Imperial College London

Reservoir Properties Prediction in Carbonate Reservoirs

Nasher M. AlBinHassan

Imperial College London
Department of Earth Science and Engineering
Centre for Reservoir Geophysics

A dissertation submitted in fulfilment of the
requirements for the degree of Doctor of Philosophy

July, 2010

ABSTRACT

Prediction of fluid pressure before drilling, using geophysical methods helps the industry a lot in saving human life, drilling hazards, and equipments. There are several geophysical methods available to predict the fluid pressure before drilling but the most commonly used in the industry are those based on seismic velocities. However, seismic velocities methods are applied on clastic reservoirs with the assumptions that the pressure mechanism is due to mechanical compaction. A major exploration challenge is to successfully predict the presence of high pressure zones in the carbonate reservoirs. Carbonate reservoirs have more complicated internal structure than the clastic reservoirs.

The main objective of this study is to predict the carbonate reservoir properties such as porosity and fluid pressure. The new prediction methods that I used in this thesis are called the artificial intelligent algorithms. These algorithms are better than the conventional Geophysical methods because of their ability to explore complex relationships between the input seismic attributes and the predicted properties. The algorithms include artificial neural networks and group methods of data handling.

Empirical equations from seismic prediction methods were used to transform velocities to fluid pressure. High resolution velocities (waveform tomography) proved that better prediction can be achieved when using better input velocity. The velocity methods preformed a nice prediction when used with clastic seismic data but proved to give unreliable results when used with the carbonate seismic data. This was because of the difficult internal structure of carbonate reservoirs. The neural network methods proved that they are robust in clustering and segmenting the input carbonate seismic data. The usage of more input seismic attributes made the neural network methods better than the conventional velocity methods. Also, this gave the neural network methods more information about the same physical reservoir property. Among the different seismic attributes used in the experiment, seismic inversion and coherence attributes showed good reaction to high pressure zones. Porosity results from the supervised neural network method were used as a guide to the unsupervised neural network method to predict fluid pressure.

The group method of data handling algorithm is performed here for the first time with seismic data to predict the reservoir properties. The new method showed faster and easier prediction than the neural network methods. The automation of the new method yields to better porosity and pore pressure prediction.

This work is dedicated to.....

My beloved parents Muqbel and Fatimah

My sweet heart wife Ebtisam

My lovely kids

Rayan, Muqbel, Rkan

Hala, Hattan, and Razan

ACKNOWLEDGMENTS

The greatest and most righteous thanks go to All Mighty God (Allah) for helping me completing this thesis. I then wish to thanks everyone who contributed to the values of the work described in this thesis. Very special thanks are owed to my supervisor Prof. Yanghua Wang for his great supervision, immense help, constant encouragement and invaluable suggestions throughout the project.

Special thanks to the sponsors of the Centre for Reservoir Geophysics, Imperial College London, for supporting this research. Grateful thanks to Saudi Aramco Oil Company for sponsoring my scholarship and providing the data for the project. Thanks are also due to Saleh AlDossary, Saied AlZahrani, Husam AlMustafa, Sulaiman AlBassam, Yi Lou, Abdulkarim AlZahrani, and Muhammad AlSaggaf and all my friends from Saudi Aramco.

I am grateful to my colleagues in the Imperial College who helped in one way or another. Special thanks to CRG group members for their unparalleled work ethic, against which I could measure myself, and camaraderie, which made daily work that much more enjoyable.

Finally, I reserve the utmost gratitude for my parents, my brothers and sister, and my wife and kids for their unwavering love, continuous encouragement, reassurance and companionship during my PhD.

.

CONTENTS

Abstract ……..……..…… i

Chapter 1. Introduction and overview ……..……..…….. 1

1.1 Terminology of pore pressure ……..…… 1

1.2 Pore pressure prediction techniques and difficulty in carbonate reservoirs 4

1.3 Seismic Attributes ……..… 7

1.4 Artificial Neural Network ……..…… 8

1.5 Group method of data handling (GMDH) 13

1.6 Summary of this thesis……..…… 15

Chapter 2. Carbonate pore pressure cannot be predicted by seismic velocity 18

2.1 Pore pressure prediction using tomographic velocities……...… 18

2.2 A real carbonate seismic data set ……..…… 24

2.3 Geopressure prediction based on velocity ……..…… 28

2.4 Summary ……..…… 30

Chapter 3. Unsupervised artificial neural networks.....32

3.1 Self-organizing map concept……..... 32

3.2 The implementation of the SOM method ……..…… 35

3.3 Clustering the SOM Map ……..….....43

3.4 Application to carbonate data ……..…… 49

3.5 Application of the SOM to other seismic attributes 55

3.6 Summary ……..…… 64

Chapter 4. Supervised artificial neural networks ......…… 66

4.1 Multilayer perceptrons …….....……...…… 67

4.2 Learning, training and back-propagation…….....…… 71

4.3 Data preparations and quality control…….....…… 80

4.4 2D Operation 83

4.5 3D operation (and comparison with OpendTect) 87

4.6 Application result to Saudi Aramco data 96

4.7 Relationship between geopressure and porosity 102

4.8 Summary 105

Chapter 5. The group method of data handling (GMDH) ....…… 106

5.1 Group method of data handling …….....……...…… 107

5.2 Learning and training …….....…… 110

5.3 Enhancements to the GMDH algorithm …….....……..…… 114

5.4 Application of the GMDH algorithm - Synthetic Example.....116

5.5 Clastic reservoir porosity prediction (comparison to ANN).....120

5.6 Carbonate reservoir geopressure prediction 134

5.7 Porosity prediction in carbonate reservoirs.....137

5.8 Summary 142

Chapter 6. Reservoir characterizations of Jilh carbonate formation…..…… 143

6.1 Jilh carbonate formation 143

6.2 Geological interpretation and reservoir characterization 151

6.2.1 Pore pressure prediction.....158

6.2.2 Porosity prediction 162

6.3 Summary..... 169

Chapter 7. Overall conclusions and future works 170

7.1 Summary of each chapter 172

7.3 Future research 175

References ……..……...……177

List of Publications ……..……..…183

LIST OF TABLES

4.1 The job of each layer in the MLP.....67

4.2 Mathematical expressions for some activation functions 70

4.3 My code for random initialization of the weights.....73

4.4 Computation of hidden layer activations 74

4.5 Computation of output layer activations and errors 75

4.6 Back-propagation algorithm and updating the weights 77

4.7 Application code of multilayer perceptrons 81

5.1 Input to the GMDH algorithm. Y is the real output. X_1, X_2, and X_3 are the input variables 118

5.2 Prediction of classical and enhanced GMDH algorithms. Y is the real output, YPC and AEC are the prediction and absolute error of the classical method. YPE and AEE are the prediction and absolute error of the enhanced method 119

5.3 Equations used to predict geopressure 136

5.4 Equations used to predict porosity 139

LIST OF FIGURES

1.1 Pore pressure terminology (modified from Chopra and Huffman, 2006). 3

1.2 Compaction curves for clastic and carbonate materials. (modified from Huffman, 2002).....6

1.3 A Biological Neuron.. 10

1.4 An Artificial Neuron.. 11

1.5 Feed-forward ANN (Blue arrows) and Feed-backward ANN (Red arrows).....12

1.6 Basic structure of the GMDH algorithm.. 14

2.1 Velocity image from using travel-time inversion. (b) Pore-pressure image predicted from the velocity image. 20

2.2 Velocity image obtained from waveform tomography. (b) Pore pressure predicted. 20

2.3 (a) Average predicted pore pressure curves from travel-time tomography (blue) and waveform tomography (red). (b) Resolution curves: The blue curve is from the travel-time tomographic velocity. The red curve is the resolution of pore pressure predicted from the waveform tomographic velocity.. 21

2.4 The frequency spectra of the predicted pore pressures, using 6 different *A* values and a constant *B* value. 22

2.5 Pore Pressure Predicted using well logs (blue curves) compared with pore pressure predicted using seismic inversion (red curves).. 23

2.6 Uthmaniyah portion of the Ghawar field.. 24

2.7 (a) A 3D seismic data set with a horizon picked at the top of the target carbonate reservoir. The display has a seismic amplitude line (left) and an inversion line (right). (b) Well-logs, synthetic seismic trace, and real seismic section. 26
2.8 A horizon slice or time map that covers the top of the target carbonate formation. Wells A, B, and C are located in lines A-A', B-B', and C-C' respectively.. 27
2.9 (a, b) Velocity images inverted from line A-A' and B-B'. (c, d) Geopressure profiles predicted based on the velocity images.. 28
2.10 (a) Velocity image inverted from line C-C'. (b) Geopressure predicted based on the velocity image. The square around Well C does show an over-pressure zone, which should be a *normal pressure zone* revealed in drilling. 29
2.11 (a) Velocity horizon slice extracted from the input inversion cube. (b) Predicted geopressure after applying the Bowers method to the input inversion horizon slice.....30
3.1 A schematic representation of the SOM method, which consists of three layers..33
3.2 The training process of the SOM network. 34
3.3 (a) A model with 3 layers. (b) The count map. (c) The U-matrix map. (d) The magnitude map... 36
3.4 (a) An anticline model. (b) The count map. (c) The U-matrix map. (d) The magnitude map.. 37
3.5 (a) An input real 2D line. (b) The count map. (c) The U-matrix map. (d) The magnitude map.. 38
3.6 (a) A 2D line extracted from the input 3D volume (zone of interest 1.680 ms – 1.705 ms is shown). (b) The count map. (c) The U-matrix map. (d) The magnitude map.. 39
3.7 (a) Original input data. (b) The result of SOM code after applying the code to (a). Note the difference before and after the SOM algorithm, inside the area of interest.....39
3.8 (a) A time slice (1.69 sec) extracted from the original input 3D volume. (b) The same time slice in (a) after applying the SOM code.. 40
3.9 The input 3D seismic data with the target horizon posted. The pink dots are the locations of the randomly selected input vectors. 41
3.10 The final clusters of OpendTect neural network algorithm... 42
3.11 The final clusters of my in-house SOM code.. 42

3.12 Clustering process 44
3.13 The k-means clustering process. (a) Random pick of clusters (4 cluster centres picked). (b) Association of data points to clusters. (d) Moving to new cluster location. (d) Association again. Process will stop when clusters are no longer moving 46
3.14 Curve produced by examining the objective function in k-means clustering.....46
3.15 (a) K-means clustering gives membership values (u_{ji} = 0 or 1) between all data points and cluster number II. (b) FCM clustering gives smoother values for the membership with cluster number II 48
3.16 (a) The target points are located directly in the target horizon. The SOM method uses both the red and blue points as the input for geopressure prediction. (b) Prediction of geopressure after using the SOM to cluster the impedance data. Note that this is not the final clustering and the SOM will be clustered again. (c) A crossplot showing the nonlinear relationship between the pore pressure and seismic impedance using 19 wells. 50
3.17 (a) different clusters inside the SOM lattice with colours representing each cluster. (b) And (c) show the representative traces for each of the four clusters 52
3.18 (a) Clustering of SOM using 2 clusters. (b) Clustering using 3 clusters. (c) Clustering using 4 clusters. (d) Clustering using 5 clusters. (e) Clustering using 6 clusters. (f) Clustering using 7 clusters. (g) Clustering using 8 clusters. (h) Clustering using 9 clusters. 54
3.19 Curve produced by examining the objective function in k-means clustering with the carbonate data as an input. 55
3.20 (a) Final clustering result of the k-means algorithm. (b) Final clustering result of the FCM algorithm 56
3.21 (a) A horizon slice from the input seismic amplitude volume. (b) Final clustering after using the seismic amplitude as an input to the SOM code. Solid dots are wells with high pressure. Hollow dots are wells with normal pressure. 57
3.22 (a) A horizon slice from the input seismic impedance volume. (b) Final clustering after using the seismic impedance as an input to the SOM code 57
3.23 (a) A horizon slice from the input instantaneous amplitude volume. (b) Final clustering after using the instantaneous amplitude as an input to the SOM code 58

3.24 (a) A horizon slice from the input frequency weighted envelope volume. (b) Final clustering after using the frequency weighted envelope as an input to the SOM code. ...58
3.25 (a) A horizon slice from the input thin-bed volume. (b) Final clustering after using the thin-bed as an input to the SOM code 59
3.26 (a) A horizon slice from the input attenuation (1/Q) volume. (b) Final clustering after using the attenuation (1/Q) as an input to the SOM code 59
3.27 (a) A horizon slice from the input coherence of seismic amplitude volume. (b) Final clustering after using the coherence of seismic amplitude as an input to the SOM code. 60
3.28 (a) A horizon slice from the input curvature of seismic impedance volume. (b) Final clustering after using the curvature of seismic impedance as an input to the SOM code. 60
3.29 A graph for the statistics of overpressured areas under each seismic attribute. Colours are clusters (1-4). 62
3.30 A graph for the statistics of normal pressured areas under each seismic attribute. Colours are clusters (1-4). 63
4.1 An artificial neuron (perceptron). The processing element is composed of summation which creates a weighted sum of all inputs and activation which is a non-linear operator to transfer the summation into a desired output. 68
4.2 Four activation functions. (A) and (C) are producing a data limit between -1 and +1. (B) The linear function that does not change the summation values. (D) Sigmoid function that squashed the infinite range of input into the range 0 to +1. 69
4.3 General topology of the multilayer perceptron. The input data are normalized for better data handling and then sent to the input layer. The weights between the input and hidden layers and between hidden and output layers are trained using a back-propagation method. Each perceptron is composed of two operations: summation and activation…..71
4.4 Learning process in the multilayer perceptron networks. The neural network output is compared to the desired output and the back-propagation will adjust the network weights. 72

4.5 (a) A good generalization with proper non-linear interpolation between the examples (hollow circles). (b) A poor generalization with over-fitted nonlinear mapping. 79
4.6 2D trained operator used for geopressure prediction. The network is trained using values of geopressure indication from wells. One data sample of geopressure is used from each well 84
4.7 Geopressure prediction based on seismic impedance. (a) The result of unsupervised neural network SOM. (b) The result of supervised neural network multilayer perceptions. Solid dots indicate a well location with an over-pressure and the hollow dot indicates a well with normal pressure. (c) cross-plot of predicted pressure using supervised neural network MLP vs. pressure data (mud weights) at 17 wells……85
4.8 2D trained operator used for porosity prediction. The network is trained using values of porosity from porosity logs 86
4.9 (a) Geopressure prediction using a supervised neural network. (b) Porosity prediction using a supervised neural network.. 87
4.10 3D trained operator used with multilayer perceptrons. The 2D slice is now moving to cover 74 horizon slices. 88
4.11 Workflow of supervised neural network used by OpendTect software. (modified after (Aminzadeh and Groot, 2006) 89
4.12 (a) Input acoustic impedance used by OpendTect. (b) Predicted porosity of OpendTect multilayer neural network 90
4.13 Error verses learning parameter alpha. During coding, the learning rate is changed many times to come up with the best and stable learning rate that produce the lowest network error 91
4.14 Error verses number of nodes in the hidden layer. As the number of hidden nodes increase, the error is decreases. This is not always good because using too many hidden nodes will misguide the network and over-fitting will happen 91
4.15 Comparison of original porosity log (black), porosity predicted by OpendTect (blue), and my porosity prediction (red). (b) Over-fitting or overtraining problem. (c) Final acceptable result 93
4.16 (a) Acoustic impedance input data. (b) Porosity prediction using OpendTect. ...94

