direction in which to change the weights, the negative of the gradient of E_p and ∇E_p , with respect to the weights, w_{kj} should be calculated.

The next step is to adjust the values of weights in such a way that the total error is reduced.

From equation (A.14) and the definition of δ_{pk} in equation A.13, each component of ∇E_p can be considered separately as demonstrated by;

$$E_p = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 \quad (\text{A.15})$$

and,

$$\frac{\partial E_p}{\partial w_{kj}^o} = -(y_{pk} - o_{pk}) \frac{\partial f_k^o}{\partial (NET_{pk}^o)} \frac{\partial (NET_{pk}^o)}{\partial w_{kj}^o} \quad (\text{A.16})$$

The chain rule is applied in equation (A.16)

The derivative of f_k^o will be denoted as $f_k^{o'}$

$$\frac{\partial (NET_{pk}^o)}{\partial w_{kj}^o} = \frac{\partial}{\partial w_{kj}^o} \sum_{j=1}^L w_{kj}^o i_{pj} + \theta_k^o = i_{pj} \quad (\text{A.17})$$

Equation A.18 is the result of combining equations (A.16) and (A.17), which yields the negative gradient as follows

$$\frac{-\partial E_p}{\partial w_{kj}^o} = (y_{pk} - o_{pk}) f_k^{o'} (NET_{pk}^o) i_{pj} \quad (\text{A.18})$$

As far as the magnitude of the weight change is concerned, it is proportional to the negative gradient. Thus, the weights on the output layer are updated according to;

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \Delta_p w_{kj}^o(t) \quad (\text{A.19})$$

Where the second term in equation A.19 can be further manipulated by;

$$\Delta_p w_{kj}^o(t) = \eta (y_{pk} - o_{pk}) f_k^{o'} (NET_{pk}^o) i_{pj} \quad (\text{A.20})$$

The factor η is called the learning-rate parameter, ($0 < \eta < 1$).

A.6.2 Output Function

The output function $f_k^o(NET_{jk}^o)$ should be differentiable as suggested previously. This requirement eliminates the possibility of using linear threshold unit since the output function for such a unit is not differentiable at the threshold value. Output function is usually selected as linear function as illustrated by;

$$f_k^o(NET_{jk}^o) = (NET_{jk}^o) \quad (A.21)$$

This defines the linear output unit.

In the first case:

$$f_k^{o'} = 1$$

$$w_{kj}^o(t+1) = w_{kj}^o(t) + \eta(y_{pk} - o_{pk})i_{pj} \quad (A.22)$$

Equation 3.22 can be used for the linear output regardless of the functional form of the output function f_k^o .

A.6.3 Update of Hidden-Layer Weights

The same procedure will be followed to derive the update of the hidden-layer weights. The problem arises when a measure of the error of the outputs of the hidden-layer units is needed. The total error, E_p , must be somehow related to the output values on the hidden layer. To do this, equation A.15 can be used as starting point:

$$E_p = \frac{1}{2} \sum_k (y_{pk} - o_{pk})^2 \quad (A.15)$$

$$E_p = \frac{1}{2} \sum_k (y_{pk} - f_k^o(NET_{pk}^o))^2 \quad (A.23)$$

$$E_p = \frac{1}{2} \sum_k \left(y_{pk} - f_k^o \left(\sum_{j=1} w_{kj}^o i_{pj} + \theta_k^o \right) \right)^2 \quad (A.24)$$

Taking into consideration, i_{pj} depends on the weights of the hidden layer through equations (A.10) and (A.11). This fact can be exploited to calculate the gradient of E_p with respect to the hidden-layer weights, as shown by;

$$\begin{aligned} \frac{\partial E_p}{\partial w_{ji}^h} = & \frac{1}{2} \sum_K \frac{\partial}{\partial w_{ji}^h} (y_{pk} - o_{pk})^2 = \\ & - \sum_K (y_{pk} - o_{pk}) \frac{\partial o_{pk}}{\partial (NET_{pk}^o)} \frac{\partial (NET_{pk}^o)}{\partial i_{pj}} \frac{\partial i_{pj}}{\partial (NET_{pj}^h)} \frac{\partial (NET_{pj}^h)}{\partial w_{ji}^h} \end{aligned} \quad (\text{A.25})$$

Each of the factors in equation (A.25) can be calculated explicitly from the previous equations. The result is summarized by equation A.26 as follows;

$$\frac{\partial E_p}{\partial w_{ji}^h} = - \sum_K (y_{pk} - o_{pk}) f_k^{o'}(NET_{pk}^o) w_{kj}^o f_j^{h'}(NET_{pj}^h) x_{pi} \quad (\text{A.26})$$

A.6.4 Stopping Criteria

Since back-propagation algorithm is a first-order approximation of the steepest-descent technique in the sense that it depends on the gradient of the instantaneous error surface in weight space, weight adjustments can be terminated under certain circumstances, [Haykin, 1994]. Kramer and Sangiovanni (1989) formulated sensible convergence criterion for back-propagation learning, [Kramer and Sangiovanni, 1989]; the back-propagation algorithm is considered to have converged when:

1. The Euclidean norm of the gradient vector reaches a sufficiently small gradient threshold.
2. The absolute rate of change in the average squared error per epoch is sufficiently small.
3. The generalization performance is adequate, or when it is apparent that the generalization performance has peaked.

A.7 Resilient Back-Propagation

A.7.1 Historical Background

The standard back-propagation network follows the gradient descent algorithm. This algorithm is based on the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. To achieve the optimum performance from the standard back-propagation algorithm there are a number of variations that should be considered on basic algorithm. Such variations include utilizing different kinds of training algorithms. Resilient back-propagation is applying one of these variations. The RPROP algorithm was brought into existence by Martin Riedmiller and Heinrich Braun in 1994, [Riedmiller and Braun, 1994]. This algorithm is working under the scheme of local adaptive learning for supervised learning feed-forward neural networks. The main objective of this algorithm is to eliminate the harmful effect of the magnitudes of the partial derivative on the weight step. On contrary to other gradient descent algorithms, which count for the change of magnitude of weight derivative and its sign; this algorithm only counts for the sign of the direction of weight.

Multilayer networks normally use sigmoid transfer functions in the hidden layers. These functions are synonymously called squashing functions, because of their nature in squeezing an infinite input range into a finite output range. Sigmoid functions are distinguished by the fact that their slopes are reaching zero as the input becomes large. This triggers a problem when steepest descent is used to train a multilayer network with sigmoid functions, because the gradient can have a very small magnitude and, therefore, causes small changes in the weights and biases, even though the weights and biases are far from their optimal values, [izmiran, 2010].

A.7.2 General Description of RPROP

The algorithm acts on each weight separately. For each weight, if there is a sign change of the partial derivative of the total error function compared to the last iteration, the update value for that weight is multiplied by a factor η^- , where $0 < \eta^- < 1$. If the last iteration produced the same sign, the update value is multiplied by a factor of η^+ , where $\eta^+ > 1$. The update values are calculated for each weight in the

above manner, and finally each weight is changed by its own update value, in the opposite direction of that weight's partial derivative, so as to minimize the total error function. η^+ is empirically set to 1.2 and η^- to 0.5.

To elaborate the above description mathematically; the individual update-value $\Delta_{ij}(t)$ for each weight w_{ij} will be introduced. This exclusively determines the magnitude of the weight-update. This update value can be expressed mathematically according to the learning rule for each case based on the observed behavior of the partial derivative during two successive weight-steps by the following formula:

$$\Delta_{ij}(t) = \begin{cases} \eta^+ \cdot \Delta_{ij}(t-1), & \text{if } \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) > 0 \\ \eta^- \cdot \Delta_{ij}(t-1), & \text{if } \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \\ \Delta_{ij}(t-1), & \text{else} \end{cases} \quad (\text{A.27})$$

where $0 < \eta^- < 1 < \eta^+$.

A clarification of the adaptation rule based on the above formula can be stated. It is evident that whenever the partial derivative of the equivalent weight w_{ij} varies its sign, which indicates that the last update is large in magnitude and the algorithm has skipped over a local minima, the update-value $\Delta_{ij}(t)$ is decreased by the factor η^- . If the derivative holds its sign, the update-value is to somewhat increased in order to speed up convergence in shallow areas.

When the update-value for each weight is settled in, the weight-update itself tracks a very simple rule: if the derivative is positive, the weight is decreased by its update-value, if the derivative is negative, the update-value is added:

$$\Delta w_{ij}(t) = \begin{cases} -\Delta_{ij}(t), & \text{if } \frac{\partial E}{\partial w_{ij}}(t) > 0 \\ \Delta_{ij}(t), & \text{if } \frac{\partial E}{\partial w_{ij}}(t) < 0 \\ 0, & \text{else} \end{cases} \quad (\text{A.28})$$

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (\text{A.29})$$

However, there is one exception. If the partial derivative changes sign, that is the previous step is too large and the minimum is missed, the previous weight-update is reverted:

$$\Delta w_{ij}(t) = -\Delta w_{ij}(t-1), \quad \text{if } \frac{\partial E}{\partial w_{ij}}(t) \cdot \frac{\partial E}{\partial w_{ij}}(t-1) < 0 \quad (\text{A.30})$$

Due to that ‘backtracking’ weight-step, the derivative is assumed to change its sign once again in the following step. In order to avoid a double penalty of the update-value, there should be no adaptation of the update-value in the succeeding step. In practice this can be done by setting $\frac{\partial E}{\partial w_{ij}}(t-1) = 0$ in the Δ_{ij} update-rule above.

The partial derivative of the total error is given by;

$$\frac{\partial E}{\partial w_{ij}}(t) = \frac{1}{2} \sum_{p=1}^P \frac{\partial E_p}{\partial w_{ij}}(t) \quad (\text{A.31})$$

Hence, the partial derivatives of the errors must be accumulated for all training patterns. This indicates that the weights are updated only after the presentation of all training patterns, [Riedmiller and Braun, 1994].

It is noticed that resilient back-propagation is much faster than the standard steepest descent algorithm. Resilient back-propagation (RPROP) training algorithm was adopted to train the proposed ANN model as mentioned previously.

A.8 Fundamentals and Procedure of GMDH-Based Abductive Networks

The proposed algorithm is based on a multilayer structure using the general form, which is referred to as the Kolmogorov-Gabor polynomial (Volterra functional series)

$$y = a_0 + \sum_{i=1}^m a_i x_i + \sum_{i=1}^m \sum_{j=1}^m a_{ij} x_i x_j + \sum_{i=1}^m \sum_{j=1}^m \sum_{k=1}^m a_{ijk} x_i x_j x_k \dots \quad (\text{A.32})$$

Where; the external input vector is represented by $X = (x_1, x_2 \dots)$, y is the corresponding output value, and a is the vector of weights and coefficients. The polynomial equation represents a full mathematical description. The whole system of equations can be represented using a matrix form as shown below in equation A.33

$$X = \begin{bmatrix} x_{11} & x_{12} & \dots & \dots & x_{1M} \\ x_{21} & x_{22} & \dots & \dots & x_{2M} \\ \dots & \dots & \dots & x_{ij} & x_{iM} \\ \dots & \dots & \dots & \dots & \dots \\ x_{N1} & x_{N2} & \dots & \dots & x_{NM} \end{bmatrix}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ \dots \\ y_{N1} \end{bmatrix} \quad (\text{A.33})$$

Equation (A.32) can be replaced by a system of partial polynomial for the sake of simplicity as shown in equation (A.34)

$$y = a_0 + a_1 x_i + a_2 x_j + a_3 x_i x_j + a_4 x_i^2 + a_5 x_j^2 \quad (\text{A.34})$$

Where $i, j = 1, 2, \dots, M; i \neq j$.

The inductive algorithm follows several systematic steps to finally model the inherent relationship between input parameters and output target, [Madala and Ivakhnenko, 1994].

Data sample of N observations and M independent variables (as presented in equation A.33) corresponding to the system under study is required; the data will be split into training set A and checking set B ($N = N_A + N_B$).

Firstly all the independent variables (matrix of X represented by equation A.33) are taken as pair of two at a time for possible combinations to generate a new

regression polynomial similar to the one presented by equation A.34 where p and q are the columns of the X matrix.

$$y_i = a_{pq} + b_{pq}x_{ip} + c_{pq}x_{iq} + d_{pq}x_{ip}^2 + e_{pq}x_{iq}^2 + f_{pq}x_{ip}x_{iq},$$

$$\begin{cases} p = 1, 2, \dots, M & p \neq q \\ q = 1, 2, \dots, M & p < q \\ i = 1, 2, \dots, N \end{cases} \quad (\text{A.35})$$

A set of coefficients of the regression will be computed for all partial functions by a parameter estimation technique using the training data set A and equation A.35.