4.17 (a) My porosity prediction with over-fitting. (b) My porosity prediction of the final result. 95
4.18 3D view of seismic amplitude data in the background and the in-house code result of porosity in the foreground 96
4.19 Seismic acoustic impedance profile. This 2D line is extracted from the 3D volume to show the impedance events and layers 97
4.20 The original porosity (black curves) and predicted porosity (red curves) after using 3 wells (a, b, and c) as training wells 98
4.21 3 wells to test the porosity prediction after using 3 training wells. The original porosity (black curves) and predicted porosity (red curves).Note the miss-prediction of shallow values of well (a). 99
4.22 The original porosity (black curves) and predicted porosity (red curves) after using 7 wells (a to g) as training wells 99
4.23 3 wells to test the porosity prediction after using 7 training wells. The original porosity (black curves) and predicted porosity (red curves). 100
4.24 (a) Predicted porosity profile. This is the porosity output when I used the seismic impedance 2D line from figure 4.19. (b) A plot of impedance trace and porosity trace at one well location. (c) A crossplot showing the nonlinear relationship between the porosity and seismic impedance using 19 wells 101
4.25 A 3D view showing one impedance cross-line (right) and one inline that has both porosity and seismic amplitude (left). 102
4.26 (*a*) The pressure prediction after applying the SOM to the predicted porosity. (*b*) The prediction picture after applying the SOM to the attributes, including the impedance and the porosity estimate. (c) A crossplot showing the nonlinear relationship between the porosity and pore pressure values using 19 wells 104
5.1 Basic scheme of the GMDH topology. Each pair of input data forms a regression equation. The outputs of the regression equations form new inputs to the next layer. The final output consists of selected regression equations from all layers 109
5.2 Basic propagation scheme of GMDH algorithm. Layer one is showing the relation between two input pares as regression equations. Layer two is showing the relation between the variables from layer one 110

5.3 Stopping criterion of GMDH. (Edited after Farlow 1984a) 112
5.4 The predicted square error PSE. KP is increasing and FSE is decreasing with increasing complexity of the model. (Edited after (Montgomery and Drake, 1990)). ...116
5.5 A comparison between the classical and the enhanced GMDH methods. Y is the real output, YPC and AEC are the prediction and absolute error of the classical method. YPE and AEE are the prediction and absolute error of the enhanced method 120
5.6 (a) The Effect of changing CPM values on the values of FSE, KP, and PSE. The horizontal axis is the CPM values Iused for a real data set, and the vertical axis is the actual values for FSE, KP and PSE obtained from tests corresponding to different CPM. (b) The average PSE (black circle) for each layer (after using CPM = 0.1). The lowest PSE is at layer 2, where the training stops. (c) The effect of changing CPM values (0.1-0.3) on the values of PSE 123
5.7 Predicted square errors for layer one. The vertical index *i* and the horizontal index *j* indicates that the output in the first layer is the combination of the *i*th and the *j*th vectors of the original input. (a) 45 relationships in layer one. (b) Ten relationships with the lowest predicted square errors at layer one. These 10 outputs will be used as the input to layer two 125
5.8 Predicted square errors for layer two. The vertical index *i* and the horizontal index *j* indicates that the output in the second layer is the combination of the *i*th and *j*th vectors in layer one. (a) 45 relationships in layer two. (b) 20 relationships with the predicted square errors less than 0.35×10^{-3} from layer two..... 126
5.9 (a) 20 predicted porosity results based on 20 relationships from layer two. (b) The median value curve (in red) of those 20 outputs (as the final prediction), which is compared with the true porosity curve (in black). (c-d) Comparison between the prediction and true curves from the other two more porosity logs. The correlation coefficients between two curves in (b), (c) and (d) are 0.95, 0.50, and 0.83 respectively 128
5.10 The standard deviations and the predicted square errors of 20 relations with lowest PSEs from layer two. As all of these 20 outputs have very small PSEs and deviations, the median values of them are selected to form the final prediction 129

5.11 The structure of the GMDH network for porosity prediction. The input consists of 10 seismic attributes, and the output is the median filtered result of 20 combinations in layer two; in this way GMDH effectively uses all of the 10 seismic attributes in the prediction process 130
5.12 Cross section of a porosity logs. Black curve is the original porosity log. Dotted red curve is extracted from the porosity volume that was predicted using supervised ANN or (MLP).Red curve is the predicted porosity extracted from the GMDH porosity volume. 131
5.13 (a) A 2D line extracted from the original seismic amplitude volume. (b) The same 2d line extracted from the porosity volume that was predicted using MLP algorithm. (c) The porosity section, predicted using the GMDH algorithm 133
5.14 The final structure of the GMDH network for predicting geopressure. Explanations of N, F, and U are given in Table 5.3... 136
5.15 Geopressure prediction by using the GMDH algorithm 137
5.16 The final structure of the GMDH network for predicting porosity. Explanations of N, F, and U are in table 5.4.... 139
5.17 Cross section of 7 porosity logs used during training. Black curves are the original porosity logs and red curves are the predicted porosity extracted from the final porosity volume 141
5.18 5.13 Cross section of 3 testing porosity logs that were never used during training. Black curves are the original porosity logs and red curves are the predicted porosity extracted from the final porosity volume 141
5.19 2d line extracted from the final porosity volume 142
6.1 Uthmaniyah portion of the Ghawar field. The fields are located east of Saudi Arabia 144
6.2 (a) Stratigraphic column of Minjur, Jilh, and Sudair formations. (b) The colour index or legend of (a) which is showing the lithology type at each formation. Khuff formation is an important gas producing zone which is under Sudair formation 145
6.3 (a) A top view map of the study area with a horizon picked on top of Jilh formation. The white dots (hollow and solid) are 19 available wells. A random line is

connecting wells A, B, and C. (b) A 3D view of (a) with posting a cross section of seismic amplitude data. 148

6.4 Three wells from Figure 6.3 showing a cross section of Jilh formation. Well A from west of the study area is about 10 km away from well C. Some well logs such as porosity, density, and sonic are displayed 149

6.5 Time slices and Jilh horizon displayed on a 3D view. (a) Time slice at 1.39 sec. (b) Time slice at 1.43 sec. (c) Time slice at 1.447 sec. Jilh formation is clearly showing high structures to the east 150

6.6 The 200 km^2 study area with time structure map of top of Jilh formation. The white circles represent the available wells with solid circles indicating wells with high pressure and hollow circles indicating wells with normal pressure. W-E 2d line is going through the normal pressure well D and S-N 2d line is going through the high pressure well F 153

6.7 3D image of the study area with two time structure maps picked on top of Jilh and Khuff formations. Well D is posted on the inline going west to east and Well F is posted on the cross line going south to north. The 2D lines are showing seismic amplitude cross sections 154

6.8 Well D synthetic seismogram and nearby seismic traces. Minjur, Jilh, Sudair, and Khuff tops or markers are shown. The blue traces represent the synthetic seismogram. The red curves represent the P-wave and density well logs 155

6.9 Seismic amplitude 2D lines with posting the wells and horizons (a) W-E 2D line with well D posted and Jilh (blue) and Khuff (green) horizons. (b) S-N 2D line with posting Well F and Jilh and Khuff horizons 156

6.10 Seismic impedance (velocity) 2D lines with posting the wells and horizons (a) W-E 2D line with well D posted. (b) S-N 2D line with posting Well F. Jilh (blue) and Khuff (brown) horizons are also posted. The blue well log is porosity and the brown well log is density 157

6.11 (a) A horizon slice of top of Jilh formation extracted from the impedance seismic data. (b) Cluster of the impedance data using the SOM method. (c) Clustering of the porosity data using the SOM method. (d) Multiple clustering of impedance and porosity using the SOM method. (e) Prediction of pore pressure using the supervised MLP method.

(f) The prediction of pore pressure using the GMDH method. GMDH method gave better results than other methods but still very qualitative result due to lack of well log data .161
6.12 Porosity prediction using MLP method. The input data was seismic impedance and porosity values at the top of Jilh formation 163
6.13 Porosity prediction using the supervised methods. Seven porosity logs (black curves) were used during the training. Red solid curves are the porosity prediction of GMDH method. Red dotted curves are the porosity prediction of the neural network MLP method 164
6.14 Porosity prediction using the supervised methods. Three wells that were used during training. Black curves are the original porosity logs. Red solid curves are the porosity prediction of GMDH method. Red dotted curves are the porosity prediction of the neural network MLP method 164
6.15 Porosity prediction using the supervised neural network method. (a) The porosity result of the W-E 2D line with well D displayed. (b) The porosity result of the S-N 2D line with well F displayed. Jilh horizon is yellow and Khuff horizon is brown. The yellow logs are porosity logs and the brown logs are density logs 167
6.16 Porosity prediction using the GMDH method. (a) The porosity result of the W-E 2D line with well D displayed. (b) The porosity result of the S-N 2D line with well F displayed. Jilh horizon is yellow and Khuff horizon is brown. The yellow logs are porosity logs and the brown logs are density logs 168
6.17 Jilh Horizon slices of the predicted porosity using the supervised methods. (a) Horizon slice extracted from the porosity volume that was predicted using MLP. (b) Horizon slice extracted from the porosity volume that was predicted using GMDH. The solid and hollow circles are the available well data 169

Chapter 1

Introduction and Overview

Carbonate reservoirs, compared to siliciclastic reservoirs, offer unique geophysical challenge in reservoir characterization. The challenge includes (1) tight rock fabric resulting in problematic and not widely accepted rock physics models, (2) higher seismic velocities which will cause lower seismic resolution, and (3) physical and chemical alterations causing fracturing. Approximately 60% of the world's petroleum reserves are carbonates but, unfortunately, carbonate reservoirs account for only 40% of the total hydrocarbon production. Therefore, incremental production enhancement through advanced reservoir characterization technology can result in significant reserve addition and economic benefits.

The objective of this PhD research is to predict the carbonate reservoir parameters based on seismic attributes. The carbonate reservoir parameters include porosity and pore pressure. The seismic attributes include analytical trace attributes, geometrical attributes, and many other attributes that might be generated during the study. I use nonlinear inversion algorithms such as "neural networks" and "group method of data handling" methods in the prediction. These algorithms will be used as a tool for extracting the desired information from seismic, rather than a scientific method. These tools are not unique to seismic data from clastics, but presumably the complex way in which carbonate data encode the desired information makes their use appealing.

1.1 Terminology of pore pressure

Knowledge of formation pore-pressure is essential in the development of oil and gas fields because it will increase our understanding of the rock and fluid characteristics of

subsurface formations. Fluid pressures in the pore spaces of rocks are critical to several aspects of petroleum exploration and production. The terms “geo-pressure”, “overpressure”, and “pore pressure” will be used throughout this thesis to describe the pore fluid pressures which are greater than the hydrostatic pressure. Hydrostatic pressure, also called normal pressure, is the pressure required to raise a column of water to the surface. The pressure that results from the combined weight of the rock matrix and the fluids in the pore space is called the overburden pressure. By using the Terzaghi’s principle, one can derive the effective pressure which is the difference between the overburden pressure and pore pressure (Terzaghi, 1943). Some of the references for the pressure prediction problem include Terzaghi (1930), Hottman and Johnson (1965), Pennebaker, (1968), Eaton (1972), Bowers (1995), Dutta (2002), Huffman (2002), and many others.

As illustrated in Figure 1.1, I divide the underground formations into two zones, and call the upper zone the ‘unlocked system’ and the lower zone the ‘locked system’. The unlocked system is located above the “top of overpressure” line, and represents the state of normally pressured formations which are able to maintain hydraulic communication with the surface. The locked system is located below the “top of overpressure” line, and represents the state of under-compaction of the formation where the porosity is decreasing and the fluid pressure in the pore spaces is increasing. Geopressures are formed because the fluid motion is retarded over geologic time. The main cause of this retardation of the pore fluid is the compaction mechanism or the rapid deposition of fine-grained sediments.

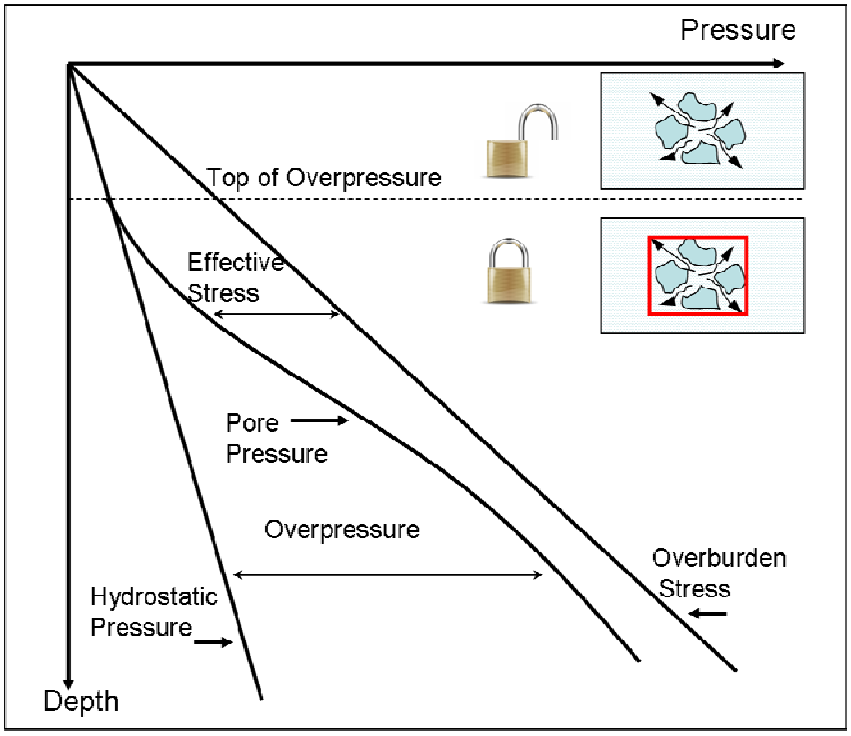


Figure 1.1. Pore pressure terminology (modified from Chopra and Huffman, 2006).

Some of the important mechanisms related to geo-pressure are: mechanical compaction disequilibrium, clay dehydration and alteration caused by burial diagnosis, dipping permeable beds embedded in shales, buoyancy, tectonic uplift and erosion, aqua-thermal pressuring, fluid expansion, and lateral transfer (Bruce, 2002; Chopra and Huffman, 2006; Dutta and Khazanehdari, 2006).

In the drilling phase, pressure prediction is a pre-requisite for safe and economic drilling, where an optimized casing and mud programme design can avoid well control problems. Geopressure prediction before drilling can be used during exploration and development

(1) To constrain the porosity and pressure regimes surrounding accumulations of hydrocarbons.

(2) To determine where source rocks are actively maturing.

(3) To map the hydrocarbon migration pathways.

(4) To determine the sealing characteristics of faults.

(5) To determine the connectivity for a given reservoir based on its location relative to pressure.

1.2 Pore pressure prediction techniques and difficulty in carbonate reservoirs

Most of the pressure detection methods are based on the observation that rocks with higher pore pressure are associated with under-compacted shales. The greater the under-compaction, the higher the porosity, and the greater the amount of the load carried by the fluid (Chilingar *et al.,* 2002). Well logs provide the most extensively used and reliable means to detect overpressures, but this information is helpful only after the well has been drilled. Overpressure detection before drilling is very useful as precautions can be taken and planning can be done accordingly. Reflection seismic data are also used for pore pressure prediction, because the overpressured intervals have lower seismic velocities and impedances that are lower than normally pressured intervals at equivalent depth. Other methods include borehole method, gravity method, 4D seismic method and neural classification of seismic attributes (Musgrave and Hicks, 1966; Tsimel'zon, 1959; Fertl, 1976; Liu et al., 2004; Vlastos et al., 2003; Al Mustafa and Al Zahrani, 2001).