The new regression coefficients will be stored into a new matrix C.

$$C = a_{pq} + b_{pq} + c_{pq} + d_{pq} + e_{pq} + f_{pq}, \begin{cases} p = 1, 2, \dots, M & p \neq q \\ q = 1, 2, \dots, M & p > q \\ i = 1, 2, \dots, N \end{cases} \quad (\text{A.36})$$

According to the mathematical law, the number of combinations of input pairs is determined by;

$$\text{number of combinations} = \frac{M(M-1)}{2} \quad (\text{A.37})$$

The polynomial at every N data points will be evaluated to calculate a new estimate called z_{pq} as;

$$z_{i,pq} = a_{pq} + b_{pq}x_{ip} + c_{pq}x_{iq} + d_{pq}x_{ip}^2 + e_{pq}x_{iq}^2 + f_{pq}x_{ip}x_{iq} \quad (\text{A.38})$$

The process will be repeated in an iterative manner until all pairs are evaluated to generate a new regression pairs that will be stored in a new matrix called Z matrix. This new generation of regression pairs can be interpreted as new improved variables that have a better predictability than the original set of data X (presented by equation A.40).

$$Z = \{z_{ij}\}, \begin{cases} i = 1, 2, \dots, N \\ j = 1, 2, \dots, M(M-1)/2 \end{cases} \quad (\text{A.39})$$

$$Z = \begin{bmatrix} z_{11} & z_{12} & \dots & \dots & z_{1, M(M-1)/2} \\ z_{21} & z_{22} & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & z_{ij} & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ z_{N1} & z_{N2} & \dots & \dots & z_{N, M(M-1)/2} \end{bmatrix} \quad (\text{A.40})$$

Quality measures of these functions will be computed according to the objective rule chosen using the testing data set B. This can be done through comparing each column of the new generated matrix Z with the dependent variable y. The external criterion may somewhere be called regularity criterion (root mean squared values) and defined as;

$$r_j^2 = \sum_{i=1}^m \frac{(y_i - z_{ij})^2}{(y_i^2)}, \quad j = 1, 2, \dots, M(M-1)/2 \quad (\text{A.41})$$

The whole procedure is repeated until the regularity criterion is no longer smaller than that of the previous layer. The model of the data can be computed by tracing back the path of the polynomials that corresponds to the lowest mean squared error in each layer.

The best measured function will be chosen as an optimal model. If the final result is not satisfied, F number of partial functions will be chosen which are better than all (this is called "freedom-of-choice") and do further analysis. Schematic diagram of self-organizing GMDH algorithm is depicted in Fig A6

A.9 Types of Abductive networks

A variety of algorithms differ in how they go through partial functions. They are grouped into two types: single-layer and multi-layer algorithms. Combinatorial is the main single-layer algorithm. Multi-layer algorithm is the layered feed-forward algorithm. Harmonic algorithm uses harmonics with non-multiple frequencies and at each level the output errors are fed forward to the next level. Other algorithms like multilevel algorithm are comprised of objective system analysis and two-level,

multiplicative-additive, and multilayer algorithms with error propagations, [Madala and Ivakhnenko, 1994].

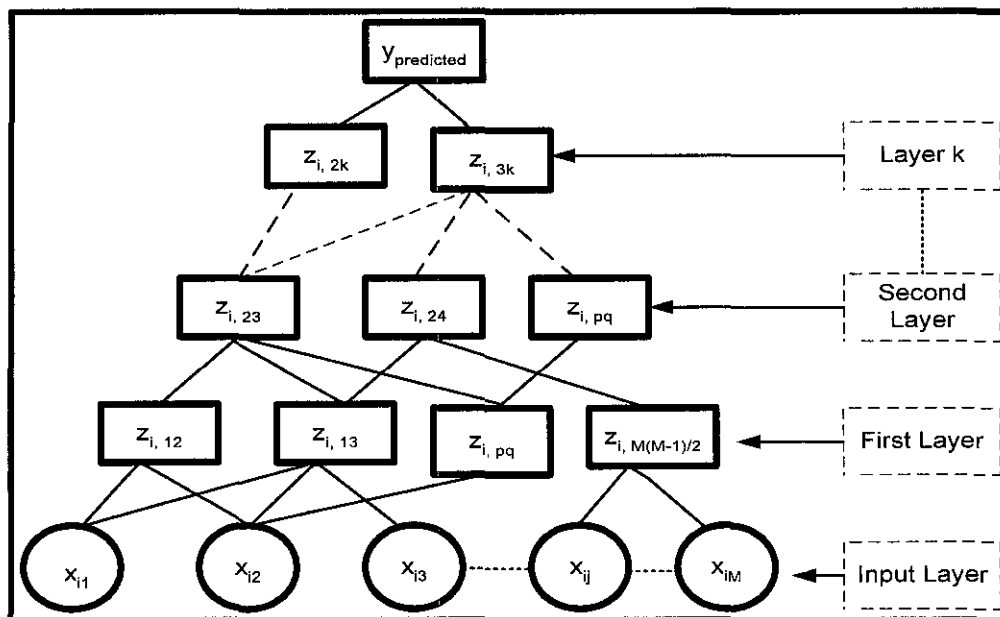


Fig A6: Schematic Diagram of Self-Organizing Algorithm with M Inputs and K Layers

A short description of the multi-layered algorithm will be provided in this Section, which is equivalent to the artificial neural network model. It is synonymously known as polynomial neural network.

A.9.1 Polynomial Neural Network

A.9.1.1 Layer Unit

As presented in the Section of fundamentals and procedure of AIM, the system consists of a sequence of layers and each layer has a group of units connected to the adjacent layer. Each unit has a weight value that is estimated through the application of regularity criterion, or simply minimizing the error by generally applying an external criterion. This measurement will serve two missions. The first one it makes the unit "on" or "off" in comparison with the checking data N_B which is another part of the total data set N . Secondly, it is reflected to attain the optimum output response.

The “on” unit, which is judged to connect to the unit in the next layer, will become a new input for it. The process continues layer after layer in an iterative manner.

A.9.1.2 Multilayer Algorithm

Multilayer network is a parallel bounded structure that is built up based on the type of connection approach given in the basic iterative algorithm with linearized input variables and information in the network flows forward only. Each layer has a number of simulated units depending upon the number of input variables. Two input variables are passed on through each unit (as illustrated in Section A.9).

If there are M input variables, the first layer generates $M_1 (= c_M^2)$ functions. $F_1 (\leq M_1)$ units as per the threshold values are made "on" to the next layer. Outputs of these functions become inputs to the second layer and the same procedure is repeated in the second layer. It is further repeated in successive layers until a global minimum on the error criterion is achieved, [Madala and Ivakhnenko, 1994].

A.9.1.3 Mathematical Description of the System

The system can be described as a system of nonlinear function in its arguments, which may include higher order terms and delayed inputs;

$$y = f(x_1, x_2, \dots, x_1^2, x_2^2, \dots, x_1 x_2, x_1 x_3, \dots, x_{1(-1)}, \dots, x_{2(-1)}, \dots) \quad (\text{A.42})$$

Where; $f()$ is a function of higher degree and y is its estimated output. However, all arguments of x can be calculated as;

$$y = f(u_1, u_2, \dots, u_M) = a_0 + a_1 u_1 + a_2 u_2 + \dots + a_M u_M, \quad (\text{A.43})$$

Where u_i , $i = 1, 2, \dots, M$ are the reconstructed terms of x ; a_k , $k = 0, 1, \dots, M$ are the coefficients and M is total number of arguments. These M input variables become inputs to the first layer, (as illustrated previously). The partial functions generated at this layer can be rewritten as follows:

$$\left[\begin{array}{l} y_1 = v_{01}^{(1)} + v_{11}^{(1)}u_1 + v_{21}^{(1)}u_2, \\ y_2 = v_{01}^{(2)} + v_{11}^{(2)}u_1 + v_{21}^{(2)}u_3, \\ y_3 = v_{01}^{(3)} + v_{11}^{(3)}u_3 + v_{21}^{(3)}u_4, \\ \text{-----} \\ y_{m1} = v_{01}^{(M1)} + v_{11}^{(M1)}u_{(M-1)} + v_{21}^{(M1)}u_M, \end{array} \right] \quad (\text{A.44})$$

Where; $M_1 (= c_M^2)$ is the number of partial functions generated at the first layer, y_j and

$v_{i1}^{(j)}$, $j=1, 2, \dots, M_1$, $i=0, 1, 2$; are the estimated outputs and corresponding coefficients of the functions. Let us assume that F_1 functions are selected for the second layer and that there are $M_2 (= c_{F_1}^2)$ partial functions generated at the second layer. The generated partial function can be formalized as;

$$\left[\begin{array}{l} z_1 = v_{02}^{(1)} + v_{12}^{(1)}y_1 + v_{22}^{(1)}y_2, \\ z_2 = v_{02}^{(2)} + v_{12}^{(2)}y_1 + v_{22}^{(2)}y_3, \\ \text{-----} \\ z_{m2} = v_{02}^{(M2)} + v_{12}^{(M2)}y_{(M-1)} + v_{22}^{(M2)}y_M, \end{array} \right] \quad (\text{A.45})$$

Where; z_j and $v_{i2}^{(j)}$, $j=1, 2, \dots, M_2$, $i=0, 1, 2$ are the estimated outputs and corresponding coefficients of the functions. Following the same trend, assume that F_2 functions are passed on to the third layer; this means that there are $M_3 (= c_{F_2}^2)$ partial functions generated in this layer. The generated partial function can mathematically be expressed as;

$$\left[\begin{array}{l} v_1 = v_{03}^{(1)} + v_{13}^{(1)}z_1 + v_{23}^{(1)}z_2, \\ v_2 = v_{03}^{(2)} + v_{13}^{(2)}z_1 + v_{23}^{(2)}z_3, \\ \text{-----} \\ v_{m3} = v_{03}^{(M3)} + v_{13}^{(M3)}z_{(M-1)} + v_{23}^{(M3)}z_M, \end{array} \right] \quad (\text{A.46})$$

Where; v_j and $v_{i3}^{(j)}$, $j=1,2,\dots,M_3$, $i=0,1,2$ are the estimated outputs and corresponding weights of the functions. The process is repeated by imposing threshold levels of $M \geq F_1 \geq F_2 \geq F_3 \geq \dots \geq F_1$ so that finally a distinctive function is selected at one of the layers. The multilayer network structure with five input arguments and five selected nodes is depicted in Fig A7.

Finally, to get the optimal function in terms of the input arguments, the final model can be traced back as;

$$\begin{aligned}
 v_2 &= f(z_1, z_3) \\
 &\equiv f(f(y_1, y_2), f(y_1, y_4)) \\
 &\equiv f(u_1, u_2, u_3, u_5) = f(X)
 \end{aligned}
 \tag{A.47}$$

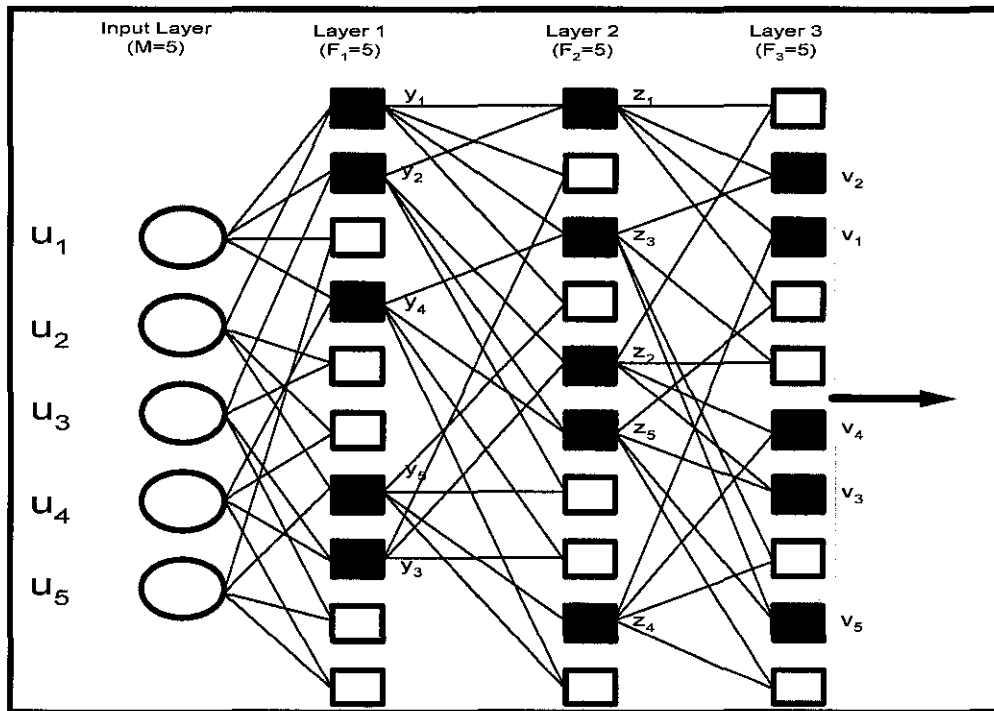


Fig A7: Multilayer Network Structure with Five Input Arguments and Selected Nodes, reprinted with permission [Madala and Ivakhnenko, 1994]

APPENDIX B

PROGRAMS LIST

This Appendix is devoted for the list of programs generated or modified by the author of this thesis.