Borehole methods predict the pore pressure after drilling the borehole. Well logs like porosity, sonic, density and resistivity recognize the changes in geopressure. This can be done using the relationship between compaction, density, porosity, and the acoustic properties of sediments (Chopra and Huffman, 2006). Vertical seismic profiles (VSP) can also be used to detect geopressure zones while drilling (Chopra and Huffman, 2006). Gravity methods can be used as a supplementary source in detecting geopressure (Fertl, 1976; Musgrave and Hicks, 1966; Tsimel'zon, 1959). Gravity methods assume that the geopressured zones are associated with low bulk density that could be detected by the gravimeter. There are gravimetric effects that can be detected from high porosity zones, such as the geopressured areas.

4D seismic modelling or time-lapse methods use seismic waves to monitor changes in oil or gas reservoirs as a function of time. Elastic properties, effective stress, and pore pressure change during the movement of fluids in the fractured media. During production and fluid draining, the effective stress increases on the fractures because of the decrease

in the pore pressure. On the other hand, during the water injection into the reservoir, the effective stress will decrease and that will increase the pore pressure (Schoenberg, 2002). Geopressure may be predicted using the 4D seismic modelling methods, based on the effect of pore pressure changes on seismic wave propagation.

Using seismic velocity to predict pore pressure (Dutta, 2002; Hottman and Johnson, 1965; Liaw, 2002; Pennebaker, 1968; Sayers *et al.,* 2002), seismic velocities with sufficient resolution are required. The Eaton (1975) method predicts the pore pressure directly from transforming the velocity to pore pressure. The Bowers (1995) method uses the velocities to estimate the effective stress and then calculates the pore pressure. Since these empirical methods used the velocity as an input, researchers make sure that they use the best available velocity for better overpressure prediction. Many types of velocities were used in this field. Reflection tomography (Stork, 1992; Wang *et al.,* 1995; Woodward *et al.,* 1998), horizon-keyed velocity analysis (Yilmaz, 2001), automated velocity analysis (Banik *et al.,* 2003; Mao *et al.,* 2000), residual velocity analysis (Swan, 2001), and velocities from seismic inversion (Chopra and Huffman, 2006) are some examples of the velocities used.

Neural classification of seismic attributes is one of the ways available to find the nonlinear relationship between the seismic data and pore pressure. Al Mustafa and Al Zahrani (2001) used a quick unsupervised vector quantization to cluster the seismic impedance data. They also used a supervised vector quantization to cluster the final prediction into 2 distinct classes. This method will be investigated further in chapters 3 and 4. More seismic attributes will be used to predict pore pressure. Porosity property will also be predicted. Al Mustafa and Al Zahrani used Matlab to train and apply the neural network. In this research, I will use my in-house neural networks prediction code.

Most of the pore pressure prediction methods dealt with Tertiary clastic or siliciclastic rocks (i.e., sand and shale). Theoretical equations from reflection seismic methods were used in this case. Further, the researchers made an assumption that the pressure mechanism is caused by mechanical compaction, in which the porosity was reduced during the stress history. This is not always the case, as, for example, in carbonate reservoirs (limestone and dolomite). Pressure mechanisms in carbonates are poorly understood, and their effect on the seismic velocity signature is usually absent.

Moreover, in carbonates the velocities are often controlled by secondary porosity and fractures (Dutta, 2002). Also, carbonates have different compaction behaviour from the clean shales used to build the compaction curves in siliciclastic rocks (Chopra and Huffman, 2006). Lithology and porosity usually control the factor of determining the sonic velocity in rocks, but pore type is also important when dealing with carbonate rocks. The theoretical equations used to predict pore pressure do not account for this modification, which might affect the sonic velocities (Anselmetti and Eberli, 1993).

Carbonate rocks have higher velocity trends than the clastic rocks. For example, a carbonate mineral called calcite is 6500 m/s and dolomite is 7300 m/s. Even for one single carbonate rock like dolomite, the velocity depends more on the type of dolomite than the amount of dolomite. A fabric destructive dolomitization will produce a loss rock and give a low velocity, but a fabric preserving dolomitization will result in a densely cemented rock with high velocity (Eberli *et al.*, 2003).

Thus pressure variations in carbonate might not be detected by most of the current prediction methodologies. Figure 1.2 shows velocity versus effective stress plot, showing compaction curves for clastic and carbonate materials and relative differences in their sensitivity.

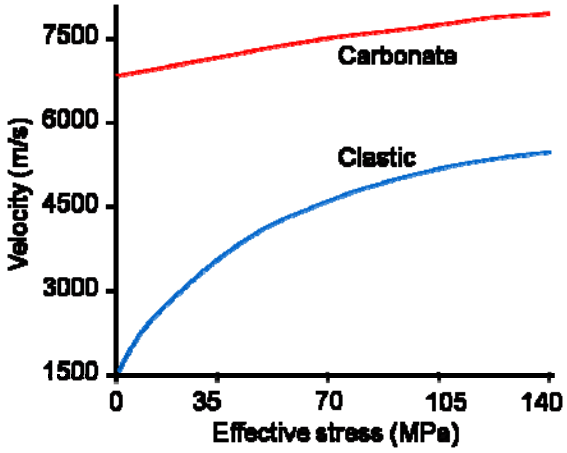


Figure 1.2. Compaction curves for clastic and carbonate materials. (modified from Huffman, 2002).

The use of neural network techniques can illuminate seismic attributes indicative of geo-pressure in gas-filled carbonate reservoirs (AlMustafa and AlZahrani, 2001). In this study, more investigations will be done using the artificial intelligent methods as a tool. These artificial intelligent methods are neural networks methods and the group method of data handling. When choosing the most effective seismic attribute that will predict the reservoir properties, I use an unsupervised statistical method. Meanwhile, I use a supervised statistical method to handle the well-logs and seismic data integration. The input data are the seismic attributes which will be described in the next section.

1.3 Seismic Attributes

Seismic attributes extracted from seismic data can be used in the interpretation of geologic structure, stratigraphy, and rock/pore fluid properties. A good seismic attribute allows us to define the structural or depositional environment and thereby to infer some feature or properties of interest.

Many seismic attributes can be derived from complex-trace analysis (Taner *et al.*, 1979). The complex seismic trace can be written as

$$c(t) = s(t) + iq(t) \tag{1.1}$$

where $c(t)$ is the complex trace, $s(t)$ is the real seismic trace $q(t)$ is the quadrature trace or Hilbert transformed trace and $i = \sqrt{-1}$. The instantaneous amplitude or the amplitude envelope can be written as

$$A(t) = \sqrt{s(t)^2 + q(t)^2} \tag{1.2}$$

The instantaneous phase can be written as:

$$\phi(t) = \tan^{-1}(q(t)/s(t)) \tag{1.3}$$

We can also get another instantaneous attribute by differentiating the instantaneous phase:

$$\omega(t) = d(\phi(t))/dt \tag{1.4}$$

where $\omega(t)$ is called the instantaneous frequency.

Many other seismic attributes can be generated from the instantaneous attributes. Here I show the seismic attributes from a 3D seismic data cube. 10 seismic attributes extracted are:

(1) Instantaneous amplitude;
(2) Instantaneous phase.
(3) Instantaneous frequency.
(4) Frequency weighted envelope (average frequency).
(5) Thin-bed (instantaneous frequency – average frequency).
(6) Instantaneous bandwidth.
(7) Normalized phase.
(8) First envelope trace derivative.
(9) Second envelope trace derivative.
(10) Instantaneous Q factor.

Some geometric attributes will also be used in the research. These attributes include coherence and curvature, and relate more to reservoir properties. Also, they can give more information with regard to the faults and fractures of the reservoir, even in different scales.

Some or all of these attributes are needed to complete the job of statistical methods which will be mentioned in the next chapters. The statistical method will decide which attribute is better for predicting the target reservoir characteristics.

1.4 Artificial Neural Network

The first artificial intelligence tool that I use is the artificial neural networks (ANN). ANN is a non-linear information processing tool that attempts to mimic some of the features of the human brain. The key element is the information processing system which

is composed of a large number of highly interconnected processing elements working together to solve specific problems. These processing elements are called neurones or nodes. ANN, like the human being, has the ability to learn by example, remember, and come up with general conclusions from a small training data set. Both learning in biological systems or ANN involves adjustments of the connections that exist between the neurones.

ANN takes a different way to solve a problem from that of conventional computers. The conventional computers follow some instructions in order to solve a problem. The computer capability of solving a problem is restricted by the way that we understand the problem. The computer needs to know the specific steps that can be followed to come up with the final result. ANN learns by example and finds out how to solve the problem by itself. While ANN is good in imitating some of the features of the human brain, conventional computers are better for precise computation and time saving.

ANN is suitable for classification, pattern matching, and clustering through learning and training. Some of the advantages of using ANN include its ability to:

(1) Learn how to do tasks based on the given data.
(2) Deal with data that are contaminated with noise or inconsistency.
(3) Visualize low-dimensional views of high-dimensional input data.
(4) Create its own representation of the input data.
(5) Adjust itself to new situations and learn by experience.
(6) Be robust and fault tolerant.

Figure 1.3 shows a biological neuron. The most important elements of the biological neuron are the cell body (soma), the dendrites, the axon, and the synapse. A typical real neuron accumulates chemical signals from other neurons through fine structures called dendrites. If the total chemical accumulation exceeds a threshold, the neuron sends its own signal of electrical activity through a long thin stand known as an axon. An axon has a length varying from a fraction of a millimetre to a metre in the human body. The axon splits into thousands of branches, and at the end of each branch there is a structure called a synapse (terminal buttons) which converts the activities from the axon into electrical

effects. Each neuron receives input from as many as 1000 synapses. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. A neuron requires a different combination of synapse inputs to trigger an output signal.

ANN tries to mimic the power and strength of biological neural networks. Figure 1.4 shows an example of the artificial neuron. The processing element is similar to the cell body (soma). The connections between the nodes resemble the axon and the synapse from the biological neurons. The processing element is comprised of a summation function that creates a weighted sum of all the inputs. The weights on each link are synonymous with the strength of the corresponding synapse.

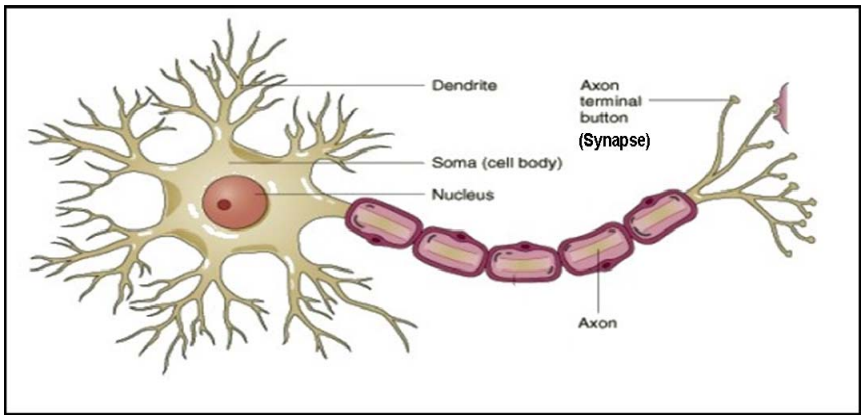


Figure 1.3. A Biological Neuron.

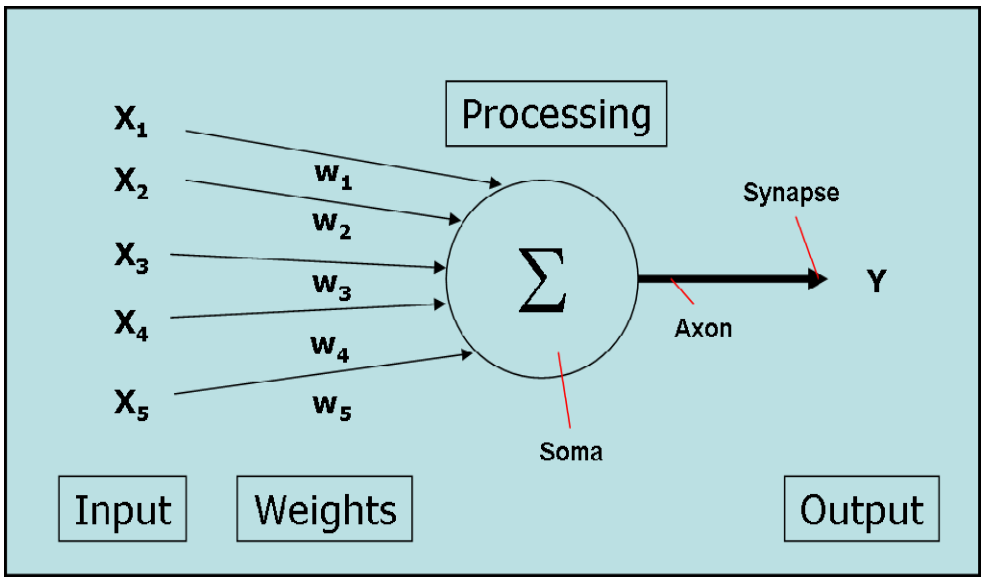


Figure 1.4. An Artificial Neuron.

In the ANN, a layer is composed of many neurons operating in parallel. A network can have one single layer or many layers. The most common layering scheme is when we have input layer, hidden layer, and output layer. The input layer is connected to the hidden layer which is connected to the output layer. The input layer represents the raw information or input data (seismic or seismic attributes in our case) that is fed into the network. The activity of the hidden layer is determined by the activities coming from the input layer and the weights on the connections between the two layers. The behaviour of the output layer depends on the activity of the hidden layer.

Based on the type of connections between neurons of the same layer or neurons from different layers, we can classify the ANN into two major types. Figure 1.5 shows the feed-forward networks (blue arrows) and the feed-backward (red arrows) networks. When the signals are travelling in one way only, from input to output, we call this type of network a 'feed-forward' ANN. The output of any layer does not affect that same layer. This type of network associates inputs with outputs, and it is extensively used in pattern recognition. The signals on the feed-backward networks can travel in both directions. This type of network is complicated but very powerful. Feed-backward networks are also called recurrent networks.

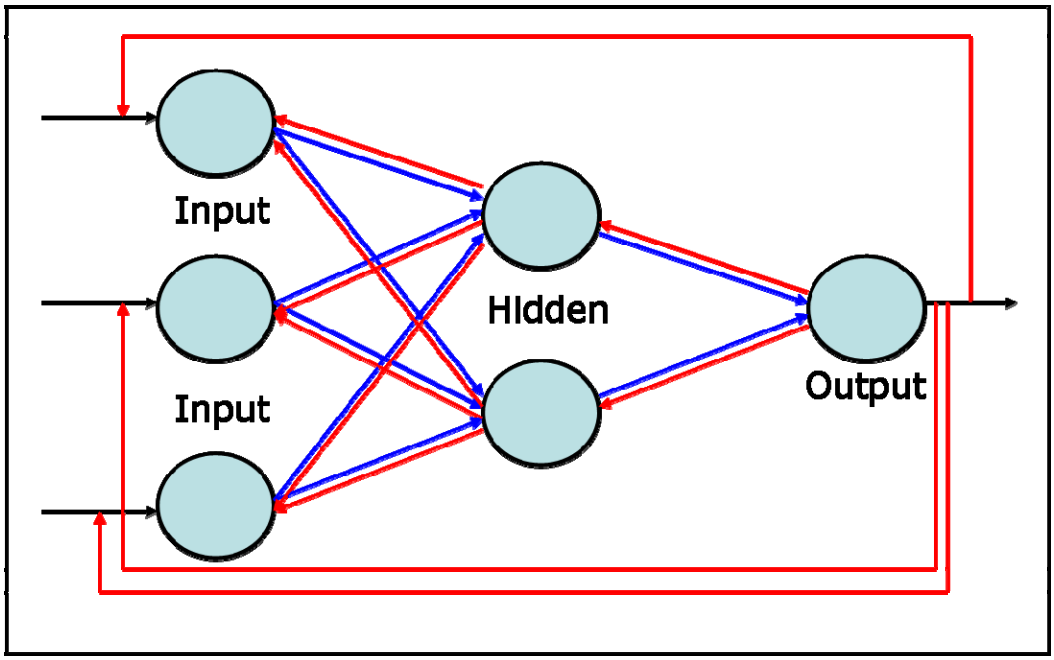


Figure 1.5. Feed-forward ANN (Blue arrows) and Feed-backward ANN (Red arrows).