ANN code: (generated by the author)

```
clc
clf
clear all;%Clears all variables and other classes of data too.
close all;
nntwarnoff;
global net tr
tic
%to reduce the risk of confusing errors.
% Step(1) Processing of the data:
% =====
% Step (2) Reading the input file
% =====
% Loads data and prepares it for a neural network.
%ndata= xlsread('all_data.xls');
ndata= xlsread('CLEAN.xlsx');

%50% of data will be used for training
%25% of data will be used for cross-validation
%25% of data will be used for testing
for i=1:168
    atr(i,:)=ndata(i,:);
end
for i=169:251
    aval(i-168,:)=ndata(i,:);
end
%
for i=252:length(ndata)
    atest(i-251,:)=ndata(i,:);
end

% Step (4) Generating Network structure

% =====
% Resilient Back propagation:
% *****
%
%
```

```

T=atr(:,1);
P=atr(:,2:9);
TV.P=atest(:,2:9);
TV.T=atest(:,1);
VV.T=aval(:,1);
VV.P=aval(:,2:9);
% normalizing the data using mapminmax function ... The function mapminmax
% scales inputs and targets so that they fall in the range [-1,1].
[pn,ps] = mapminmax(P);
[tn,ts] = mapminmax(T);
%
% %normalize val.P manually
vnp=mapminmax('apply',VV.P,ps);
% %normalize val.T manually
vnt=mapminmax('apply',VV.T,ts);
VV.P=vnp;
VV.T=vnt;

% %normalize test.P manually
tnp=mapminmax('apply',TV.P,ps);
%normalize test.T manually
tnt=mapminmax('apply',TV.T,ts);
TV.P=tnp;
TV.T=tnt;
S1=9;% Number of neurons in the first hidden layer
S2=3;
% S3=1;
S4=1;% Number of output variable
net=newff(minmax(pn),[S1 S2 S4],{'logsig''logsig' 'purelin'},'trainrp');

net=init(net);
net.trainParam.epochs = 500;           %Max number of iterations
net.trainParam.goal = 0.0;            %Error tolerance; stopping criterion
net.trainParam.max_fail =6;           %Maximum validation failures
net.trainParam.mem_reduc = 3;         %Factor to use for memory/speed tradeoff
net.trainParam.min_grad = 1e-6;      %Minimum performance gradient
net.trainParam.mu = 0.001;           %Initial Mu
net.trainParam.mu_dec = 0.001;       %Mu decrease factor
net.trainParam.mu_inc = 10;          %Mu increase factor
net.trainParam.mu_max = 1e10;        %Maximum Mu
net.trainParam.show = 5;             %the result is shown at every 5th iteration (epoch)
net.trainParam.time = inf;           %Maximum time to train in seconds
net.trainParam.lr = 0.05;            %Learning rate used in some gradient schemes
net.trainParam.delt_inc = 1.2        %Increment to weight change
net.trainParam.delt_dec = 0.5        %Decrement to weight change
net.trainParam.delta0 = 0.07         %Initial weight change
net.trainParam.deltamax = 70.0       %Maximum weight change

[net,tr]=train(net,pn,tn,[],[],VV,TV);

% Plotting the network error progress for training, testing, and validation data

figure(i+1)

pf1=semilogy(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf);
legend('Training','Validation','Testing',-1);

```

```

ylabel('Mean Squared Error','FontSize',16);
xlabel('Epochs','FontSize',16)
pf=legend('Training','Test','Validation','Location','best');
set(pf,'FontSize',12)
i=i+1;

% Detect whether the net simulates the input data for training data only
% *****
figure
y1=sim(net,pn);
plot(tn,'-r*');
refline(0,0)
hold
plot(y1,':ko');
grid off
set(gcf, 'color', 'white')
title('Simulated network for Pressure Loss "Training Set"')
xlabel('Data Point No')
ylabel('output of network and errors')
legend('Actual Pressure Loss','Predicted Pressure Loss','location','NorthWest')

% checking whether the model simulates the validation data set
% *****
figure
y2=sim(net,VV.P);
plot(VV.T,'-r*');
refline(0,0)
hold
% graphing the simulated network for the output
plot(y2,':ko');
grid off
set(gcf, 'color', 'white')
title('Simulated network for Pressure Loss "Validation Set"')
xlabel('Data Point No')
ylabel('output of network and errors')
legend('Actual Pressure Loss',' Predicted Pressure Loss','location',' Northwest')

% checking whether the model simulates the testing data set
% *****
figure
y3=sim(net,TV.P);
plot(TV.T,'-r*');
refline(0,0)
hold
%Graphing the simulated network for the output (Pressure Loss)
plot(y3,':ko');
grid off
set(gcf, 'color', 'white')
title('Simulated network for Pressure Loss "Testing Set"')
xlabel('Data Point No')
ylabel('output of network and errors')
legend('Actual Pressure Loss', 'Predicted Pressure Loss','location', 'Northwest')
% -----
% % Evaluation of actual and estimated targets
% % -----
% % firstly, for testing set:
% % =====
Pred_tt1=mapminmax('reverse',y3,ts);
Calc_tt1=ndata(252:length(ndata),1);

```

```

% secondly, for validation set:
% =====
Pred_v1=mapminmax('reverse',y2,ts);
Calc_v1=ndata(169:251,1);

% thirdly, for training set:
% =====
Pred_t1=mapminmax('reverse',y1,ts);
Calc_t1=ndata(1:168,1);

% Evaluating Relative Error for training set:
% =====
Et1=(Calc_t1-Pred_t1)./Calc_t1*100;
[q,z] = size(Et1);
figure
plot(Calc_t1,Pred_t1,'o')
grid off
set(gcf, 'color', 'white')
axis tight

title('Predicted Pressure Loss vs. Measured Pressure Loss');
xlabel('Measured Pressure Loss "psig"');
ylabel('Predicted Pressure Loss "psig"')
legend('Training set', 'location', 'Northwest')
% Adding Reference Line with 45 degree slope
refline(1,0)
hold
% Evaluating the correlation coefficient for training set:
% =====
Rt1=corrcoef(Pred_t1,Calc_t1);
Rt11=min(Rt1(:,1));
gtext(['correlation coefficient = (' num2str(Rt11) ')']);
hold

% Adding Reference Line with 45 degree slope
refline(1,0)

% Evaluating Relative Error for validation set:
% =====
Ev1=(Calc_v1-Pred_v1)./Calc_v1*100;
[m,n] = size(Ev1);
figure

plot(Calc_v1,Pred_v1,'o')
grid off
set(gcf, 'color', 'white')
%axis ([1500,3500,1500,3500])
title('Predicted Pressure Loss vs. Measured Pressure Loss');
xlabel('Measured Pressure Loss "psig"');
ylabel('Predicted Pressure Loss "psig"')
legend('Validation set', 'location', 'Northwest')
% Adding Reference Line with 45 degree slope
refline(1,0)

% Evaluating the correlation coefficient for validation set:
% =====
% for the first target Pressure Drop
Rv1=corrcoef(Pred_v1,Calc_v1);

```

```

Rv11=min(Rv1(:,1));
gtext(['correlation coefficient = (' num2str(Rv11) ')']);
hold

% Evaluating Relative Error for testing set:
%=====
% for the first target Pressure Drop
Ett1=(Calc_tt1-Pred_tt1)./Calc_tt1*100;
[m,n] = size(Ett1);
figure
%
plot(Calc_tt1,Pred_tt1,'o')
grid off
set(gcf, 'color', 'white')
axis tight

title('Predicted Pressure Loss vs.Measured Pressure Loss');
xlabel('Measured Pressure Loss "psig"');
ylabel('Predicted Pressure Loss "psig"')
legend('Testing set', 'location', 'Northwest')
% Adding Reference Line with 45 degree slope
refline(1,0)

% Evaluating the correlation coefficient for testing set:
%=====
Rtt1=corrcoef(Pred_tt1,Calc_tt1);
Rtt11=min(Rtt1(:,1));
gtext(['correlation coefficient = (' num2str(Rtt11) ')']);
hold

% plotting the histogram of the errors for training set:
%=====
figure
histfit(Et1,10)
h = findobj(gca,'Type','patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Histogram of Pressure Loss Errors');
legend('Training set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

% plotting the histogram of the errors for validation set:
%=====
figure
histfit(Ev1,10)
h = findobj(gca, 'Type', 'patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Histogram of Pressure Loss Errors');
legend('Validation set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

% plotting the histogram of the errors for testing set:
%=====
figure

```



```

histfit(Ett1,10)
h = findobj(gca,'Type','patch');
set(h,'FaceColor','w','EdgeColor','k')
title('Histogram of Pressure Loss Errors');
legend('Testing set')
xlabel('Error');
ylabel('Frequency')
set(gcf, 'color', 'white')
hold

%
% Estimating the residuals for training set:
% =====
figure
Errorr1 = Pred_t1-Calc_t1;
plot(Errorr1,':ro');
grid off
set(gcf, 'color', 'white')
title('Neural Network Model-Residual Estimation for Pressure Loss')
legend('training set')
xlabel('Data Point No')
ylabel('Errors')
hold
% Estimating the residuals for validation set:
% =====
figure
Errorrv1 = Pred_v1-Calc_v1;
plot(Errorrv1,':ro');
grid off
set(gcf, 'color', 'white')
title('Neural Network Model-Residual Estimation for Pressure Loss')
legend('validation set')
xlabel('Data Point No')
ylabel('Errors')
hold
% Estimating the residuals for testing set:
% =====
figure
Errorrt1 = Pred_tt1-Calc_tt1;
plot(Errorrt1,':ro');
grid off
set(gcf, 'color', 'white')
title('Neural Network Model-Residual Estimation for Pressure Loss')
legend('testing set')
xlabel('Data Point No')
ylabel('Errors')

% *****
% STATISTICAL ANALYSIS:
% *****
% Training set:
% =====
% Determining the Maximum Absolute Percent Relative Error
MaxErrt1 = max(abs(Et1));

% Evaluating the average error
Etagv1 = 1/z*sum(Et1);

% Evaluating the standard deviation

```

```

STDT1 = std(Error1);

% Determining the Minimum Absolute Percent Relative Error
MinErrt1 = min(abs(Et1));

% Evaluating Average Absolute Percent Relative Error
% =====
AAPET1 = sum(abs(Et1))/z;

% Evaluating Average Percent Relative Error
% =====
APET1 = 1/z*sum(Et1);

% Evaluating Root Mean Square
% =====
RMSET1 = sqrt(sum(abs(Et1).^2)/z);

% Validation set:
% =====
% Determining the Maximum Absolute Percent Relative Error
MaxErrv1 = max(abs(Ev1));

% Determining the Minimum Absolute Percent Relative Error
MinErrv1 = min(abs(Ev1));

% Evaluating the average error
Evavg1 = 1/n*sum(Ev1);

% Evaluating the standard deviation
STDV1 = std(Errorv1);

%
% Evaluating Average Absolute Percent Relative Error
% =====
AAPEV1 = sum(abs(Ev1))/n;

% Evaluating Average Percent Relative Error
% =====
APEV1 = 1/n*sum(Ev1);

% Evaluating Root Mean Square
% =====
RMSEV1 = sqrt(sum(abs(Ev1).^2)/n);

% Testing set:
% =====
% Determining the Maximum Absolute Percent Relative Error
MaxErrtt1 = max(abs(Ett1));

% Determining the Minimum Absolute Percent Relative Error
MinErrtt1 = min(abs(Ett1));

% Evaluating the average error
Ettavg1 = 1/n*sum(Ett1);

% Evaluating the standard deviation
STDTT1 = std(Errorrt1);

% Evaluating Average Absolute Percent Relative Error

```

```

% =====
AAPETT1 = sum(abs(Ett1))/n;

% Evaluating Average Percent Relative Error
% =====
APETT1 = 1/n*sum(Ett1);

% Evaluating Root Mean Square
% =====
RMSETT1 = sqrt(sum(abs(Ett1).^2)/n);

%-----
% Simulation: Variation of Gas Flow-rate while fixing the other parameters
% -----Gas Flow-rate variation-----
ps1=[linspace(1078,19024,10);%GAS rate [min=1078 max=19024 mean=7622]
linspace(1527,1527,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(12920.3,12920.3,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(11437,11437,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(44.6,44.6,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(7,7,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160 max=540 mean=321.6]
linspace(133,133,10)]; %WELLHEAD TEMPERATURE [min=63 max=186 mean=134.752]
% normalize data to be simulated using previous minp and maxp
psn1 = mapminmax('apply',ps1,ps);
% Now simulate
ans1 = sim(net,psn1); % Simulate the network using normalized data
as1 = mapminmax('reverse',ans1,ts); % Convert to non-normalized predicted data
% Plot Figures for Gas Flow-rate variation
figure
px1=plot(ps1(1,:),as1(1:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px1,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Gas Flow-Rate (MSCF/d)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)

%-----
% Simulation: Variation of Water Flow-rate while fixing the other parameters
% -----Water Rate variation-----

ps2=[linspace(7622,7622,10);%GAS RATE [min=1078 max=19024 mean=7622]
linspace(0.0,8335,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(12920.3,12920.3,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(11437,11437,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(44.6,44.6,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(10,10,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160max=540 mean=321.6]
linspace(133,133,10)]; %WELLHEAD TEMPERATURE [min=63 max=186 mean=134.752388]
% normalize data to be simulated using anormal
psn2 =mapminmax('apply',ps2,ps);
% Now simulate
ans2 = sim(net,psn2); % Simulate the network using normalized data
as2 = mapminmax('reverse',ans2,ts); % Convert to non-normalized predicted data
% Plot Figures for Water rate variation
figure
px2=plot(ps2(2,:),as2(1:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);

```

```

set(px2,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Water Flow-Rate (bbl/d)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)

%-----
% Simulation: Variation of Oil Flow-rate while fixing the other parameters
% -----Oil Flow-rate variation-----
ps3=[linspace(7622,7622,10);%GAS RATE [min=1078 max=19024 mean=7622]
linspace(1527,1527,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(2200,24800,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(11437,11437,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(44.6,44.6,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(10,10,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160 max=540 mean=321.6]
linspace(133,133,10)];%WELLHEAD TEMPERATURE [min=63 max=186 mean=134.752]

% normalize data to be simulated using previous minp and maxp
psn3 =mapminmax('apply',ps3,ps);
% Now simulate
ans3 = sim(net,psn3); % Simulate the network using normalized data
as3 = mapminmax('reverse',ans3,ts);% Convert to non-normalized predicted data
% Plot Figures for Oil Flowrate variation
figure
px3=plot(ps3(3,:),as3(1,:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px3,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Oil Flow-rate (bbl/D)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)

%-----
% Simulation: Variation of Length of the Pipe while fixing the other parameters
% -----Length of the Pipe variation-----
ps4=[linspace(7622,7622,10);%GAS RATE [min=1078 max=19024 mean=7622]
linspace(1527,1527,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(12920.3,12920.3,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(500,26700,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(44.6,44.6,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(10,10,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160 max=540 mean=321.6]
linspace(133,133,10)];%WELLHEAD TEMPERATURE [min=63 max=186 mean=134.7522388]
%
% normalize data to be simulated using previous minp and maxp
psn4 =mapminmax('apply',ps4,ps);
% Now simulate
ans4 = sim(net,psn4); % Simulate the network using normalized data
as4 = mapminmax('reverse',ans4,ts); % Convert to non-normalized predicted data
% Plot Figures for Length of the Pipe variation
figure
px4=plot(ps4(4,:),as4(1,:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px4,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Length of the Pipe (ft)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)

%-----
% Simulation: Variation of Angle of Deviation while fixing the other parameters
% -----Angle of Deviation variation-----