The learning mechanism is an important part of the ANN. As mentioned before, ANN layers are connected by links which are associated by weights. The values of these weights are given randomly at the beginning of the learning process. During the learning process, the network determines the best values of the weights. Following the way learning is performed, we can distinguish two major categories of ANN. The first learning type is the fixed networks in which the weights are fixed and do not change during the learning process. The second type of learning is the adaptive networks which are able to change their weights. In this study, the weights will be changed during the learning process.

Most conventional applications of neural networks deal with the adaptive learning. The adaptive learning networks can be classified into two major categories: supervised learning and unsupervised learning. The main difference between these categories lies in the amount of prior information that is supplied. In the supervised learning networks, each output node is told what its desired response to input signals ought to be. This is called learning by example or learning that includes an external teacher. The aim of this learning approach is to determine a set of weights which minimises the error between the

predicted and actual values. In this study, well logs will be used to guide the network in predicting reservoir parameters. Radial basis functions (RBF), Multi-Layer-Perceptrons (MLP), and Generalized Regression (GRNN) are examples of networks that are trained by supervised learning.

Unsupervised learning is based upon only local information. This is called competitive learning that uses no external teacher. The aim here is to find a structure within the input data. It is also called self-organisation, in the sense that similar inputs activate similar neurons while different inputs activate other neurons without any feedback about the output accuracy. Self-organizing map (SOM) is one of the most promising mathematical techniques to be applied to unsupervised pattern classification (Kohonen, 1997). The SOM is related to vector quantization methods (Haykin, 1999). In this study, SOM will be used to investigate the relationship between the input seismic attributes data and reservoir parameters.

1.5 Group method of data handling (GMDH)

The "group method of data handling" was invented in 1967 by Professor A. G. Ivakhnenko from the Institute of Cybernetics, Ukrainian Academy of Sciences. The main idea behind GMDH is that it builds a polynomial function which would behave in such a way that the prediction output would be as close as possible to the actual output. By using the resultant polynomial function, GMDH will be capable of representing complex and uncertain relationships between input and output variables (Lee *et al.,* 1999). GMDH can automatically decide on the best network structure, number of layers, and strength of nodes. This automation makes the GMDH algorithm superior to other methods such as artificial neural networks. In GMDH, complex inputs of many types, such as multiple seismic attributes, can be simplified into smaller subsystems and groups. These subsystems are generated inside multiple layers.

The general quadratic polynomial regression function used is

$$y = A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_ix_j, \tag{1.5}$$

where y is the output, (x_i, x_j) are a pair of input variables, and (A, B, C, D, E, F) are the polynomial coefficients.

(1) If the input to our algorithm is x_1, x_2, x_3, ..., x_m, then each pair of input variables (x_i, x_j) will use one quadratic polynomial equation (equation 1.5) to produce an output y.

(2) The new variables produced from multiple polynomial functions will form a new layer.

(3) The polynomial functions will be used again to evaluate values of layer 1, and the outputs from layer 1 will serve as an input to subsequent layers.

Figure 1.6 shows the basic structure of the GMDH algorithm.

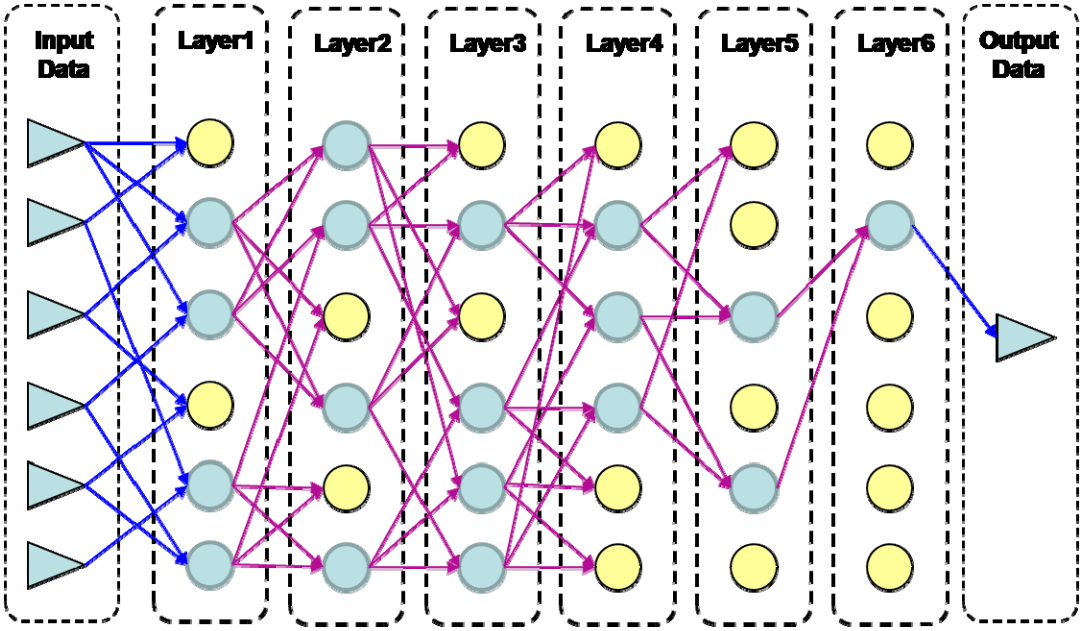


Figure 1.6. Basic structure of the GMDH algorithm.

The inputs are fed to the network and sent into the first layer. The inputs here are multiple quantities, like different seismic attributes. Each node of layer 1 is constructed using a specific pair of inputs. A classical GMDH algorithm uses only one pair of inputs to create any new node using equation 1.5. The process continues to the next iteration by using layer 1 as the new inputs. Pairs of inputs will be used again from layer 1 and outputs will feed layer 2. This process continues as long as each new layer produces

better results than the previous layers. When the new layer is found not to be as good as the previous layer, the training process ends.

The stopping criterion is done by testing the fit of the polynomial equations. The testing might use “mean square error”, “root mean square error”, or new testing methods like “the predicted square error”. In each layer, GMDH will sort the nodes based on the predicted error. The best node with the lowest error in each layer will be saved for the sake of comparing it with the next layer errors. Only nodes that have small errors will be saved and used for the next iteration. This saves the network computation time.

At the end of the training process, GMDH will save the new structure and effective nodes. The effective nodes will be saved along with the polynomial coefficients used to generate the nodes. These polynomial coefficients will be used in the application stage. The GMDH network in the application stage is used with data that were never introduced to the network during training.

GMDH was used in classification of difficult pattern recognition problems, in determining which inputs are more important to the modelled system, and in predicting the outputs of complex systems. In the industry, GMDH was used to predict tool life in drilling, predicting temperature distributions, predicting an acceptable product on T-shape tube hydro-forming process, radar applications, missile guidance, multisensory signal processing, biomedical modelling, econometric forecasting, environmental systems, and many other usages (Abdel-Aal, 2003; Barron *et al.,* 1984; Farlow, 1984b; Fulcher and Brown, 1994; Lee *et al.,* 1999; Lin and Kwan, 2004). From my reading while doing the literature review, I never came across any prediction of geophysical reservoir properties from seismic data using GMDH.

1.6 Summary of this thesis

It is important to investigate carbonate rocks because they hold more than half of the world’s petroleum reserves. Because of the good velocity signature produced from seismic data, pore pressure prediction on clastic data was very well researched and produced good pre-drill prediction results. Predicting geo-pressure before drilling is vital

in preventing blowouts, loss of circulation, formation damage, and incorrect casing design or mud weight programme.

Well logs are very helpful to know the true pore pressure of the formation, but this is all achieved after drilling the well. Reflection seismic methods are commonly used for pore pressure prediction in clastic rocks, but these methods are not suitable for carbonate rocks because of the weak effect of the pressure on the seismic interval velocities. In this study, I will use artificial intelligent methods, such as neural networks and the group method of data handling along with seismic attributes, to solve the problem of predicting carbonate reservoir parameters. The goal is to predict reservoir properties such as porosity and geopressure.

Chapter 2 presents an experiment to show the importance of choosing high resolution velocity as an input to the pore pressure prediction algorithm. In this chapter, pore pressure will be predicted using high-resolution velocities. This will be done by use of empirical equations like the Bowers equation.

Chapter 3 presents the first artificial intelligent tool which is the neural networks method. In this chapter, a more detailed explanation will be given with regard to the unsupervised neural networks method. The unsupervised neural network method will be used to predict pore pressure.

Chapter 4 introduces the supervised artificial neural networks method. Different supervised operators will be used here to predict porosity and pore pressure.

Chapter 5 is about the second artificial intelligent tool, the group method of data handling. This tool is new to the reservoir geophysics environment. The tool will be used to predict pore pressure and porosity.

Chapter 6 is showing and discussing the Jilh reservoir characterization. Geological interpretation is given and a comparison between the prediction results is shown.

Finally, chapter 7 provides the overall conclusions of the thesis and contains recommendations for future work.

In summary, my main contributions made in this study are:

(1) Development of velocity-based geopressure prediction code, to prove that geopressure in carbonate seismic data is hard to predict using empirical relation

between velocity and pressure, and to prove that using high-resolution velocity is helpful in getting better pore pressure prediction.

(2) Development of an unsupervised neural networks code, to predict geopressure in carbonate reservoirs.

(3) Development of a supervised neural network code, to predict porosity, which in turn used as an attribute for pressure prediction.

(4) Development of the group method of data handling algorithm and the first time application with seismic data, to predict porosity and geopressure.

Chapter 2

Carbonate pore pressure cannot be predicted by seismic velocity

In this chapter, I focus my discussion on the pore pressure prediction using a seismic velocity method. I first test the prediction using high-resolution velocities obtained from seismic travel time and waveform tomography, just for a quick testing of the prediction method. I then apply it to a real example of carbonate reservoirs. I will draw the conclusion that carbonate pore pressure *cannot* be predicted from seismic velocity alone.

2.1 Pore pressure prediction using tomographic velocities

Velocity analysis for geo-pressure prediction requires a level of detail in the velocity analysis. Seismic gathers are usually conditioned for geo-pressure prediction to assure the highest possible quality velocity analysis. The application of such conditioning can dramatically improve the quality of the resulting velocity analysis. Having an acceptable velocity data in hand could lead us to estimate the pore pressure using some empirical relationship between the velocity and the effective stress. Bowers (1995) provided a method to determine effective stress from velocity using the following equation:

$$\sigma = [(v - v_0)/A]^{1/B}, \tag{2.1}$$

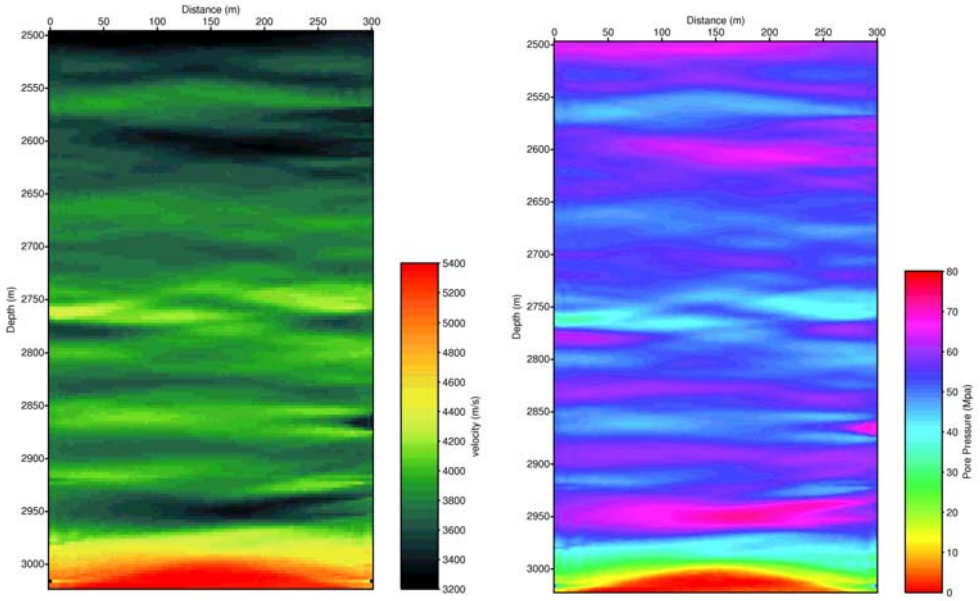
where σ is the effective pressure, v is seismic velocity, v_0 is the velocity of unconsolidated fluid-saturated sediments (v_0 = 1524 m/s here), and A and B are the constants varying with different regions. This method has recently been used successfully in the Gulf of Mexico. It is based on the fact that during compaction a velocity increase occurs. Then, a velocity reversal will happen because of the fluid expansion. The Bowers (1995) method needs only velocity as an input to produce the effective stress, and is used

here as a quick test with clastic data that contains sandstone and shale, to prove that pore pressure prediction needs high resolution data as input.

Velocities are obtained from travel time and waveform tomography in crosshole geometry. The travel time tomography uses first arrival times to reconstruct a velocity distribution between two vertical boreholes. As recorded crosshole seismic data contain not only first arrival time information but also scattered energy waveforms that are not utilized in travel-time inversion, waveform tomography may produce high-resolution velocity image. The seismic data were gathered between two parallel boreholes 300 m apart. The lithology of the area contains horizontal layers of mudstone and sandstone. The igneous rocks come at the bottom of the section. The inversion was done at a depth range from 2497 to 3022 m. 175 source points were in one vertical borehole and 175 receiver points in the other. Source and receiver intervals are 3 m.

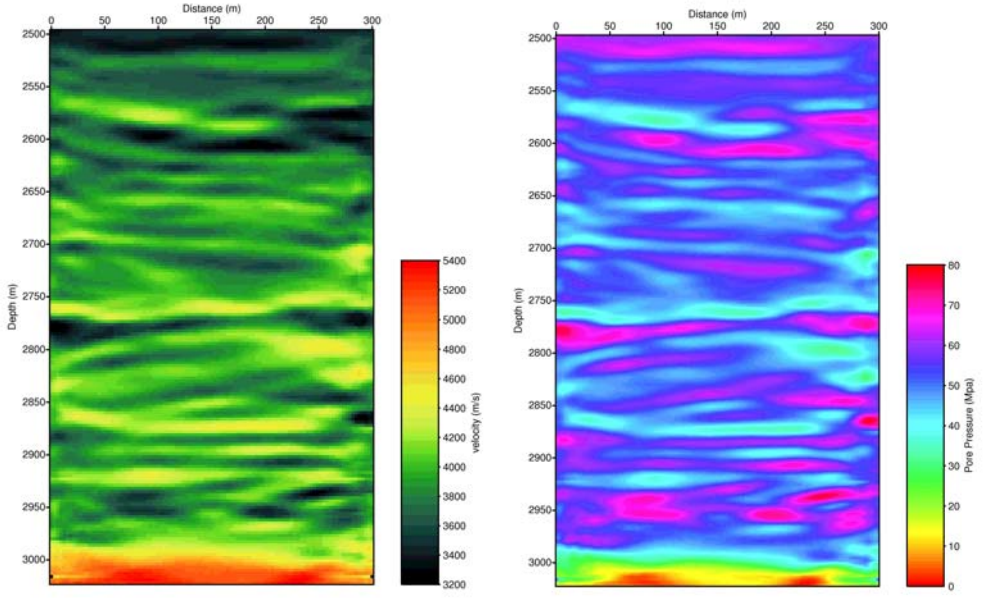
Figure 2.1 displays the pore pressure predicted based on the velocity image of traveltime tomography, presented in Rao and Wang (2005). The values of the two images look opposite to each other. If the pore pressure is high, it usually means higher porosity and low velocity. If the pore pressure is low, it usually means lower porosity and higher velocity.