```

```

ps5=[linspace(7622,7622,10);%GAS RATE [min=1078 max=19024 mean=7622]
linspace(1527,1527,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(12920.3,12920.3,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(11437,11437,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(-52,208,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(10.02,10.02,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160 max=540 mean=321.6]
linspace(133,133,10)]; %WELLHEAD TEMPERATURE [min=63 max=186 mean=134.7522388]
%
% normalize data to be simulated using previous minp and maxp
psn5 =mapminmax('apply',ps5,ps);
% Now simulate
ans5 = sim(net,psn5); % Simulate the network using normalized data
as5 = mapminmax('reverse',ans5,ts);% Convert to non-normalized predicted data
% Plot Figures for Angle of Deviation variation
figure
px5=plot(ps5(5,:),as5(1,:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px5,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Angle of Deviation (Degrees)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)

%-----
% Simulation: Variation of Diameter of the Pipe while fixing the other parameters
% -----Diameter of the Pipe variation-----
ps6=[linspace(7622,7622,10);%GAS RATE [min=1078 max=19024 mean=7622]
linspace(1527,1527,10);%WATER RATE [min=0.0 max=8335 mean=1527]
linspace(12920.3,12920.3,10);%OIL FLOWRATE [min=2200 max=24800 mean=12920.3]
linspace(11437,11437,10);%LENGTH OF THE PIPE [min=500 max=26700 mean=11437]
linspace(-20,-20,10);%ANGLE OF DEVIATION [min=-52 max=208 mean=44.6]
linspace(6.065,10.02,10);%DIAMETER OF THE PIPE [min=6.065 max=10.02 mean=8.6]
linspace(316.8,316.8,10);%WELLHEAD PRESSURE [min=160, max=540 mean=321.6]
linspace(133,133,10)]; %WELLHEAD TEMPERATURE [min=63 max=186mean=134.7522388]
% normalize data to be simulated using previous minp and maxp
psn6 =mapminmax('apply',ps6,ps);
% Now simulate
ans6 = sim(net,psn6); % Simulate the network using normalized data
as6 = mapminmax('reverse',ans6,ts); % Convert to non-normalized predicted data
% Plot Figures for Diameter of the Pipe variation
figure
px6=plot(ps6(6,:),as6(1,:),'-rs');
set(gca,'YGrid','off','XGrid','off')
set(gca,'FontSize',12,'LineWidth',2);
set(px6,'LineStyle','-','LineWidth',1.5,'Color','k','MarkerSize',6)
xlabel('Diameter of the Pipe (Inches)','FontSize',12)
ylabel('Pressure Drop (psia)', 'fontsize',12)
i=i+1;

% Net Parameters:
% * * * * *

% % Evaluating the input weight matrix (from input to hidden layers)
X1=net.IW{1,1};
%
% % Evaluating the first hidden layer's weight matrix (from the first hidden layer to the 2nd one)
X2=net.LW{2,1};
%
% % Evaluating the second hidden layer's weight matrix (from 2nd hidden layer to the 3rd one)

```

```

X3=net.LW{3,2};
%
%% Evaluating the input bias vector
X5=net.b{1};
%
%% Evaluating the first hidden layer's bias vector
X6=net.b{2};
%
%% Evaluating the second hidden layer's bias vector
X7=net.b{3};
%

```

AIM code (polynomial network code):

Function gmdhbuild

```

function [model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons, ...
decNumNeurons, p, critNum, delta, Xv, Yv, verbose)
% GMDHBUILD
% Builds a GMDH-type polynomial neural network using a simple layer-by-layer approach
%
% Call
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, ...
maxNumNeurons, decNumNeurons, p, critNum, delta, Xv, Yv, verbose)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons,
decNumNeurons, p, critNum, delta, Xv, Yv)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons,
decNumNeurons, p, critNum, delta)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons,
decNumNeurons, p, critNum)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons,
decNumNeurons, p)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, maxNumNeurons,
decNumNeurons)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore, ... maxNumNeurons)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs, inputsMore)
%[model, time] = gmdhbuild(Xtr, Ytr, maxNumInputs)
%[model, time] = gmdhbuild(Xtr, Ytr)
%
% Input
% Xtr, Ytr : Training data points (Xtr(i,:), Ytr(i)), i = 1,...,n
% maxNumInputs : Maximum number of inputs for individual neurons - if set to 3, both 2 and 3 inputs
will be tried (default = 2)
% inputsMore : Set to 0 for the neurons to take inputs only from the preceding layer, set to 1 to take
inputs also from the original input variables (default = 1)
% maxNumNeurons: Maximal number of neurons in a layer (default equal to the number of the
original input variables)
% decNumNeurons: In each following layer decrease the number of allowed neurons by
decNumNeurons until the number is equal to 1 (default = 0)
% p : Degree of polynomials in neurons (allowed values are 2 and 3) (default = 2)
% critNum : Criterion for evaluation of neurons and for stopping.
% In each layer only the best neurons (according to the criterion) are retained, and the rest are
% discarded.(default = 2)
% 0 = use validation data (Xv, Yv)
% 1 = use validation data (Xv, Yv) as well as training data
% 2 = use Corrected Akaike's Information Criterion (AICC)
% 3 = use Minimum Description Length (MDL)

```

```

%      Note that both choices 0 and 1 correspond to the so called "regularity criterion".
% delta   : How much lower the criterion value of the network's new layer must be comparing the
%           network's preceding layer
%           (default = 0, which means that new layers will be added as long as the value gets
%           better (smaller))
% Xv, Yv   : Validation data points (Xv(i,:), Yv(i)), i = 1,...,nv (used when critNum is equal to either
%           0 or 1)
% verbose  : Set to 0 for no verbose (default = 1)
%
%      Output
%      model   : GMDH model - a struct with the following elements:
%      numLayers : Number of layers in the network
%      d       : Number of input variables in the training data set
%      maxNumInputs : Maximal number of inputs for neurons
%      inputsMore : See argument "inputsMore"
%      maxNumNeurons : Maximal number of neurons in a layer
%      p       : See argument "p"
%      critNum  : See argument "critNum"
%      layer   : Full information about each layer (number of neurons, indexes of inputs for neurons,
%           matrix of exponents for polynomial, polynomial coefficients)
%           Note that the indexes of inputs are in range [1..d] if an input is one of the
%           original input variables, and in range [d+1..d+maxNumNeurons] if an input is taken
%           from a neuron in the preceding layer.
% time      : Execution time (in seconds)
%
% Please give a reference to the software web page in any publication describing research performed
% using the software, e.g. like this:
% Jekabsons G. GMDH-type Polynomial Neural Networks for Matlab, 2010, available at
% http://www.cs.rtu.lv/jekabsons/
%
% This source code is tested with Matlab version 7.1 (R14SP3).
%
% =====
% GMDH-type polynomial neural network
% Version: 1.4
% Date: March 16, 2010
% Author: Gints Jekabsons (gints.jekabsons@rtu.lv)
% URL: http://www.cs.rtu.lv/jekabsons/
%
% Copyright (C) 2009-2010 Gints Jekabsons
%
% This program is free software: you can redistribute it and/or modify it under the terms of the GNU
% General Public License as published by the Free Software Foundation, either version 2 of the
% License, or (at your option) any later version.
%
% This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY;
% without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR
% PURPOSE. See the GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License along with this program. If
% not, see <http://www.gnu.org/licenses/>.
%
% =====

if nargin < 2
    error('Too few input arguments.');
```

```
end

[n, d] = size(Xtr);
```

```

[ny, dy] = size(Ytr);
if (n < 2) || (d < 2) || (ny ~= n) || (dy ~= 1)
    error('Wrong training data sizes.');
```

end

```

if nargin < 3
    maxNumInputs = 2;
elseif (maxNumInputs ~= 2) && (maxNumInputs ~= 3)
    error('Number of inputs for neurons should be 2 or 3.');
```

end

```

if (d < maxNumInputs)
    error('Number of input variables in the data is lower than the number of inputs for individual
neurons.');
```

end

```

if nargin < 4
    inputsMore = 1;
end
if (nargin < 5) || (maxNumNeurons <= 0)
    maxNumNeurons = d;
end
if maxNumNeurons > d * 2
    error('Too many neurons in a layer. Maximum is two times the number of input variables.');
```

end

```

if maxNumNeurons < 1
    error('Too few neurons in a layer. Minimum is 1.');
```

end

```

if (nargin < 6) || (decNumNeurons < 0)
    decNumNeurons = 0;
end
if nargin < 7
    p = 2;
elseif (p ~= 2) && (p ~= 3)
    error('Degree of individual neurons should be 2 or 3.');
```

end

```

if nargin < 8
    critNum = 2;
end
if any(critNum == [0,1,2,3]) == 0
    error('Only four values for critNum are available (0,1 - use validation data; 2 - AICC; 3 - MDL).');
```

end

```

if nargin < 9
    delta = 0;
end
if (nargin < 11) && (critNum <= 1)
    error('Evaluating the models in validation data requires validation data set.');
```

end

```

if (nargin >= 11) && (critNum <= 1)
    [nv, dv] = size(Xv);
    [nvy, dvy] = size(Yv);
if (nv < 1) || (dv ~= d) || (nvy ~= nv) || (dvy ~= 1)
    error('Wrong validation data sizes.');
```

end

end

```

if nargin < 12
    verbose = 1;
end

ws = warning('off');
if verbose ~= 0
```



```

    fprintf('Building GMDH-type neural network...\n');
end
tic;

if p == 2
    numTermsReal = 6 + 4 * (maxNumInputs == 3); %6 or 10 terms
else
    numTermsReal = 10 + 10 * (maxNumInputs == 3); %10 or 20 terms
end

Xtr(:, d+1:d+maxNumNeurons) = zeros(n, maxNumNeurons);
if critNum <= 1
    Xv(:, d+1:d+maxNumNeurons) = zeros(nv, maxNumNeurons);
end

%start the main loop and create layers
model.numLayers = 0;
while 1

if verbose ~= 0
    fprintf('Building layer #%d...\n', model.numLayers + 1);
end

    layer(model.numLayers + 1).numNeurons = 0;
    modelsTried = 0;
    layer(model.numLayers + 1).coefs = zeros(maxNumNeurons, numTermsReal);

for numInputsTry = maxNumInputs:-1:2

%create matrix of exponents for polynomials
if p == 2
    numTerms = 6 + 4 * (numInputsTry == 3); %6 or 10 terms
if numInputsTry == 2
    r = [0,0;0,1;1,0;1,1;0,2;2,0];
else
    r = [0,0,0;0,0,1;0,1,0;1,0,0;0,1,1;1,0,1;1,1,0;0,0,2;0,2,0;2,0,0];
end
else
    numTerms = 10 + 10 * (numInputsTry == 3); %10 or 20 terms
if numInputsTry == 2
    r = [0,0;0,1;1,0;1,1;0,2;2,0;1,2;2,1;0,3;3,0];
else
    r = [0,0,0;0,0,1;0,1,0;1,0,0;0,1,1;1,0,1;1,1,0;0,0,2;0,2,0;2,0,0; ...
        1,1,1;0,1,2;0,2,1;1,0,2;1,2,0;2,0,1;2,1,0;0,0,3;0,3,0;3,0,0];
end
end

%create matrix of all combinations of inputs for neurons
if model.numLayers == 0
    combs = nchoosek(1:1:d, numInputsTry);
else
if inputsMore == 1
    combs = nchoosek([1:1:d d+1:1:d+layer(model.numLayers).numNeurons], numInputsTry);
else
    combs = nchoosek(d+1:1:d+layer(model.numLayers).numNeurons, numInputsTry);
end
end

%delete all combinations in which none of the inputs are from the preceding layer
if model.numLayers > 0

```

```

        i = 1;
    while i <= size(combs,1)
    if all(combs(i,:) <= d)
        combs(i,:) = [];
    else
        i = i + 1;
    end
    end
    end

    %try all the combinations of inputs for neurons
    for i = 1 : size(combs,1)

        %create matrix for all polynomial terms
        Vals = ones(n, numTerms);
        if critNum <= 1
            Valsv = ones(nv, numTerms);
        end
        for idx = 2 : numTerms
            bf = r(idx, :);
            t = bf > 0;
            tmp = Xtr(:, combs(i,t)) .^ bf(ones(n, 1), t);
            if critNum <= 1
                tmpv = Xv(:, combs(i,t)) .^ bf(ones(nv, 1), t);
            end
            if size(tmp, 2) == 1
                Vals(:, idx) = tmp;
            if critNum <= 1
                Valsv(:, idx) = tmpv;
            end
            else
                Vals(:, idx) = prod(tmp, 2);
            if critNum <= 1
                Valsv(:, idx) = prod(tmpv, 2);
            end
            end
            end

        %calculate coefficients and evaluate the network
        coefs = (Vals' * Vals) \ (Vals' * Ytr);
        modelsTried = modelsTried + 1;
        if ~isnan(coefs(1))
            predY = Vals * coefs;
            if critNum <= 1
                predYv = Valsv * coefs;
            if critNum == 0
                crit = sqrt(mean((predYv - Yv).^2));
            else
                crit = sqrt(mean([(predYv - Yv).^2; (predY - Ytr).^2]));
            end
            else
                comp = complexity(layer, model.numLayers, maxNumNeurons, d, combs(i,:)) +
                size(coefs, 2);
            if critNum == 2 %AICC
            if (n-comp-1 > 0)
                crit = n*log(mean((predY - Ytr).^2)) + 2*comp + 2*comp*(comp+1)/(n-comp-1);
            else
                coefs = NaN;
            end
        end
    end

```

```

else%MDL
    crit = n*log(mean((predY - Ytr).^2)) + comp*log(n);
end
end
end

if ~isnan(coefs(1))
%add the neuron to the layer if
%1) the layer is not full;
%2) the new neuron is better than an existing worst one.
    maxN = maxNumNeurons - model.numLayers * decNumNeurons;
if maxN < 1, maxN = 1; end;
if layer(model.numLayers + 1).numNeurons < maxN
%when the layer is not yet full
if (maxNumInputs == 3) && (numInputsTry == 2)
    layer(model.numLayers + 1).coefs(layer(model.numLayers + 1).numNeurons+1, :) =
[coefs' zeros(1,4+6*(p == 3))];
    layer(model.numLayers + 1).inputs(layer(model.numLayers + 1).numNeurons+1, :) =
[combs(i, :) 0];
else
    layer(model.numLayers + 1).coefs(layer(model.numLayers + 1).numNeurons+1, :) =
coefs;
    layer(model.numLayers + 1).inputs(layer(model.numLayers + 1).numNeurons+1, :) =
combs(i, :);
end
    layer(model.numLayers + 1).comp(layer(model.numLayers + 1).numNeurons+1) =
length(coefs);
    layer(model.numLayers + 1).crit(layer(model.numLayers + 1).numNeurons+1) = crit;
    layer(model.numLayers + 1).terms(layer(model.numLayers + 1).numNeurons+1).r = r;
    Xtr2 = [];
    Xtr2(:, layer(model.numLayers + 1).numNeurons+1) = predY;
if critNum <= 1
    Xv2(:, layer(model.numLayers + 1).numNeurons+1) = predYv;
end
if (layer(model.numLayers + 1).numNeurons == 0) || ...
(layer(model.numLayers + 1).crit(worstOne) < crit)
    worstOne = layer(model.numLayers + 1).numNeurons + 1;
end
    layer(model.numLayers + 1).numNeurons = layer(model.numLayers + 1).numNeurons +
1;
else
%when the layer is already full
if (layer(model.numLayers + 1).crit(worstOne) > crit)
if (maxNumInputs == 3) && (numInputsTry == 2)
    layer(model.numLayers + 1).coefs(worstOne, :) = [coefs' zeros(1,4+6*(p == 3))];
    layer(model.numLayers + 1).inputs(worstOne, :) = [combs(i, :) 0];
else
    layer(model.numLayers + 1).coefs(worstOne, :) = coefs;
    layer(model.numLayers + 1).inputs(worstOne, :) = combs(i, :);
end
    layer(model.numLayers + 1).comp(worstOne) = length(coefs);
    layer(model.numLayers + 1).crit(worstOne) = crit;
    layer(model.numLayers + 1).terms(worstOne).r = r;
    Xtr2(:, worstOne) = predY;
if critNum <= 1
    Xv2(:, worstOne) = predYv;
end
    [dummy, worstOne] = max(layer(model.numLayers + 1).crit);
end
end

```

```

end
end

end

end

if verbose ~= 0
    fprintf('Neurons tried in this layer: %d\n', modelsTried);
    fprintf('Neurons included in this layer: %d\n', layer(model.numLayers + 1).numNeurons);
if critNum <= 1
    fprintf('RMSE in the validation data of the best neuron: %f\n', min(layer(model.numLayers +
1).crit));
else
    fprintf('Criterion value of the best neuron: %f\n', min(layer(model.numLayers + 1).crit));
end
end

%stop the process if there are too few neurons in the new layer
if ((inputsMore == 0) && (layer(model.numLayers + 1).numNeurons < 2)) || ...
    ((inputsMore == 1) && (layer(model.numLayers + 1).numNeurons < 1))
if (layer(model.numLayers + 1).numNeurons > 0)
    model.numLayers = model.numLayers + 1;
end
break
end

%if the network got "better", continue the process
if (layer(model.numLayers + 1).numNeurons > 0) &&...
    ((model.numLayers == 0) || ...
    (min(layer(model.numLayers).crit) - min(layer(model.numLayers + 1).crit) > delta) )
%(min(layer(model.numLayers + 1).crit) < min(layer(model.numLayers).crit)) )
    model.numLayers = model.numLayers + 1;
else
if model.numLayers == 0
    warning(ws);
    error('Failed.');
```

```

end
break
end
```

```

%copy the output values of this layer's neurons to the training
%data matrix
Xtr(:, d+1:d+layer(model.numLayers).numNeurons) = Xtr2;
if critNum <= 1
    Xv(:, d+1:d+layer(model.numLayers).numNeurons) = Xv2;
end
```

```
end
```

```

model.d = d;
model.maxNumInputs = maxNumInputs;
model.inputsMore = inputsMore;
model.maxNumNeurons = maxNumNeurons;
model.p = p;
model.critNum = critNum;
```

```

%only the neurons which are actually used (directly or indirectly) to
%compute the output value may stay in the network
```

```

[dummy best] = min(layer(model.numLayers).crit);
model.layer(model.numLayers).coefs(1,:) = layer(model.numLayers).coefs(best,:);
model.layer(model.numLayers).inputs(1,:) = layer(model.numLayers).inputs(best,:);
model.layer(model.numLayers).terms(1).r = layer(model.numLayers).terms(best).r;
model.layer(model.numLayers).numNeurons = 1;
if model.numLayers > 1
for i = model.numLayers-1:-1:1 %loop through all the layers
    model.layer(i).numNeurons = 0;
for k = 1 : layer(i).numNeurons %loop through all the neurons in this layer
    newNum = 0;
for j = 1 : model.layer(i+1).numNeurons %loop through all the neurons which will stay in the next
layer
for jj = 1 : maxNumInputs %loop through all the inputs
if k == model.layer(i+1).inputs(j,jj) - d
if newNum == 0
        model.layer(i).numNeurons = model.layer(i).numNeurons + 1;
        model.layer(i).coefs(model.layer(i).numNeurons,:) = layer(i).coefs(k,:);
        model.layer(i).inputs(model.layer(i).numNeurons,:) = layer(i).inputs(k,:);
        model.layer(i).terms(model.layer(i).numNeurons).r = layer(i).terms(k).r;
        newNum = model.layer(i).numNeurons + d;
        model.layer(i+1).inputs(j,jj) = newNum;
else
        model.layer(i+1).inputs(j,jj) = newNum;
end
break
end
end
end
end
end
end
end

time = toc;
warning(ws);

if verbose ~= 0
    fprintf('Done.\n');
    used = zeros(d,1);
for i = 1 : model.numLayers
for j = 1 : d
if any(any(model.layer(i).inputs == j))
    used(j) = 1;
end
end
end
    fprintf('Number of layers: %d\n', model.numLayers);
    fprintf('Number of used input variables: %d\n', sum(used));
    fprintf('Execution time: %0.2f seconds\n', time);
end

return

%===== Auxiliary functions =====

function [comp] = complexity(layer, numLayers, maxNumNeurons, d, connections)
%calculates the complexity of the network given output neuron's %connections (it is assumed that the
complexity of a network is equal %to the number of all polynomial terms in all it's neurons which are
%actually connected(directly or indirectly) to network's output)
comp = 0;

```

```

if numLayers == 0
return
end
c = zeros(numLayers, maxNumNeurons);
for i = 1 : numLayers
    c(i, :) = layer(i).comp(:);
end
%{
%unvectorized version:
for j = 1 : length(connections)
    if connections(j) > d
        comp = comp + c(numLayers, connections(j) - d);
        c(numLayers, connections(j) - d) = -1;
    end
end
%}
ind = connections > d;
if any(ind)
    comp = comp + sum(c(numLayers, connections(ind) - d));
    c(numLayers, connections(ind) - d) = -1;
end
%{
%unvectorized version:
for i = numLayers-1:-1:1
    for j = 1 : layer(i).numNeurons
        for k = 1 : layer(i+1).numNeurons
            if (c(i+1, k) == -1) && (c(i, j) > -1) && ...
                any(layer(i+1).inputs(k,:) == j + d)
                comp = comp + c(i, j);
                c(i, j) = -1;
            end
        end
    end
end
end
%}
for i = numLayers-1:-1:1
for k = 1 : layer(i+1).numNeurons
if c(i+1, k) == -1
    inp = layer(i+1).inputs(k,:);
    used = inp > d;
if any(used)
    ind = inp(used) - d;
    ind = ind(c(i, ind) > -1);
if ~isempty(ind)
    comp = comp + sum(c(i, ind));
    c(i, ind) = -1;
end
end
end
end
end
return

```

function gmdhpredict

```

function Yq = gmdhpredict(model, Xq)
% GMDHPREDICT
% Predicts output values for the given query points Xq using a GMDH model
%
```

```

% Call
% [Yq] = gmdhpredict(model, Xq)
%
% Input
% model : GMDH model
% Xq : Inputs of query data points (Xq(i,:)), i = 1,...,nq
%
% Output
% Yq : Predicted outputs of query data points (Yq(i)), i = 1,...,nq
%
% Please give a reference to the software web page in any publication
% describing research performed using the software, e.g. like this:
% Jekabsons G. GMDH-type Polynomial Neural Networks for Matlab, 2010,
% available at http://www.cs.rtu.lv/jekabsons/

% This source code is tested with Matlab version 7.1 (R14SP3).

% =====
% GMDH-type polynomial neural network
% Version: 1.4
% Date: March 16, 2010
% Author: Gints Jekabsons (gints.jekabsons@rtu.lv)
% URL: http://www.cs.rtu.lv/jekabsons/
%
% Copyright (C) 2009-2010 Gints Jekabsons
%
% This program is free software: you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation, either version 2 of the % License, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details.
%
% You should have received a copy of the GNU General Public License
% along with this program. If not, see <http://www.gnu.org/licenses/>.
%
% =====

if nargin < 2
    error('Too few input arguments.');
```

```

%create matrix for all polynomial terms
    numTerms = size(model.layer(i).terms(j).r,1);
    Vals = ones(numTerms,1);
for idx = 2 : numTerms
    bf = model.layer(i).terms(j).r(idx, :);
    t = bf > 0;
    tmp = Xq(q, model.layer(i).inputs(j,t)) .^ bf(1, t);
if size(tmp, 2) == 1
    Vals(idx,1) = tmp;
else
    Vals(idx,1) = prod(tmp, 2);
end
end

%predict output value
    predY = model.layer(i).coefs(j,1:numTerms) * Vals;
if i ~= model.numLayers
%Xq(q, d+j) = predY;
    Xq_tmp(j) = predY;
else
    Yq(q) = predY;
end

end
if i ~= model.numLayers
    Xq(q, d+1:d+model.layer(i).numNeurons) = Xq_tmp;
end
end
end

return

```

function gmdhtest

```

function [MSE, RMSE, RRMSE, R2] = gmdhtest(model, Xtst, Ytst)
% GMDHTEST
% Tests a GMDH-type network model on a test data set (Xtst, Ytst)
%
% Call
% [MSE, RMSE, RRMSE, R2] = gmdhtest(model, Xtst, Ytst)
%
% Input
% model : GMDH model
% Xtst, Ytst: Test data points (Xtst(i,:), Ytst(i)), i = 1,...,ntst
%
% Output
% MSE : Mean Squared Error
% RMSE : Root Mean Squared Error
% RRMSE : Relative Root Mean Squared Error
% R2 : Coefficient of Determination

% Copyright (C) 2009-2010 Gints Jekabsons

if nargin < 3
    error('Too few input arguments.');
```



```

end
if model.d ~= size(Xtst, 2)
    error('The matrix should have the same number of columns as the matrix with which the model was
built.');
```

```

end
MSE = mean((gmdhpredict(model, Xtst) - Ytst) .^ 2);
RMSE = sqrt(MSE);
if size(Ytst, 1) > 1
    RRMSE = RMSE / std(Ytst, 1);
    R2 = 1 - MSE / var(Ytst, 1);
else
    RRMSE = Inf;
    R2 = Inf;
end
return
```

function gmdheq

```

function gmdheq(model, precision)
% gmdheq
% Outputs the equations of a GMDH model.
%
% Call
% gmdheq(model, precision)
% gmdheq(model)
%
% Input
% model      : GMDH-type model
% precision  : Number of digits in the model coefficients
%             (default = 15)
%
% Please give a reference to the software web page in any publication
% describing research performed using the software, e.g. like this:
% Jekabsons G. GMDH-type Polynomial Neural Networks for Matlab, 2010,
% available at http://www.cs.rtu.lv/jekabsons/

% This source code is tested with Matlab version 7.1 (R14SP3).

%
=====
% GMDH-type polynomial neural network
% Version: 1.4
% Date: March 16, 2010
% Author: Gints Jekabsons (gints.jekabsons@rtu.lv)
% URL: http://www.cs.rtu.lv/jekabsons/
%
% Copyright (C) 2009-2010 Gints Jekabsons
%
% This program is free software: you can redistribute it and/or
% modify it under the terms of the GNU General Public License as
% published by the Free Software Foundation, either version 2 of the % License, or (at your option)
% any later version.
%
% This program is distributed in the hope that it will be useful,
% but WITHOUT ANY WARRANTY; without even the implied warranty of
% MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
% GNU General Public License for more details.
%
```

```
% You should have received a copy of the GNU General Public License
% along with this program. If not, see <http://www.gnu.org/licenses/>.
%
```

```

if nargin < 1
    error('Too few input arguments.');
```

```
end
if (nargin < 2) || (isempty(precision))
    precision = 15;
end

if model.numLayers > 0
    p = ['%. ' num2str(precision) 'g'];
    fprintf('Number of layers: %d\n', model.numLayers);
    for i = 1 : model.numLayers %loop through all the layers
        fprintf('Layer #%d\n', i);
        fprintf('Number of neurons: %d\n', model.layer(i).numNeurons);
        for j = 1 : model.layer(i).numNeurons %loop through all the neurons in the ith layer
            [terms inputs] = size(model.layer(i).terms(j).r); %number of terms and inputs
            if (i == model.numLayers)
                str = ['y = ' num2str(model.layer(i).coefs(j,1),p)];
            else
                str = ['x' num2str(j + i*model.d) ' = ' num2str(model.layer(i).coefs(j,1),p)];
            end
            for k = 2 : terms %loop through all the terms
                if model.layer(i).coefs(j,k) >= 0
                    str = [str '+'];
                else
                    str = [str ' '];
                end
                str = [str num2str(model.layer(i).coefs(j,k),p)];
            end
            for kk = 1 : inputs %loop through all the inputs
                if (model.layer(i).terms(j).r(k,kk) > 0)
                    for kkk = 1 : model.layer(i).terms(j).r(k,kk)
                        if (model.layer(i).inputs(j,kk) <= model.d)
                            str = [str '*x' num2str(model.layer(i).inputs(j,kk))];
                        else
                            str = [str '*x' num2str(model.layer(i).inputs(j,kk) + (i-2)*model.d)];
                        end
                    end
                end
            end
            disp(str);
        end
    end
    disp('The network has zero layers.');
```

```
end
return
```

APPENDIX C

GRAPHICAL USER INTERFACE OF ANN MODEL

C.1 Vision and Scope

A graphical user interface (GUI) has been built to allow for easier usage and to implement the pressure drop calculation using an ANN model. The interface has been generated using a user friendly visual basic interfacing program (vb). The interface is capable of displaying data directly into a flexible grid that allows the user to define how many sets of data he would like to insert for prediction. Basically, the interface is easy to be operated and no need for professional knowledge to master it. All data can be entered manually and no need for capturing data from outside source for easier implementation. The GUI is being served as a valuable tool for easy execution of pressure drop estimation at wide range of angles of inclination. In this Appendix, the interfacing of the program is demonstrated step-wisely from data entry and presentation of pressure drop results (both in graphical and tabulated forms).