The waveform tomography, presented in Wang and Rao (2006), gave us a better velocity image (Figure 2.2a) in terms of resolution. It shows more details and features, compared with the traveltime tomography image. As the inversion method improves, the pore pressure prediction (Figure 2.2b) has more detailed information and better resolution.



(a) (b)

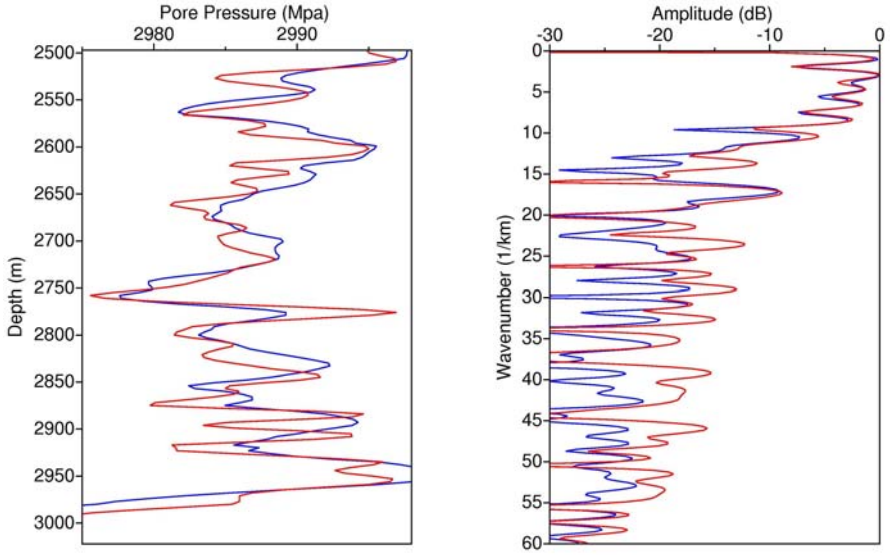
Figure 2.1. (a) Velocity image from using travel-time inversion. (b) Pore-pressure image predicted from the velocity image.



(a) (b)

Figure 2.2. (a) Velocity image obtained from waveform tomography. (b) Pore pressure predicted.

Now I compare the resolutions of pore pressures semi-quantitatively in Figures 2.1b and 2.2b, respectively. Figure 2.3a shows that the lateral average of predicted pore pressure values from waveform tomography (red curve) shows more details than that from traveltime tomography (blue curve). Figure 2.3a does not show the physical meaning or the type of lithology in the area because the displayed traces are the average of predicted pressure traces from the 2D line in Figures 2.1b and 2.2b. Figure 2.3b clearly indicates that the pore pressure predicted using the waveform tomography velocity (red curve) has higher resolution than that using the velocity of traveltime inversion (blue curve). Here I transfer each column of the predicted pore pressure to the wavenumber (spatial frequency) domain, and average the wavenumber spectra over all columns, to generate these wavenumber traces that reflects the spatial resolution in vertical direction. I noticed that at lower wavenumber, both the traveltime and waveform tomography have a similar response, whereas at a higher wavenumber the waveform tomography has larger amplitudes.



(a) (b)

Figure 2.3. (a) Average predicted pore pressure curves from travel-time tomography (blue) and waveform tomography (red). (b) Resolution curves: The blue curve is from the travel-time tomographic velocity. The red curve is the resolution of pore pressure predicted from the waveform tomographic velocity.

The Bowers method in equation (2.1) used different values of constant A and B for coast wells and deepwater wells: $A = 3.02013$ and $B = 0.8168$ for coast wells, and $A = 28.07654$ and $B = 0.6207$ for deepwater regions. I set up a simple experiment to test the influence of the constants to the pore pressure prediction result: I change the A value between 4 and 29 and keep B fixed. Figure 2.4 shows the wavenumber spectra of the predicted pore pressures, by using 6 different A values: 4.0, 9.0, 14.0, 19.0, 24.0 and 29.0. The red curve has a value of $A = 29.0$ while the black curve has the value of $A = 4.0$. This figure reveals that the frequency content is almost the same for all of the variable constants, and that the magnitude is slightly different when I use different A values. It suggests that for the current example, changing in coefficients will not have much difference in the distribution of final result of pore pressure prediction, although they could cause the local magnitude difference. In the example shown in this chapter, I use the second group of constants of Bowers (1995).

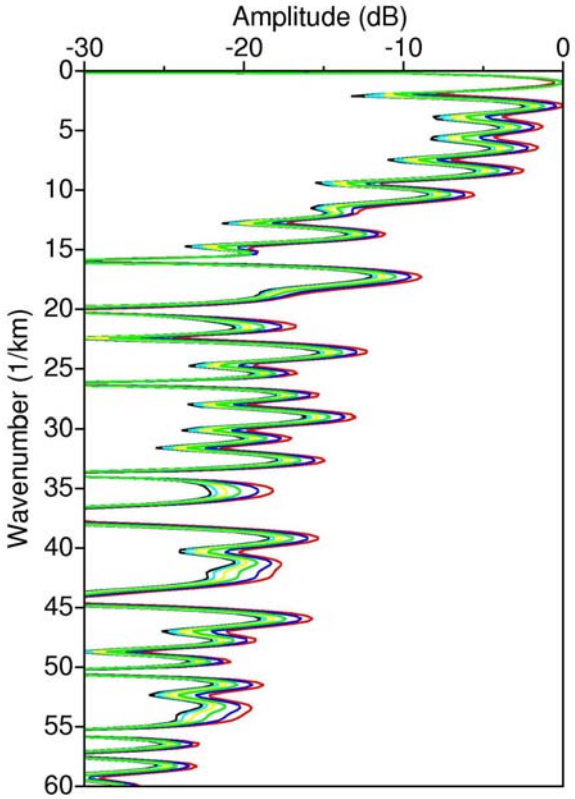


Figure 2.4 The frequency spectra of the predicted pore pressures, using 6 different A values and a constant B value.

I apply the pore pressure prediction method to the sonic logs (velocity data) from both wells, and attempt to compare the pore pressure predicted using the well log data with the one predicted using the waveform tomography seismic inversion data. The sonic log data for borehole number one, located on the left side, starts from a depth of 2500 metres. The sonic log on the right side starts from about 2600 metres deep. Figure 2.5 compares directly the pore pressure predicted from the sonic log (blue curve) and the trace of pore pressure image predicted from the seismic velocity (red curve). As the sonic log has higher resolution than the seismic velocity (obtained from waveform tomography), the resolution of the pore pressure is also high. However, the two results are consistent and the two curves generally follow the same trend.

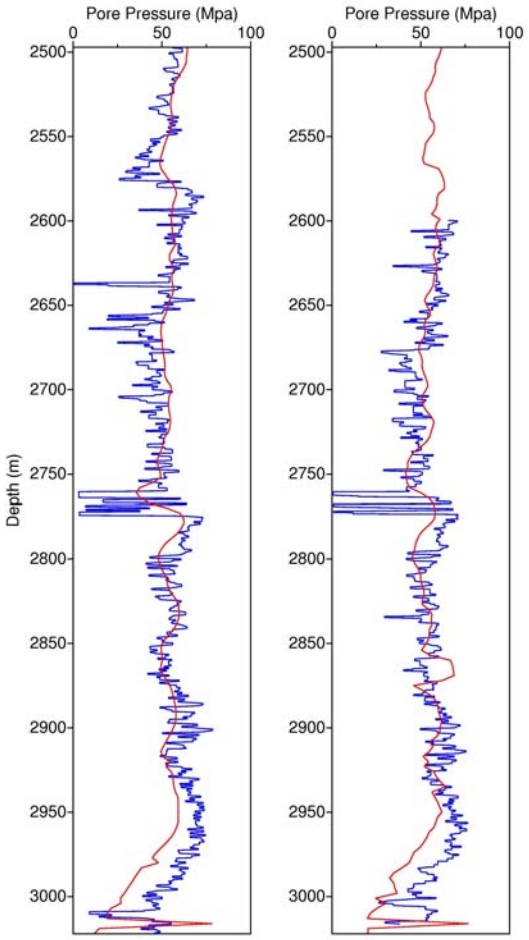


Figure 2.5 Pore Pressure Predicted using well logs (blue curves) compared with pore pressure predicted using seismic inversion (red curves).

2.2 A real carbonate seismic data set

I use a 3D seismic data set acquired over a carbonate reservoir area to prove that the velocity-based geopressure prediction method is not applicable in this carbonate case. The study area was extracted from the Uthmaniyah portion of Ghawar field. Figure 2.6 shows the study area, which is from the largest conventional oilfield in the world, which accounts for more than half of the cumulative oil production of Saudi Arabia. The target carbonate formation is called the Jilh formation and is divided into three layers of dolomite, namely upper, middle, and base dolomite. The formation is of the Middle Triassic age. Our target formation consists mainly of dolomite and inter-bedded thin shale bodies. The total thickness of the formation is about 900 feet (274.3 m).

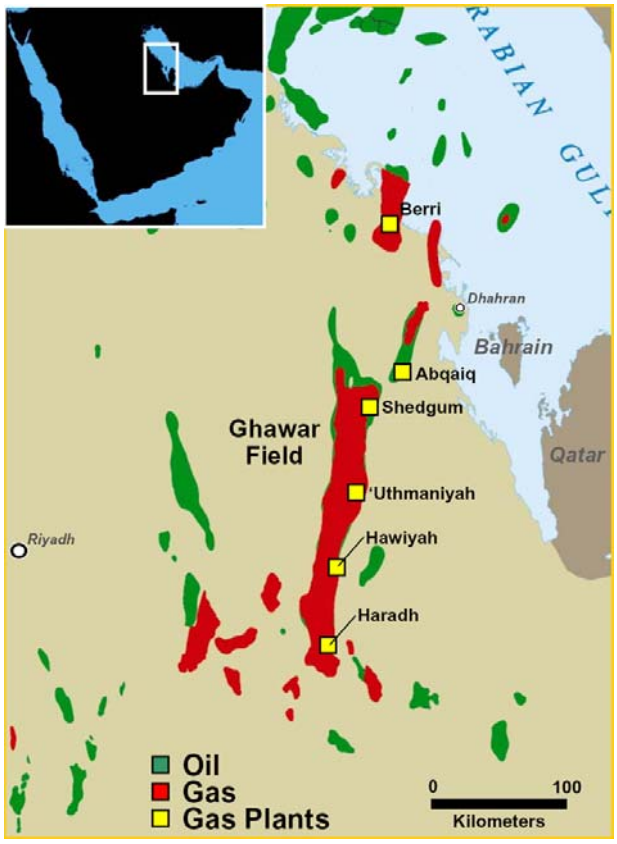
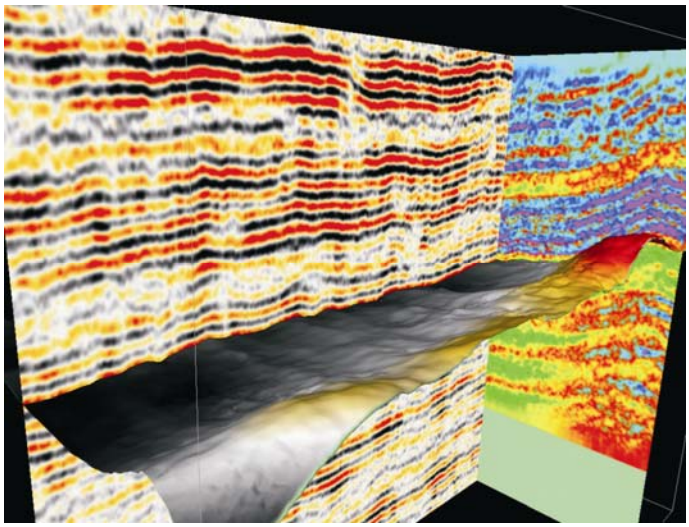


Figure 2.6 Uthmaniyah portion of the Ghawar field.

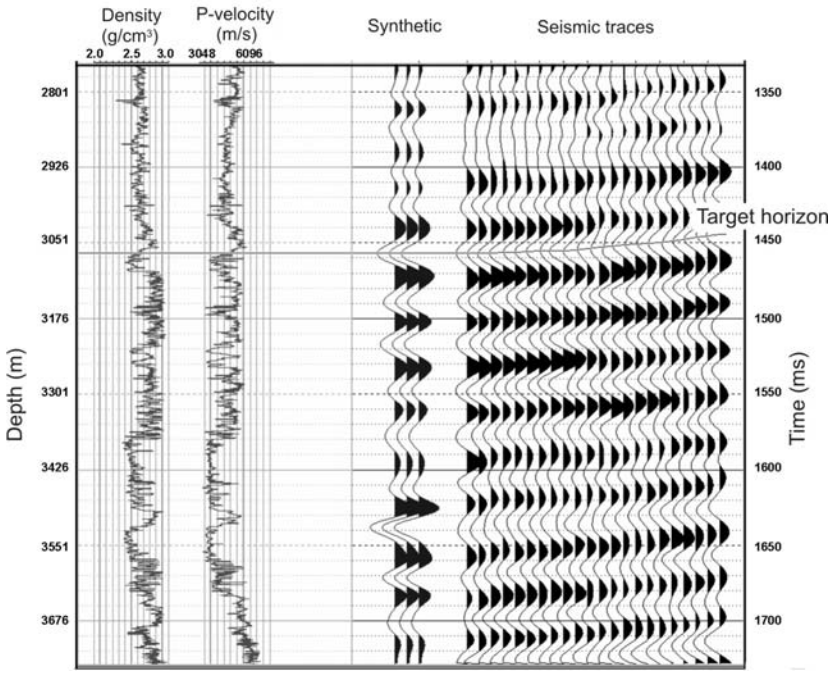
Throughout the field, the depth of the target formation varies between 2956.6 and 3200.4 m (9700 and 10500 feet). The study area encompasses a 3-D seismic volume consisting of 411 sub-lines (in-lines) and 226 cross-lines (CDP's) with an interval of 25 m in both directions. The extent of the field is about 200 km^2. Data were processed at 2 ms. The zone of interest is equal to 100 ms on the seismic time section. The seismic inversion data were also provided by the Saudi Aramco Company, and the inversion was performed using the industry standard STRATA software. Figure 2.7a shows the 3D seismic dataset with a horizon picked at the top of the target carbonate formation. Figure 2.7b shows the tie between the synthetic seismogram and the seismic data at one example well in the study area.

A complete geological understanding of the formation in the study area is lacking owing to the limited number of cored wells. The geopressure in this formation is maintained by a competent seal, but in some places this seal loses its competency, perhaps by fractures or high porosity, and the pressured fluids are permitted to escape. In addition to the vertical notion of pressure maintenance, there is an important lateral aspect as well, related to regional faulting in the area. Based on the geological understanding of this formation, the overpressure mechanisms are likely to be caused by fluid volume increase or oil to gas cracking.

Geopressure data, in terms of occurrence or non-occurrence of abnormally high formation pressures across the target carbonate formation, is available from 19 wells: 10 of these wells have over-pressure and 9 wells have normal pressure. The study of geopressure in this area will be qualitative in nature owing to the lack of geopressure logs. The 19 wells have some well logs like sonic log, gamma ray, and porosity. Porosity logs will be used to predict seismic porosity cubes.