C.2 Overview

The software is a window-based user interface developed under Visual Basic programming environment. The program consists of different components, (however, description of these components is beyond the scope of this Appendix). The most important feature of this program it can be run independently under any windows-based portable means. Consequently, no further need to install additional components to run it.

C.3 Visual Basic Project Main Interface

For the built vb project, the splash screen (welcome window) is presented in the Fig C1 below:

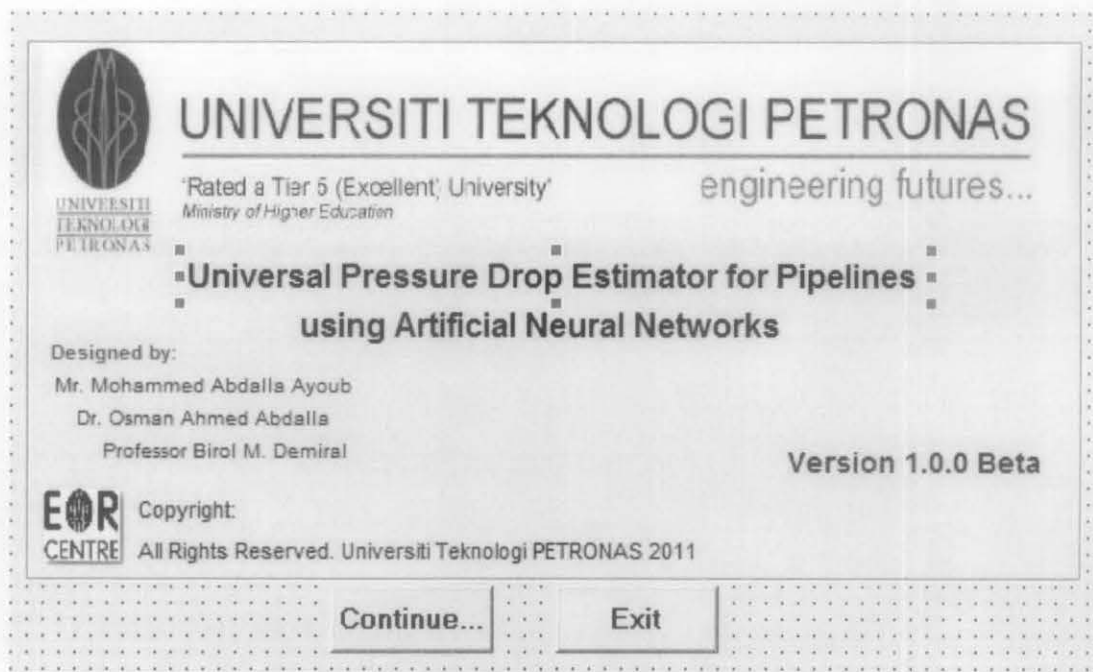


Fig C1: Welcome Screen

The user has the freedom to continue using this program or exit it. Two label buttons of “Continue” and “Exit” are presented at the bottom of splash screen. If the user decided to quit the program a message box will appear to confirm his decision as illustrated in Fig C2 below.

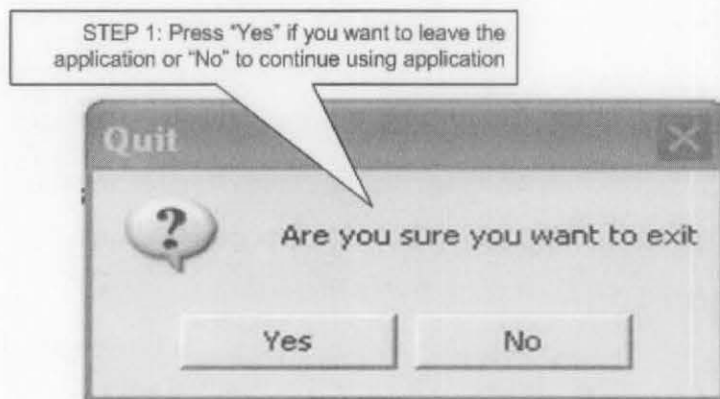


Fig C2: Exit Confirmation Message

Moreover, if the user decided to proceed and use the program a “Continue” button will direct him to the main screen as presented in Fig C3 below.

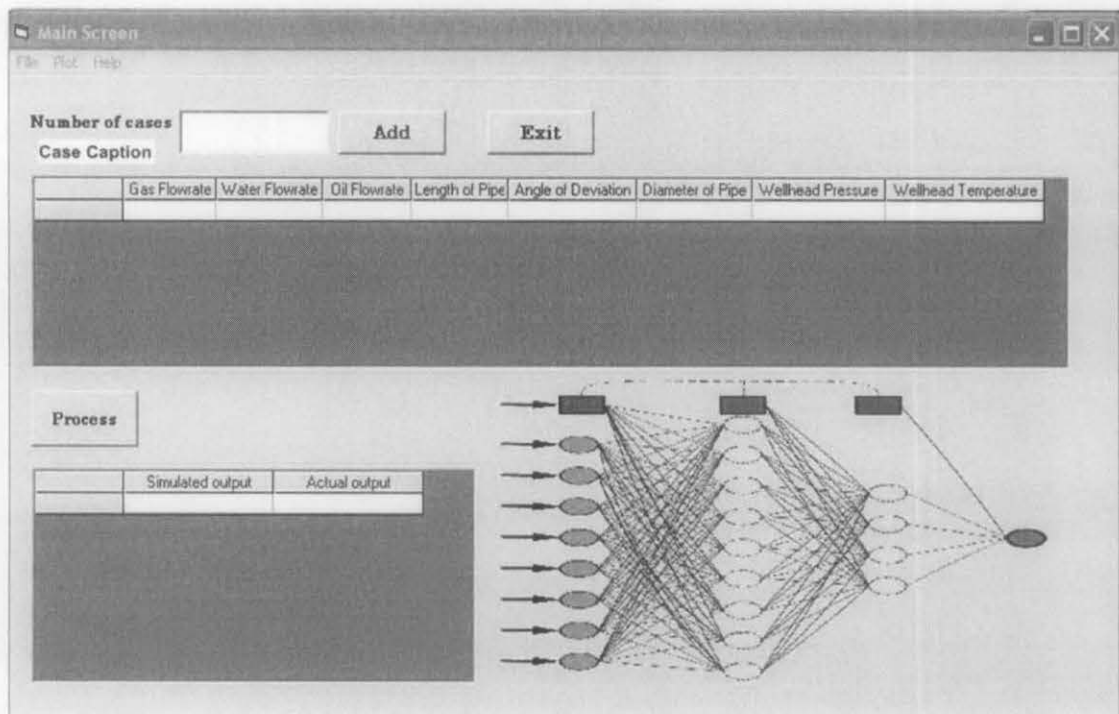


Fig C3: Main Screen

C.3.1 Initial Data Entry

The main screen consists of a text box for the user to enter the number of cases he would like to get prediction for. A user is required to press “**Add**” button in order for the flexible grid to be generated. The user has the ability to add as many data sets (unlimited) and get the corresponding simulated network output. A row of non-editable text consists of variable names is presented. The user has to make sure all units are consistent with the oilfields, e.g; Gas Flow-Rate in MSCFD, Water Flow-Rate, Oil Flow-Rate are in bbl/d, Length of pipe in Ft, angle of Deviation in degrees, Diameter of the Pipe in Inches, Wellhead Pressure in psia, and Wellhead Temperature in Fahrenheit Degrees.

The software will ask for raw data to be entered to proceed to the next step and the user must supply the data as requested in empty cells.

After the process of data entry finished, a user is prompted to hit the “**Process**” Button in order to get the value/values of pressure drop estimation created by the ANN model. However, if the user selected only one case for prediction (single predictor mode) the plot menu will be halted and no graph will be produced by

default. Case caption button may be selected by the user if he wants to highlight certain data set with specific text such “bad data sector starts here”...etc. Moreover, the same case caption will appear in the results of the ANN model for easier follow-up.

C.3.2 Results presentation

The obtained results are presented graphically and in table format for flexible display.

C.3.2.1 Results in Tables

Network output will be presented in a separate column while the user is prompted to enter the values of the actual pressure drop for the sake of comparison and for plotting purposes as shown in Fig C4 below.

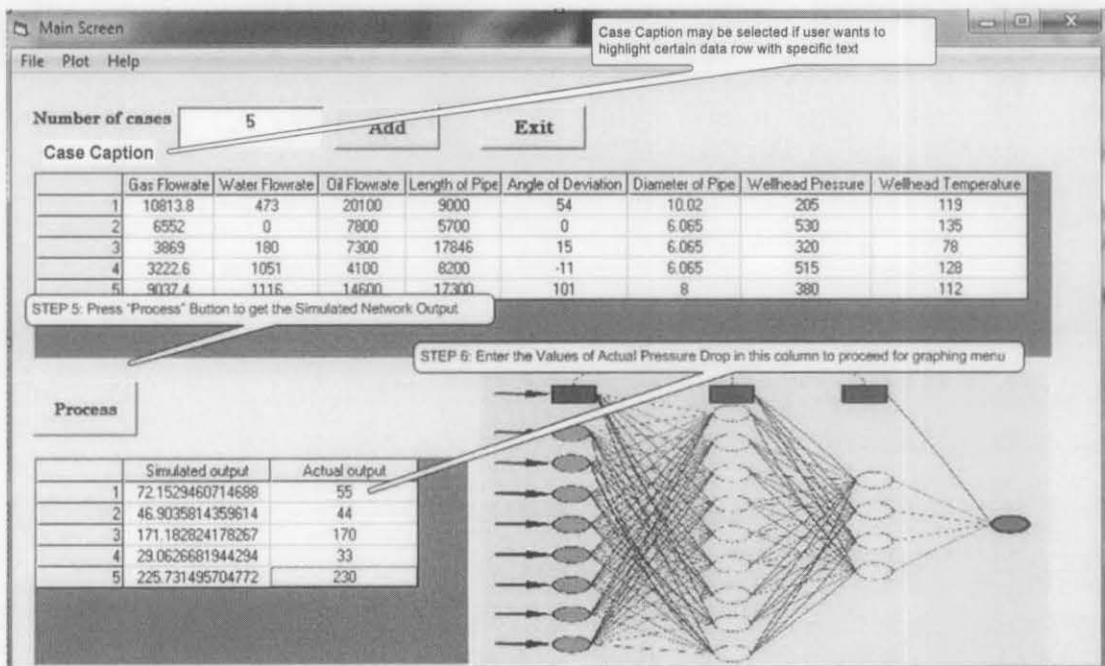


Fig C4: Data Entry Form

C.3.2.2 Graphical Results

The plot menu consists of three types of plots as shown in Fig C5 below. Output plot will be used to graph the network performance in different graphical forms. The

second plot type is cross-plot where predicted pressure drop values will be plotted against actual values. The third type is the residual plot; where the difference between estimated and actual pressure drop will be plotted against each point of interest. Additionally, each plot type is provided with short-cut for quick manipulation. At the lower right corner of the main screen a network topology (architecture) used for generating the graphical user interface has been presented for additional illustration.

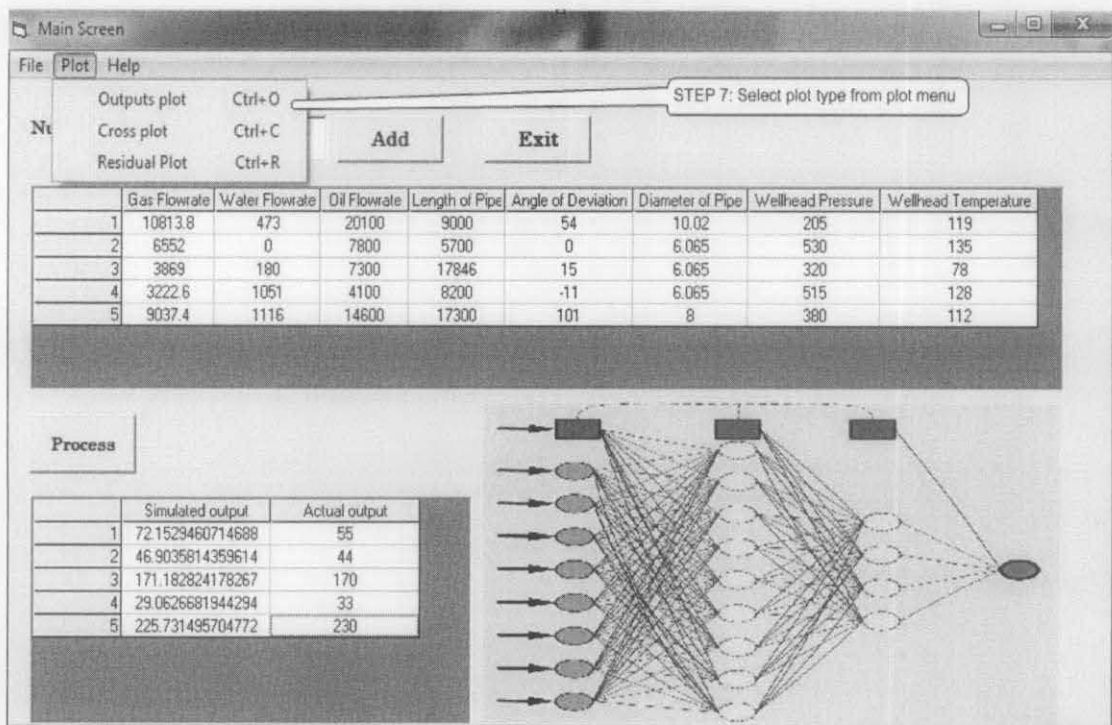


Fig C5: Plot Types

The following figures demonstrate different graphical representations produced by the GUI for different kinds of plot types.

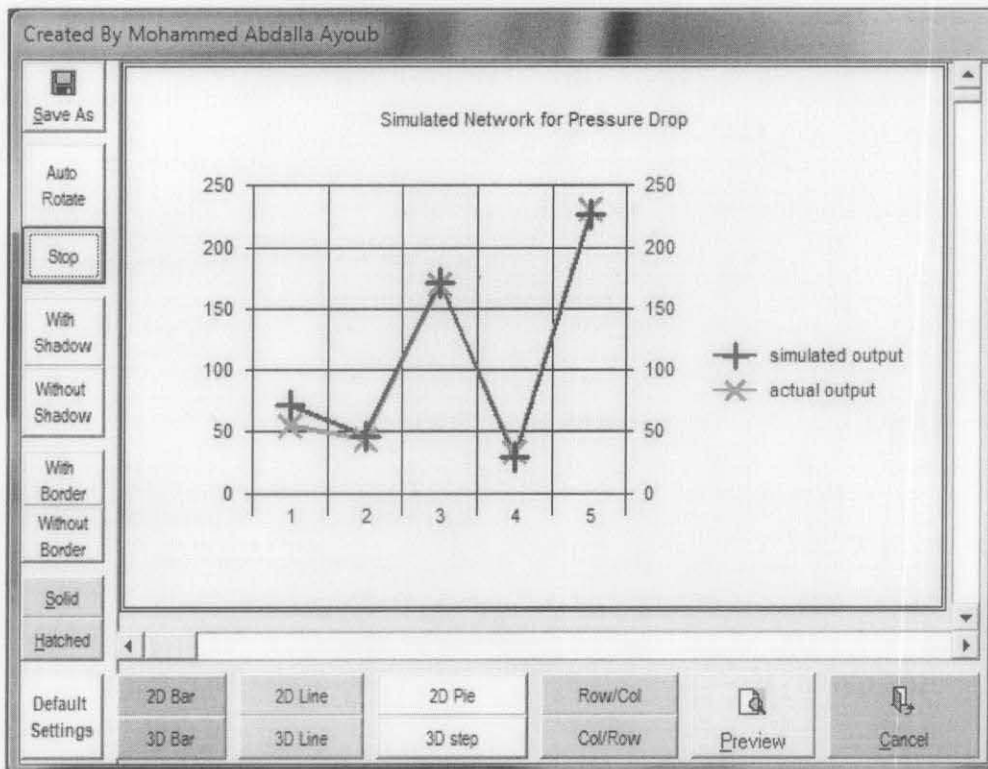


Fig C6: Network Performance Chart

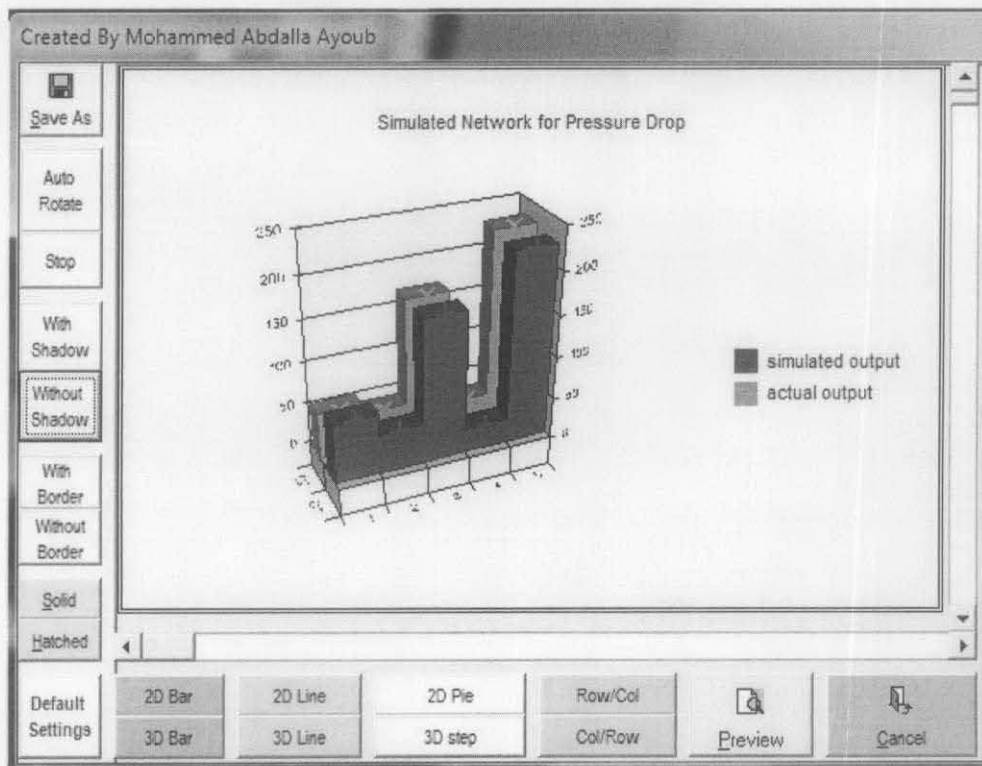


Fig C7: 3-D Network Performance Chart

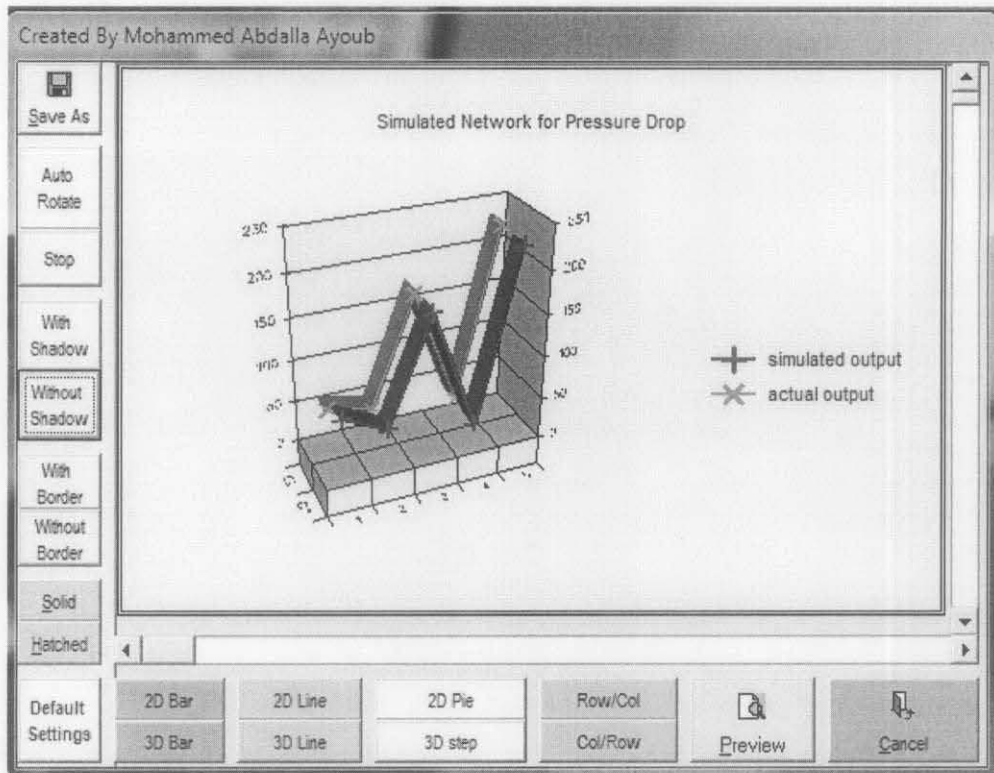


Fig C8: 3-D Step Network Performance Chart

The graph panel is set to become more flexible with many available options for the user to get the required graphing presentations to draw clear conclusion about the simulation results. For instance; the user has the ability to store and save the produced graph in any format into his hard drive. A default setting is also presented to add more flexibility to graphing tool. Overview also is possible for each graph type. In addition, rotational aspect of three dimension figure is also possible with stopping option. Cancellation can be done through cancel button as shown in Fig C9 below.