(a)



(b)

Figure 2.7 (a) A 3D seismic data set with a horizon picked at the top of the target carbonate reservoir. The display has a seismic amplitude line (left) and an inversion line (right). (b) Well-logs, synthetic seismic trace, and real seismic section.

Figure 2.8 displays a horizon slice (in two-way travel time) that covers the top of target formation. This time map shows a structure or a closure on the north east side. Three 2-D seismic lines (A-A', B-B' and C-C'), each of which consists of 100 traces, are extracted from three cross-lines for testing the velocity-based geopressure prediction method. Seismic lines A-A' and B-B' pass through Well A and Well B, respectively, which show overpressure occurrence. Seismic line C-C' passes through Well C, which does not show any overpressure on the target horizon. The overpressure information was supplied as the mean of mud weights in drilling. The mud weights for Wells A and B reached about 147 MPa at depths of 3039 and 3063 m respectively.

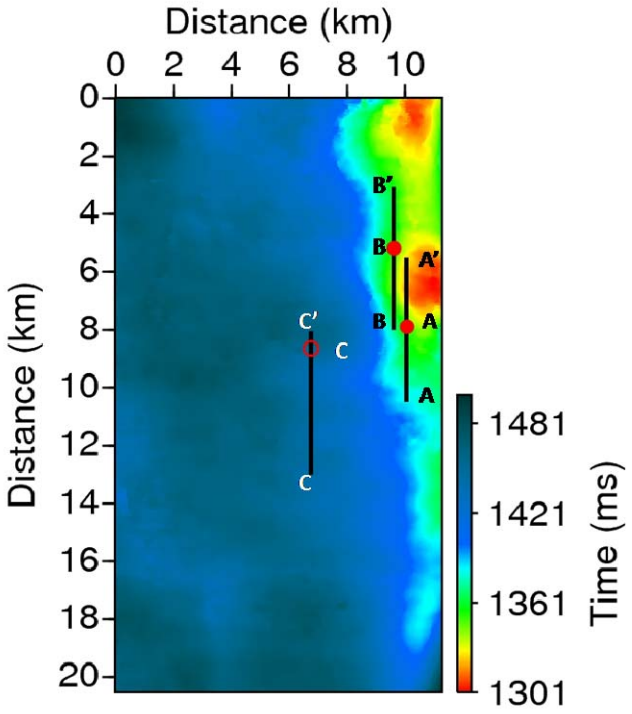
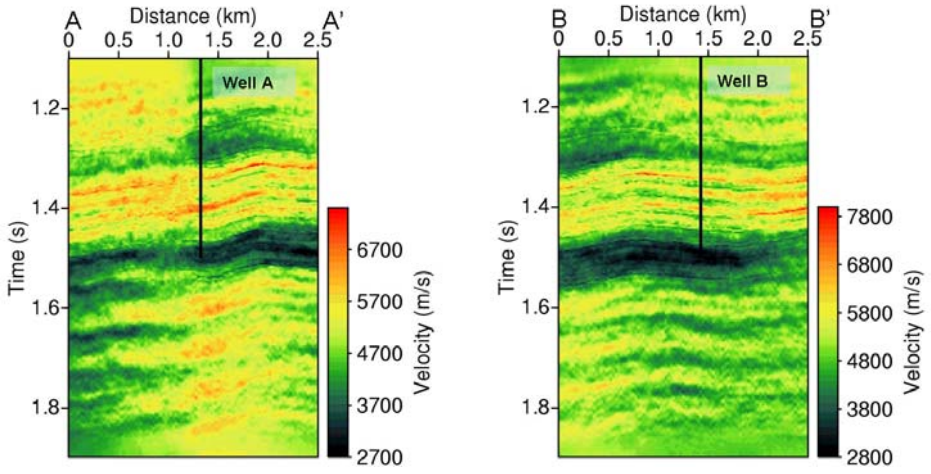


Figure 2.8 A horizon slice or time map that covers the top of the target carbonate formation. Wells A, B, and C are located in lines A-A', B-B', and C-C' respectively.

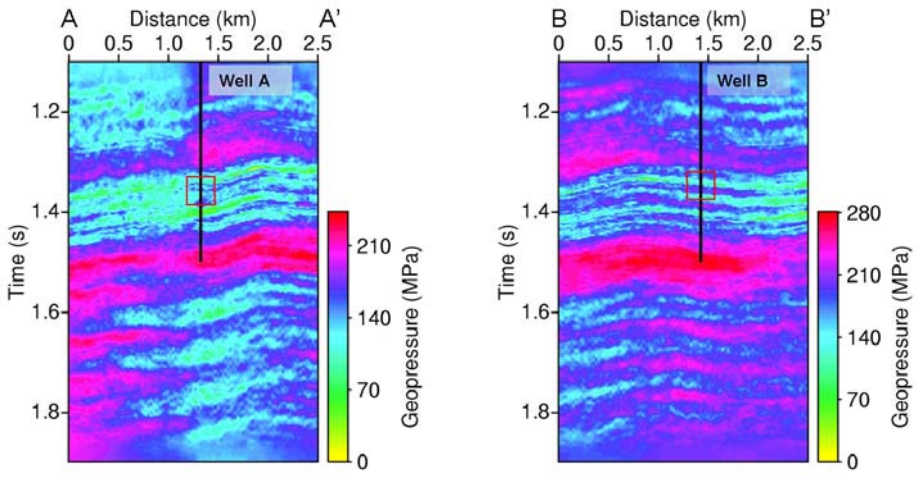
2.3 Geopressure prediction based on velocity

Velocity-based geopressure prediction methods like the Bowers method were used with clastic data with undercompaction as the dominant mechanism for overpressure. Undercompaction is not the case with our carbonate overpressured data, and this experiment should give negative results, as we will see. Figure 2.9 displays two velocity profiles inverted from seismic sections A-A' and B-B', respectively, and predicted geopressure images correspondingly. Red squares on the well trajectory show the location of the encountered overpressure zone. Figure 2.10 shows the input velocity data and geopressure prediction in line C-C' and around Well C. The result, however, does also show an over-pressure zone, which should be a normal pressure zone revealed in drilling.

Figure 2.11a is the velocity horizon slice extracted from the input inversion cube. I use this horizon slice as an input, with constant density above the horizon slice, as per the Bowers method, to predict geopressure. As I expected, the predicted geopressure in Figure 2.11b is just the opposite sign of the input velocity. Again, the predicted geopressure image shows that Well C is located over the higher pressure zone, which should be a normal pressure zone revealed in drilling.

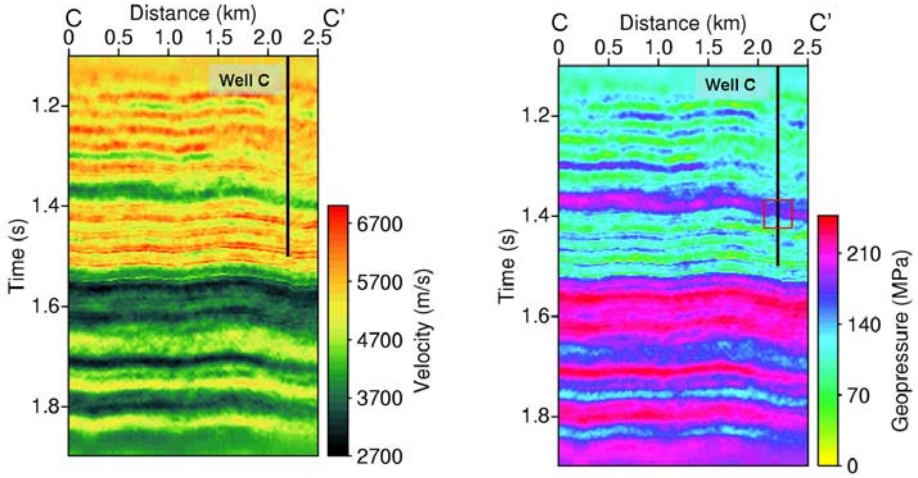


(a) (b)



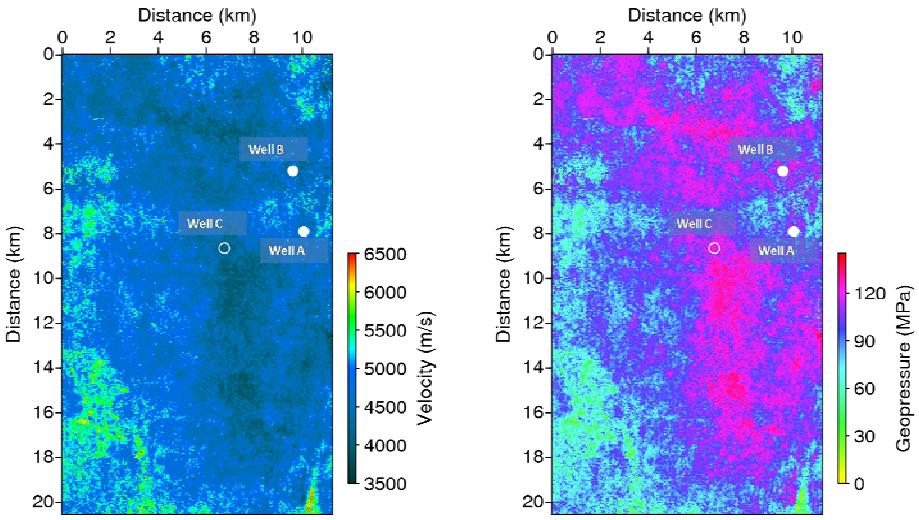
(c) (d)

Figure 2.9 (a, b) Velocity images inverted from line A-A' and B-B'. (c, d) Geopressure profiles predicted based on the velocity images.



(a) (b)

Figure 2.10. (a) Velocity image inverted from line C-C'. (b) Geopressure predicted based on the velocity image. The square around Well C does show an over-pressure zone, which should be a *normal pressure zone* revealed in drilling.



(a) (b)

Figure 2.11. (a) Velocity horizon slice extracted from the input inversion cube. (b) Predicted geopressure after applying the Bowers method to the input inversion horizon slice.

The predicted geopressure results above are not reliable unfortunately, as the predicted pressure is very high in a normal pressure zone around Well C. This is because the Bowers equation was designed to be used within the velocities of clastic reservoirs and not the velocities of carbonate reservoirs. Carbonate rocks have a flat velocity-effective stress gradient, and that results in very little velocity sensitivity to changes in pressure in these rocks (Huffman, 2002). In the following chapters, I use artificial intelligent methods to find the non-linear relation between the seismic attributes and geopressure in the carbonate reservoirs. The velocity methods were using velocity only as an input but the artificial intelligent methods will be using multiple attributes that should give me more physical meaning when predicting the reservoir properties. Also, the artificial intelligent methods can deal with complex data such as the carbonate reservoirs.

2.4 Summary

In this chapter, I have made a direct comparison between the pore pressure predictions of two seismic velocity inversion methods to show the importance of using a

high resolution velocity inversion method. Seismic waveform tomography has showed an improvement of resolution when compared with the traveltime inversion method, and that has given us better and more detailed pore pressure prediction. Therefore, high resolution velocity inversion methods are essential to produce better geopressure prediction.

However, the seismic velocity method used (Bowers) is very simple and cannot handle complicated carbonate cases. When the Bowers method is applied to a real carbonate data set, it has given a unreliable result. This experiment has proven that, for geo-pressure prediction in carbonate reservoirs, a non-linear method such as neural networks is certainly needed.

Chapter 3

Unsupervised artificial neural networks

The self organizing map (SOM) is used throughout this chapter as an unsupervised artificial neural network tool (Kohonen, 1997) to find a non-linear relationship between the seismic attributes and reservoir parameters. It is one type of neural networks that presumes no prior knowledge about the object to be classified. The classification is based entirely on the internal structure of the data.

The SOM visualizes low dimensional views of high dimensional data while preserving the topological properties of the input space. In this research, classified clusters or regions from different seismic attributes are compared to find out which seismic attribute relates most to the target reservoir property (pore pressure or porosity etc). Then additional information (like the mud weights data) is used to interpret and to label the SOM results.

3.1 Self-organizing map concept

The SOM algorithm is capable of producing a map that shows similar input data items appearing close to each other (Kohonen, 1997). The SOM classification is often called feature-discovery, in which the input space is clustered and classified into the SOM space.

A schematic representation of the SOM method is shown in Figure 3.1, which consists of three layers: an input layer, a hidden layer, and an output layer. The input layer is connected by some weights to the hidden layer. The weights can be distributed randomly at the beginning of the SOM process. The output layer produces two helpful results which are the index of the matching and the degree of match. The items in the

input data space (seismic trace) are assumed to be in a vector format. If *k* is the dimension of the input space, then every node in the SOM grid holds a *k*-dimensional vector of weights.

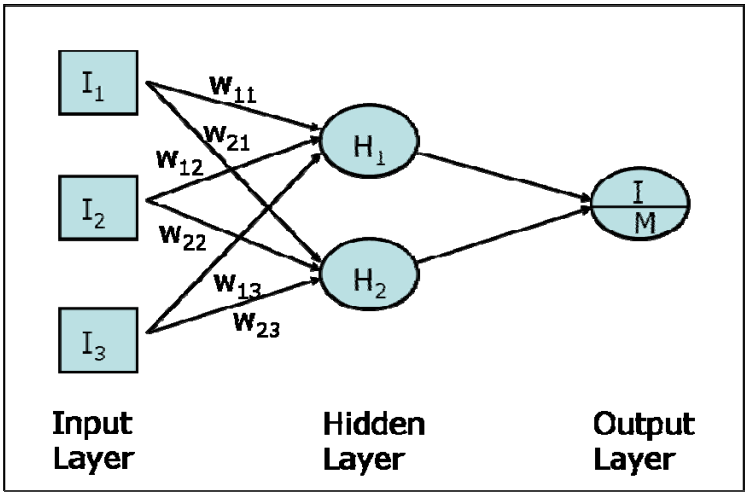


Figure 3.1 A schematic representation of the SOM method, which consists of three layers.

The SOM training process starts by inputting one trace from the input space into the neural network. Competition starts and all neurons compete to win the input trace. The neuron that most closely resembles the input trace wins the competition, and is rewarded by being moved closer to the input trace via the Kohonen learning rules (Kohonen, 1989). Therefore, each neuron migrates progressively toward a group of input traces. After some iterations the network stabilizes with each neuron at the centre of the cluster it represents.

Assuming an input vector $\mathbf{x}_i$ has *n* samples, $\mathbf{x}_i = [x_{i1},\ x_{i2},\ \cdots,\ x_{in}]^T$, there should be a weight vector $\mathbf{w}_j = [w_{j1},\ w_{j2},\ \cdots,\ w_{jn}]^T$ in the SOM space. The index *i* is the trace number in the input space and *j* is the node number in the SOM space. The learning process is competitive, that determines a winning node on the SOM map whose weight vector is similar to the input vector. The distance between the input vector and each node (weight vector) in the SOM space is measured using the Euclidean distance as

$$\ell_E \equiv \| \mathbf{x}_i - \mathbf{w}_j \| = \sqrt{\sum_{k=1}^{n} (\mathbf{x}_{ik} - \mathbf{w}_{jk})^2}, \quad (3.1)$$

The node or neuron *j* that makes the minimum Euclidean distance with the input vector *i* wins the competition. The weight vector of the wining node is also updated using the Kohonen learning rule (Kohonen, 1989):

$$\mathbf{w}_j^{new} = \mathbf{w}_j^{old} + \alpha(\mathbf{x}_i - \mathbf{w}_j^{old}), \quad (3.2)$$

where α is the learning rate having a value between 0 and 1. A big α value results in fast learning but unstable output. A small α value results in slow learning but stable output. Usually α is chosen high at the first iteration and decreased to zero toward the last iteration. The SOM algorithm uses equation (3.2) to update not only the winning node but also all nodes within a certain neighbourhood of the winning node. Figure 3.2 shows the training process of the SOM network with the aim to cluster the data points into three clusters. I randomly created three SOM nodes (yellow stars) to represent the SOM space.

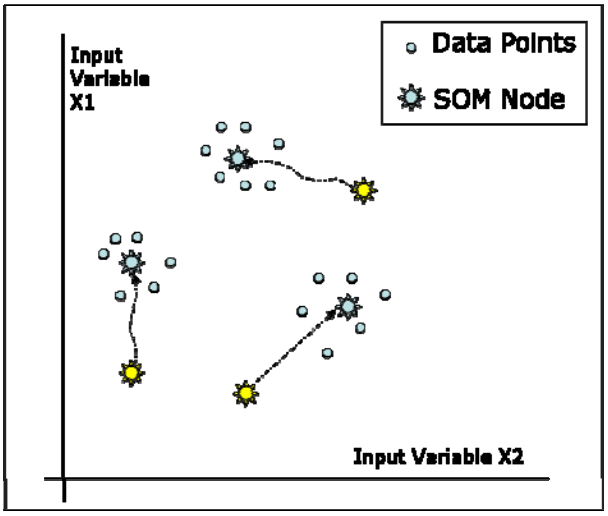


Figure 3.2. The training process of the SOM network.

In the training phase, the SOM node that is closest to the selected input data point is updated in the direction of the input vector. This process is continued until the changes are negligible. The final locations of the SOM nodes (cyan stars) define the clusters of

our data. In the application phase, the map is used to cluster any new seismic attributes, based on the previous knowledge gathered from the training phase.

3.2 The implementation of the SOM method

In the application phase of the SOM algorithm, each seismic input vector is compared with all cluster centre vectors (or SOM nodes) yielding two possible outputs: segments and match. A segment is the index of the best matching unit or winning cluster centre. A match is a measure of confidence between no confidence (value equal to zero), and input vector and winning cluster vector are identical (value equal to 1).

To validate my SOM code, I start with application on synthetic data, and then to real data to visualize the SOM results. I also compare my SOM results with the results produced from industrial software called OpendTect.

Synthetic models are created with different reflection styles. I show three SOM maps. Each SOM map is a 2D map with rows and columns representing the SOM lattice or grid. Each point on the SOM grid represents an SOM node or neuron.

(1) I use “count map” as the name of the first SOM map. The count map is a 2D map that shows a different colour on each SOM node, depending on the response of that node to the input data. Whenever an input data or trace is introduced to the SOM map, only one SOM node will respond and be chosen as the winning node. This winning node will gain one count each time it wins an introduced input trace.

(2) The second 2D map is called the U-matrix map. The U-matrix map is a unified matrix to represent the distance between clusters (Ultsch, 1993). After doing the SOM learning, the U-matrix map is generated by calculating, for each SOM node vector, the distance between the neighbour vectors and their average. The U-matrix shows the gradient between each node and not just the average distance. In the U-matrix map, the intensity of each pixel corresponds to the respective estimated distance.

(3) The third SOM map is the magnitude map. This is a SOM map that I generated using the magnitude of each vector on the SOM map after training. The magnitude map will show a clear view of different clusters on the input data, but most of the time these clusters are overlapped.

The first synthetic model is shown in Figure 3.3. I create these three layers (Figure 3.3a) to test if the SOM code can cluster the input data into three clusters. The 10x10 count map (Figure 3.3b) shows that only three SOM nodes responded to the input data. In another words, these three nodes are the closest values to the three input reflections. Figure 3.3c shows the U-matrix map. Since our input model is small, it is not very clear if the U-matrix can show different clusters here. The U-matrix shows different colours around each cluster; these colours are supposed to be the boundaries of clusters. Figure 3.3d shows the magnitude map of each cluster.

The second synthetic model shows a continuous reflection recorded from a different depth or an anticline (Figure 3.4a). Since I have many different types of clusters, the count map (Figure 3.4b) shows more responses from the SOM nodes (16 responses in this case). The U-matrix map (Figure 3.4c) shows the boundaries between the clusters. Again, in a small scale of a 10x10 grid, the boundaries are not very clear. The magnitude map (Figure 3.4d) tries to give each cluster a distinct colour.

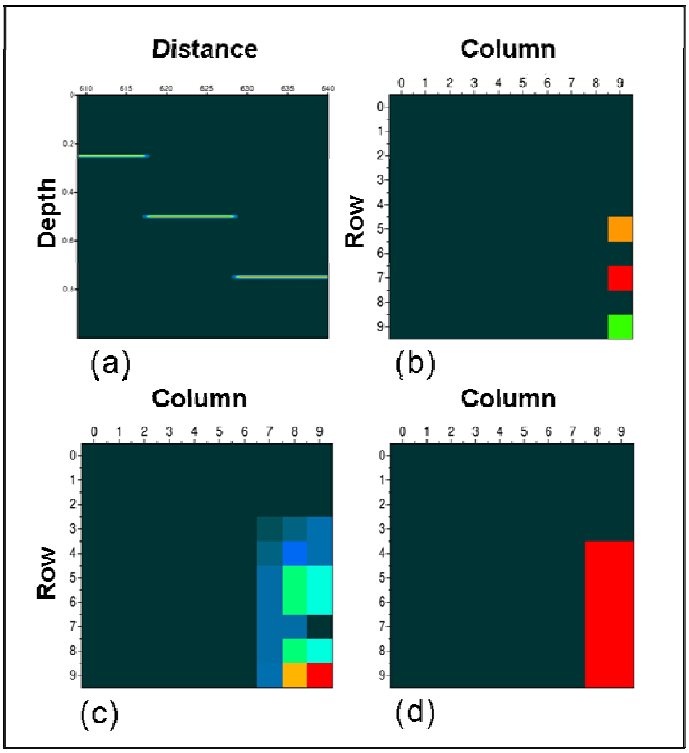


Figure 3.3 (a) A model with 3 layers. (b) The count map. (c) The U-matrix map. (d) The magnitude map.

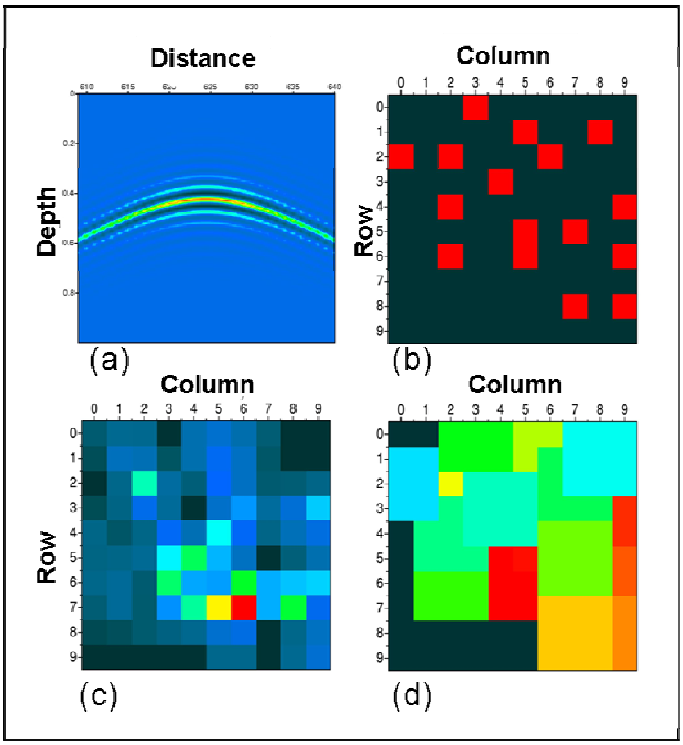


Figure 3.4. (a) An anticline model. (b) The count map. (c) The U-matrix map. (d) The magnitude map

I now use a real 2D seismic data as an example (Figure 3.5a). The 2D line consists of 32 traces, each with 0.5 s (501 samples). The input to the SOM algorithm is seismic traces, with the goal to cluster seismic waveforms. The SOM space is 45 rows by 45 columns. The count map (Figure 3.5b) with clear evidence that my input data have many different clusters. The U-matrix (Figure 3.5c) shows clearly the boundaries between the clusters. The magnitude map (Figure 3.5d) depicts many colours representing different clusters. I notice that the magnitude map shows many overlaps between clusters.

Now let me apply SOM in a small 3D seismic data set. Since using the full trace as an input is time consuming and can mislead the results, I now cluster the seismic amplitudes within a given time interval. Random traces are used as input vectors to the SOM. My time interval is between 1.680 s and 1.705 s. Figure 3.6a shows one of the input lines with the area of interest. Figure 3.6b shows the count map. I noticed that a small area of interest results in more clusters, as shown in the U-matrix and magnitude maps (Figures 3.6c and 3.6d).

After training the SOM algorithm with the 3D input in the previous example, I applied the results in the full 3D seismic data. Figure 3.7a shows one 2D line extracted from the original 3D volume. Figure 3.7b is the same 2D line extracted after applying the trained SOM on the 3D volume. Notice that Figure 3.7b shows a clustered version of the seismic data between 1.680 s and 1.705 s. This clearly shows that SOM was able to group and cluster the seismic traces inside the time range based on the previous learning and training.

Figures 3.8a and 3.8b shows a comparison between the two time slices extracted from the 3D volume before and after applying the trained SOM. The new time slice on Figure 3.8b shows more features after applying the clustering method. This is because the time slice is displaying a top view of the 3D data with traces being shown as points not vectors or full traces. After clustering and segmenting, Figure 3.8b shows more data because the traces were rearranged into groups and clusters.

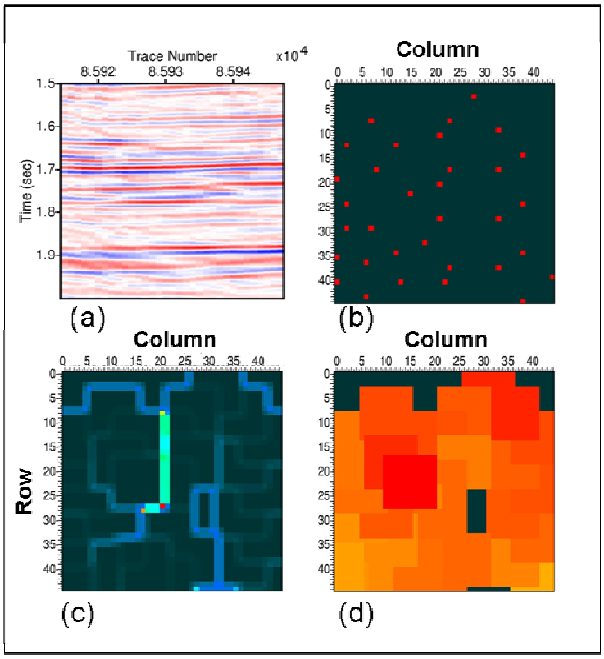


Figure 3.5. (a) An input real 2D line. (b) The count map. (c) The U-matrix map. (d) The magnitude map.

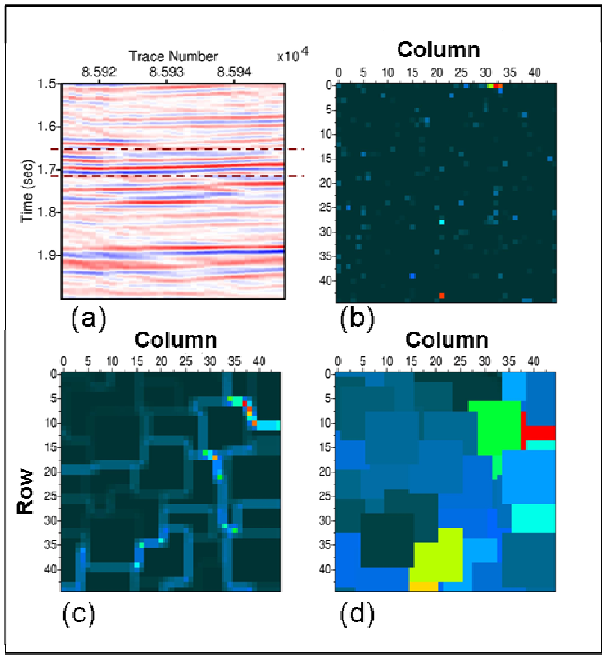


Figure 3.6 (a) A 2D line extracted from the input 3D volume (zone of interest 1.680 ms – 1.705 ms is shown). (b) The count map. (c) The U-matrix map. (d) The magnitude map.

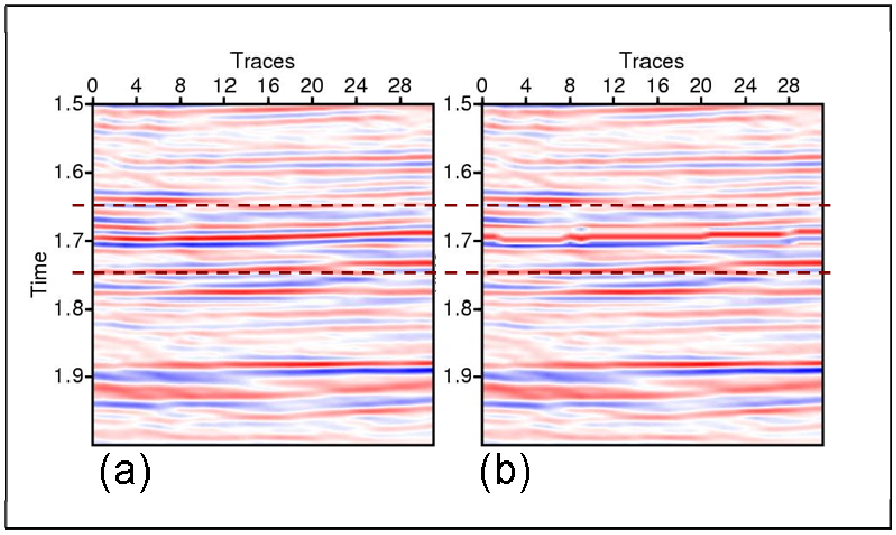


Figure 3.7 (a) Original input data. (b) The result of SOM code after applying the code to (a). Note the difference before and after the SOM algorithm, inside the area of interest.

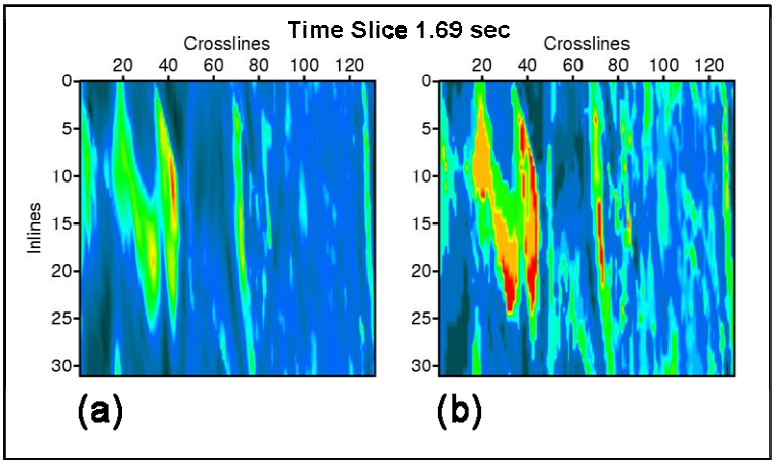


Figure 3.8 (a) A time slice (1.69 sec) extracted from the original input 3D volume. (b) The same time slice in (a) after applying the SOM code.