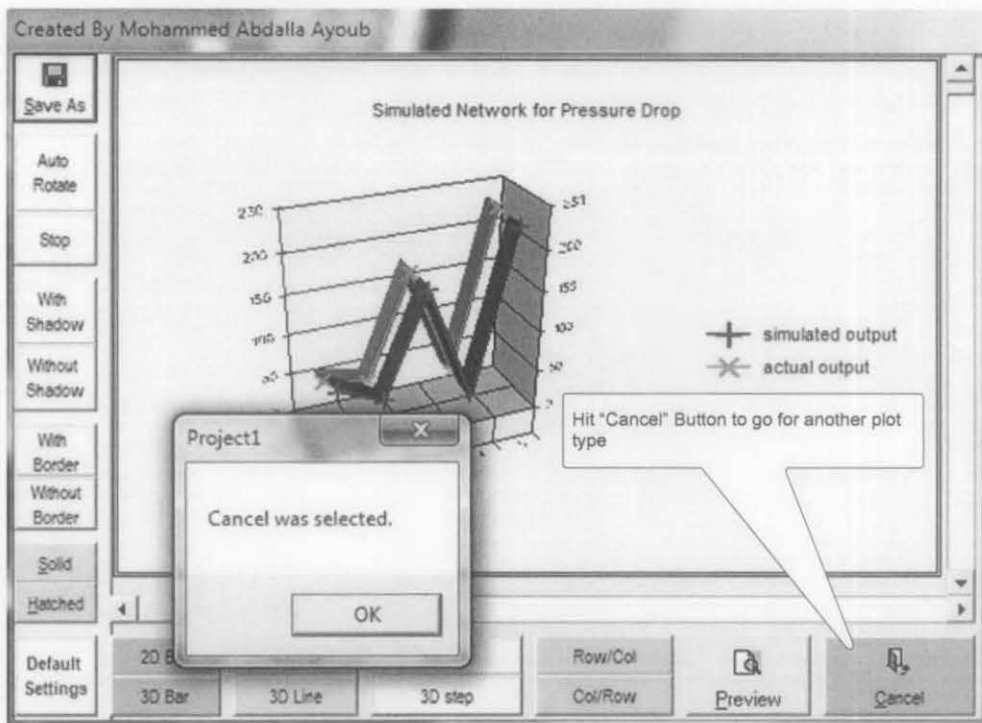


Fig C9: selection of Cancel Button

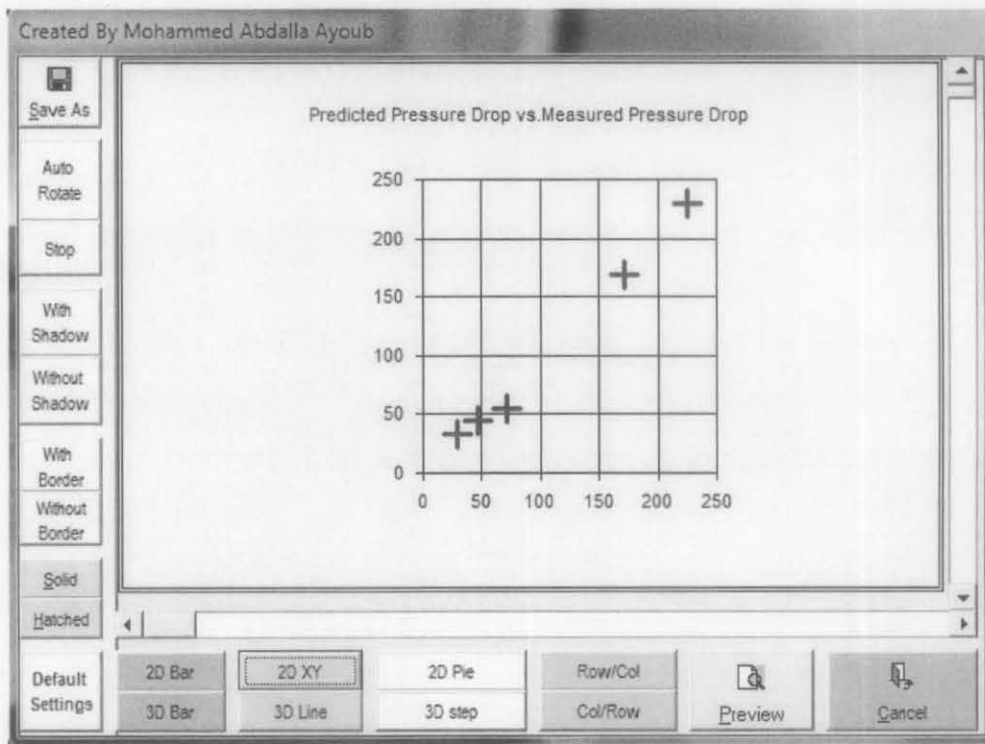


Fig C10: Cross plot Between Estimated and Actual Pressure Drop Values

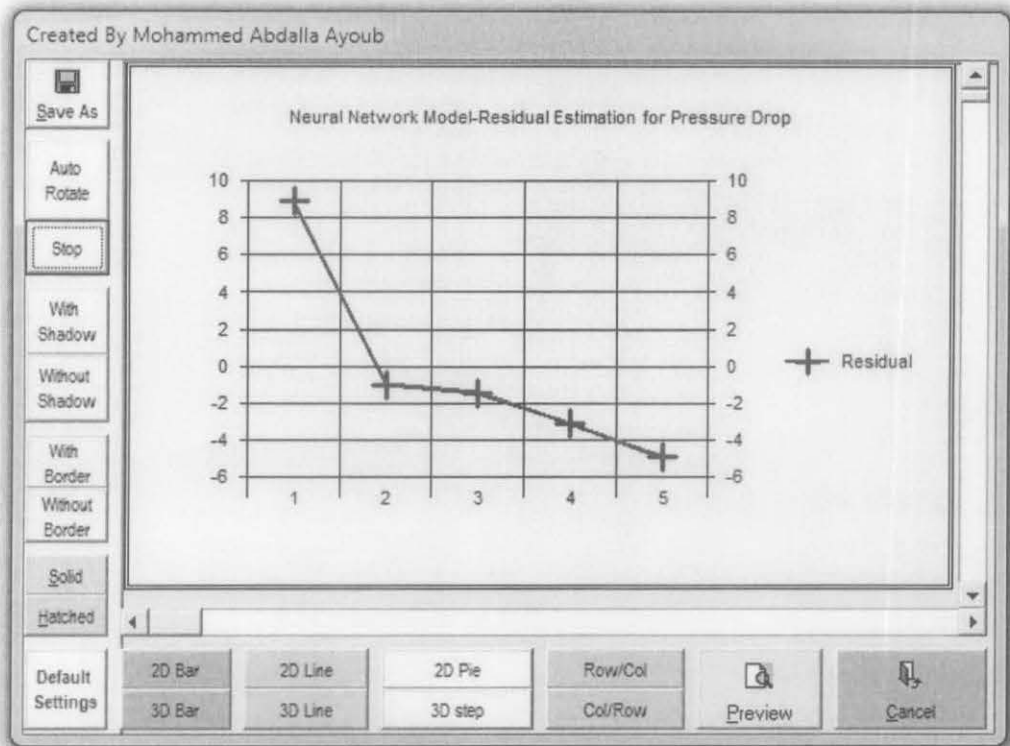


Fig C11: 2-D line Residual Plot

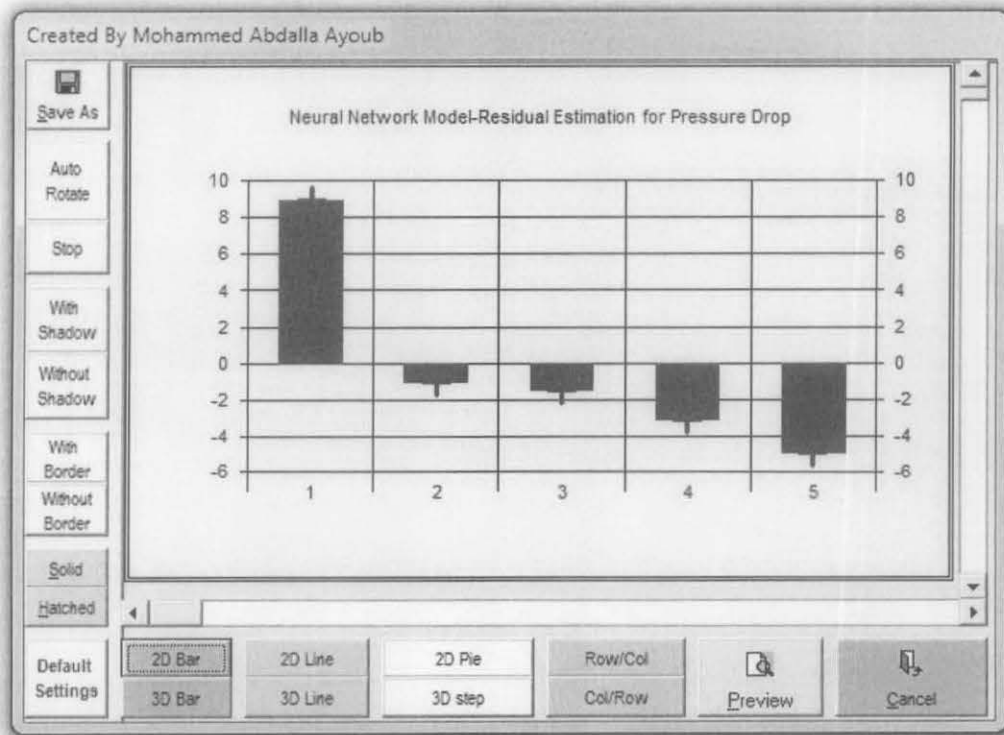


Fig C12: 2-D Bar Residual Plot

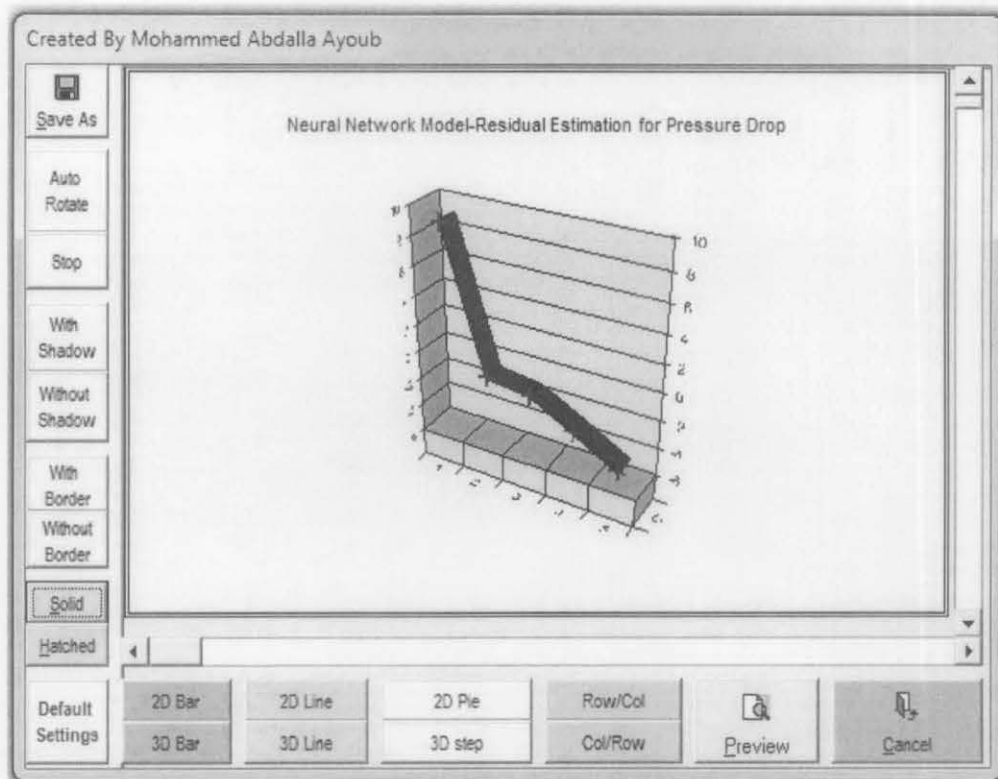


Fig C13: 3-D line Residual Plot

APPENDIX D

RESEARCH DATA

This Appendix is devoted for the range of the research data used for building the assigned ANN and AIM models; Needless to mention that the testing set is being utilized for calculating the pressure drop for all investigated models (Xiao et al. and Gomez et. al) and correlation (Beggs and Brill)

Training Data Range:

Property	Pressure Drop (Psia)	Gas flow-rate (MSCF/D)	Water Flow-rate (Bbl/d)	Oil Flow-rate (Bbl/d)	Length of the Pipe (Ft)	Angle of Inclination (Degrees)	Diameter of Pipe (Inches)	Well-Head Pressure (Psia)	Well-Head Temperature (0F)
Minimum	10	1078	0	2200	500	-52	6.065	160	63
Maximum	240	19024	8335	24800	26700	208	10.02	540	186
Mean	80.6191	7594.57	1523.49	12852.5	11447.4	44.9524	8.6042	322.964	133.756
Standard Deviation	56.5395	3203.1	1952.78	5743.26	6247.44	59.5522	1.7412	133.655	22.0260

Validation Data Range:

Property	Pressure Drop (Psia)	Gas flow-rate (MSCF/D)	Water Flow-rate (Bbl/d)	Oil Flow-rate (Bbl/d)	Length of the Pipe (Ft)	Angle of Inclination (Degrees)	Diameter of Pipe (Inches)	Well-Head Pressure (Psia)	Well-Head Temperature (°F)
Minimum	10	3346.6	0	4400	3600	-13	6.065	160	82
Maximum	250	19278	8424	25000	26700	208	10.02	540	168
Mean	84.120	7384.21	2824.01	13234.4	13590.6	72.927	9.3729	265.710	132.891
Standard Deviation	46.209	3154.73	2377.77	4877.89	7395.66	69.03442	1.14549	92.5294	19.08965

Testing Data Range:

Property	Pressure Drop (Psia)	Gas flow-rate (MSCF/D)	Water Flow-rate (Bbl/d)	Oil Flow-rate (Bbl/d)	Length of the Pipe (Ft)	Angle of Inclination (Degrees)	Diameter of Pipe (Inches)	Well-Head Pressure (Psia)	Well-Head Temperature (°F)
Minimum	20	3239	0	3800	4700	-52	6.065	170	72
Maximum	250	19658.2	8010	22700	25000	128	10.02	545	173
Mean	83.75	7583.855	1336.9	12112.8	10411.1	31.7619	8.31893	354.96	138.5833
Standard Deviation	64.4433	2458.774	2016.5	5105.85	5196.26	46.7587	1.82076	142.02	20.05066

Proposed ANN Model's Weight Matrices

Evaluating the input weight matrix (from input to the first hidden layers)

Property \ Node #	Node #								
	Node-1	Node-2	Node-3	Node-4	Node-5	Node-6	Node-7	Node-8	Node-9
Gas flow-rate (MSCF/D)	1.0343	-1.0365	1.065	1.6953	-0.3292	1.3839	-0.1248	1.3919	2.4669
Water Flow-rate (Bbl/d)	-1.1749	0.0115	3.6286	0.0224	1.4317	0.8349	1.0413	-0.1731	-0.219
Oil Flow-rate (Bbl/d)	-1.017	-2.1898	2.8755	0.0522	-1.1335	1.2097	1.0389	1.3601	4.0701
Length of the Pipe (Ft)	-2.9654	-2.1958	-0.6039	-0.1957	-1.8564	1.3142	1.7503	0.8118	0.0277
Angle of Inclination (Degrees)	-1.2865	-1.5136	0.6805	-2.7552	-0.0055	-1.2735	-0.8582	2.2914	-3.4085
Diameter of Pipe (Inches)	-2.143	0.5539	1.944	1.0459	2.0622	-1.689	-0.9914	-1.3084	0.013
Well-Head Pressure (Psia)	5.3126	1.6488	-42.363	-0.1329	-2.2287	0.1782	0.74	0.3338	1.4531
Well-Head Temperature (°F)	1.8905	0.8211	0.622	-0.1035	-0.2658	-0.1842	1.3821	-1.3303	-0.7986

Evaluating the first hidden layer's weight matrix (from the first hidden layer to the 2nd one)

Evaluating the first hidden layer's weight matrix (from the first hidden layer to the 2nd one)

Node-1	0.9989	-1.8994	-0.7938	1.6057	-2.9226	-0.2931	1.1508	3.2188	0.8182
Node-2	8.7708	-0.5214	-1.2608	2.1613	1.0451	-2.2038	-0.6524	-0.9156	-2.0392
Node-3	9.7687	0.2867	2.5976	-1.1331	-0.4069	-0.2256	-3.2426	-2.9504	-2.0132
Node-4	-10.0494	-2.9247	-0.5774	-2.8685	2.1803	2.2858	1.7217	-2.1561	-0.4904

Evaluating the 2nd hidden layer's weight matrix (from the 2nd hidden layer to the output)

Node-1	Node-2	Node-3	Node-4
0.6951	-0.7911	-0.9162	0.2264

Evaluating the input bias vector

Node-1	Node-2	Node-3	Node-4	Node-5	Node-6	Node-7	Node-8	Node-9
-5.7063	-3.1357	1.4998	-1.2803	1.6987	1.4067	0.4247	0.4056	3.615

Evaluating the first hidden layer's bias vector

Node-1	Node-2	Node-3	Node-4
-0.7413	1.8413	6.0594	0.2204

Evaluating the second hidden layer's bias vector

Node-1
0.5664