Now make comparison of my SOM to OpendTect SOM. OpendTect is a seismic attribute processing, interpretation and visualization system. The neural networks tool box is integrated as a plug-in to the OpendTect system. I obtained personal permission from the author of the demo data (via email) to use the data on my SOM code for the purpose of comparison. A 3D seismic data was acquired from the Dutch sector of the North Sea to explore for oil and gas in the Upper-Jurassic Lower-Cretaceous strata. The survey covers an interval of Upper-Jurassic Lower-Cretaceous strata. The upper part of the data (1200 ms) belongs to the Miocene. A large deposit of a deltaic package occurred in the area with an overall porosity of 20% to 30%. The deltaic package consists of sand and shale with some carbonate-cemented streaks. The seismic amplitude data was filtered using a dip-steered median filter to remove noise. Then data were inverted to acoustic impedance using the industry standard STRATA software. There are 4 available wells in the area with sonic and gamma ray logs. Two wells have density logs.

In the OpendTect tool box, the unsupervised network method is trained on a representative set of input vectors (attributes extracted at different locations) to find the cluster centres. Each cluster centre is then represented by a vector. In the application phase, each seismic input vector is compared with all cluster centre vectors yielding the possible segment. In the example given here, the unsupervised neural network method is used to visualize seismic patterns at one of the mapped horizons. The aim is to visualize

seismic patterns pertaining to the interval just below the mapped horizon. Figure 3.9 shows the 3D seismic data with the target horizon posted. The pink dots appearing on the horizon surface are the locations of the randomly selected input vectors. Each vector has 7 samples, three above the target horizon and three below it. The middle sample is chosen directly at the horizon surface. In the application phase, the 3D seismic volume is used as an input to produce the new clustered surface. Figure 3.10 shows the final clusters where areas with similar waveforms are visualized.

The same data and horizon were exported from OpendTect to be used on my SOM code. The SOM training is preformed with the same number of samples used for the OpendTect software. The result of my SOM code is shown in Figure 3.11. The two results shown in Figure 3.10 and Figure 3.11 are generally similar, except for the colours and some unclear events on the North West side of the map. Many parameters on the SOM code can be changed to produce an exactly identical picture. These parameters may include the learning rate, the neighbourhood radius, and number of iterations. Also, the scale of the two pictures and the colours used might make the two pictures look different. This result gives me an impression that my SOM code is reliable and can be applied in other seismic data.

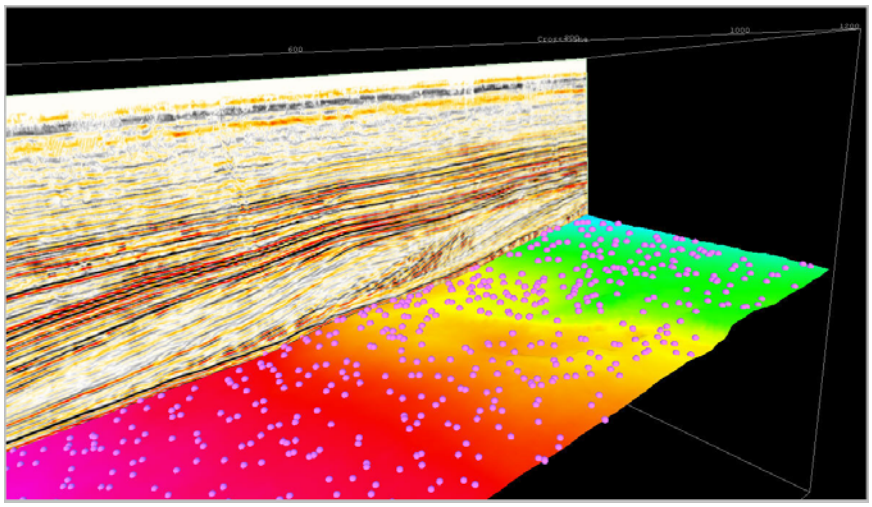


Figure 3.9. The input 3D seismic data with the target horizon posted. The pink dots are the locations of the randomly selected input vectors.

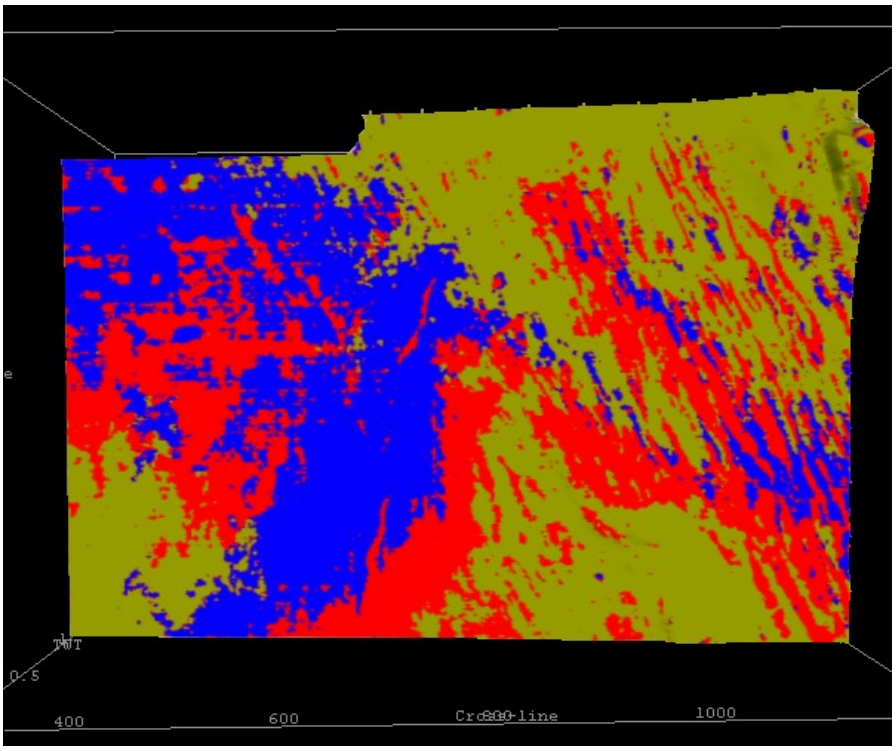


Figure 3.10. The final clusters of OpendTect neural network algorithm.

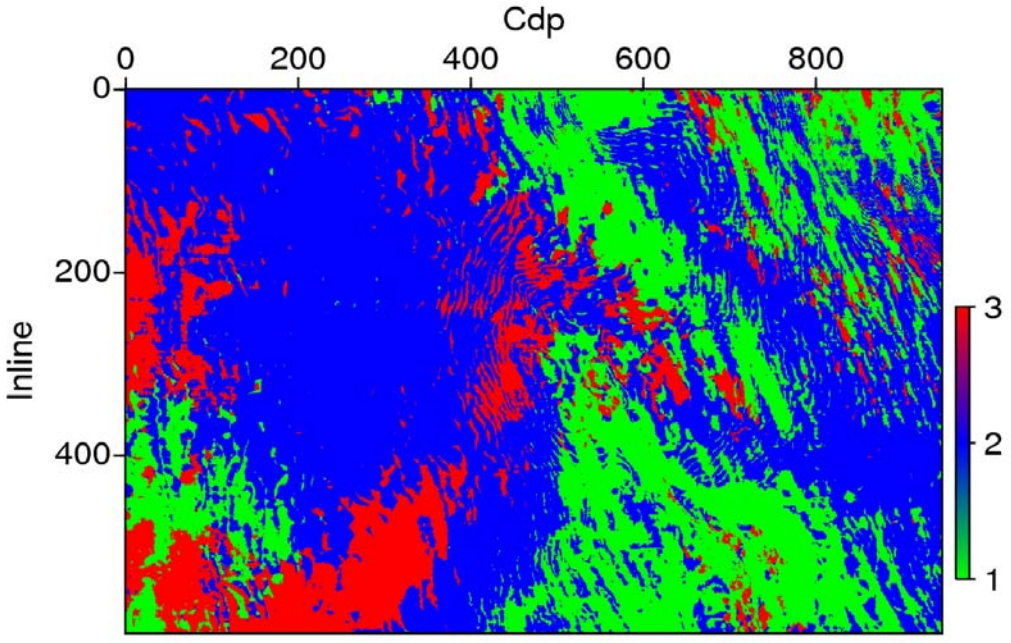


Figure 3.11. The final clusters of my in-house SOM code.

3.3 Clustering the SOM Map

As was shown early on this chapter, SOM performed a very efficient and robust job in summarizing the data. This can be seen from the results of the synthetic and real examples. Summarizing here means clustering, size reducing, and clarifying the data. The SOM clustered the input data by partitioning a dataset into a set of groups, classes or clusters. It reduces the input data size by transferring the high-dimensional input space onto a low-dimensional SOM grid. The visualization property of the SOM clarifies the input data by producing a new two-dimensional grid representing the clusters that exist in the data. However, this visualization characteristic of the SOM is only good in obtaining qualitative clustering information.

The SOM map has a good representation of the clusters and segments of the data, but I need to find out how many clusters I have and which vector or trace represents each cluster. Also, the SOM is a very good representation of the input data, but a quantitative description of the visualized SOM clusters is needed. This can be achieved by using some methods to select the interesting groups from the SOM. This can be called clustering or interpreting the SOM clustered map. Therefore, the full process will have two steps. The first step is to cluster the input data using the SOM algorithm. The second step is to cluster the SOM and find a representative prototype vector for each cluster. In this way, I reduce the computational load considerably by using only the cluster's representatives to represent the original data. This is done on the application phase and when I apply the new clustered SOM map to all of the data. Two clustering algorithms will be shown here with some examples. The clustering algorithms are the K-means and Fuzzy C-means clustering algorithms.

Clustering is the process of organizing similar objects into the same group. Each cluster should have members that are similar to each other and dissimilar to members from other clusters. Figure 3.12 shows an illustration of the clustering process. The clustering that was performed on the illustration was based on the shape and distance between the objects. Other clustering concepts might use any common relationship between the members of each cluster. Thus, the user or programmer might group the objects according to a specific descriptive clustering concept. The concept that I am using

on my clustering codes is usually based on the similarities between objects (traces). The similarities between the traces can be achieved by calculating the distance between the values of both traces. The distance will tell us the extent of the similarities between both traces.

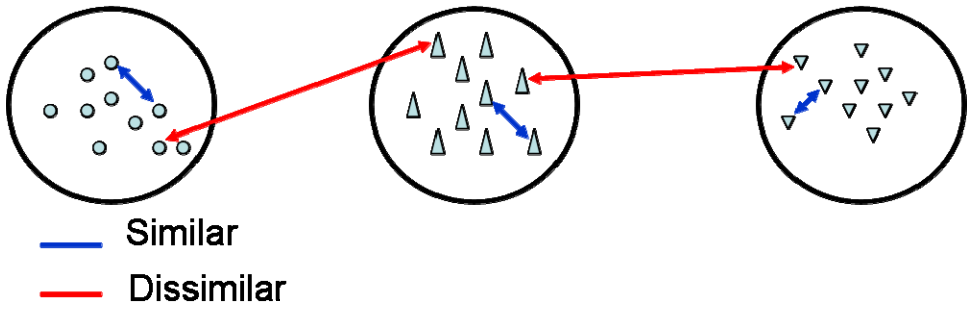


Figure 3.12. Clustering process.

The goal of grouping and clustering is to find the essential classification in a set of unlabelled data. The user is the one who can decide on which is the best clustering result. Sometimes the clustering product gives bad grouping that no one can interpret or use. On the other hand, this bad product might not be sufficient for one user but an acceptable result for another user. In my case, the clustering results should show a good relation to reservoir properties such as pore pressure and porosity. Applications of clustering include books in the libraries, document classification in the web pages, groupings of customers in marketing, and so on.

K-means (MacQueen, 1967) can be classified as one of the unsupervised clustering algorithms. Objects will be clustered into k number of groups or clusters. By saying K-means we are referring to the algorithm as the k representative of each cluster that holds the average or mean of all objects on the same group. The number of clusters (k) should be given before starting the clustering process. The main idea of K-means clustering is to predict a final k centre for clusters or a representative for each cluster. The algorithm begins in step one by randomly selecting a representative for each cluster. Step two is carried out to associate each data point on the clusters to the cluster representative. Step three is the recalculation of the positions of the clusters' representatives, based on the

associated data points. Steps two and three should be repeated until the clusters' representatives are no longer moving. The final clusters' representatives can be used to represent each cluster in the application stage. Figure 3.13 illustrates the k-means clustering steps.

The K-means algorithm tries to minimize the following squared error function or objective function as

$$E = \sum_{i=1}^{K} \sum_{j=1}^{m} \left\| \mathbf{w}_j - \mathbf{c}_i \right\|^2 , \qquad (3.3)$$

where K is the total number of clusters, $\mathbf{c}_i$ is the cluster representative, centroid, in cluster i, and m is total number of mapping vectors $\mathbf{w}_j$ on one cluster centered around $\mathbf{c}_i$. The denotation ||..|| means a distance measure between two points. In this research, I use the Euclidean equation to calculate the distances between points. The algorithm begins by randomly assigning a representative to each cluster. Step 1 associate each vectors $\mathbf{w}_j$ to different cluster representatives based on their distances. Step 2 recalculates each cluster representative that has minimum distances to any vectors within the cluster. These two steps are repeated until the cluster representatives are no more moving. Finally, for the display purpose, different colors are assigned to different clusters.

In the application phase, the SOM map is used to cluster the entire volume of the input seismic attributes based on the previous knowledge gathered from the input in the learning stage. Therefore, each trace of the input data volume is linked to a weight vector in the SOM map, associated with a color index.

Figure 3.14 shows one example of using k-means clustering with different numbers of clusters. From Figure 3.14, I noticed that the objective function is very high when there is only one cluster. Figure 3.14 shows that the objective function is about to be stable when I use more than four clusters. This helped me to decide on how many clusters we are going to choose for our input data.

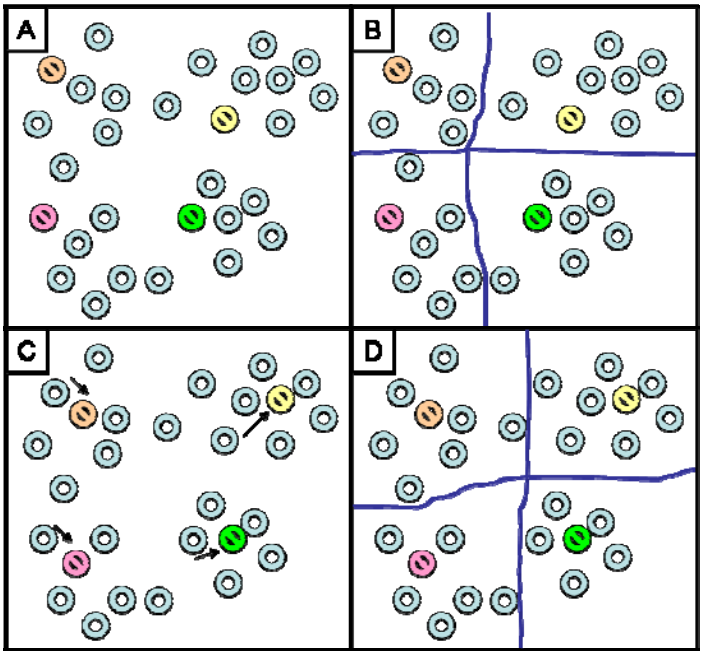


Figure 3.13. The k-means clustering process. (a) Random pick of clusters (4 cluster centres picked). (b) Association of data points to clusters. (d) Moving to new cluster location. (d) Association again. Process will stop when clusters are no longer moving.

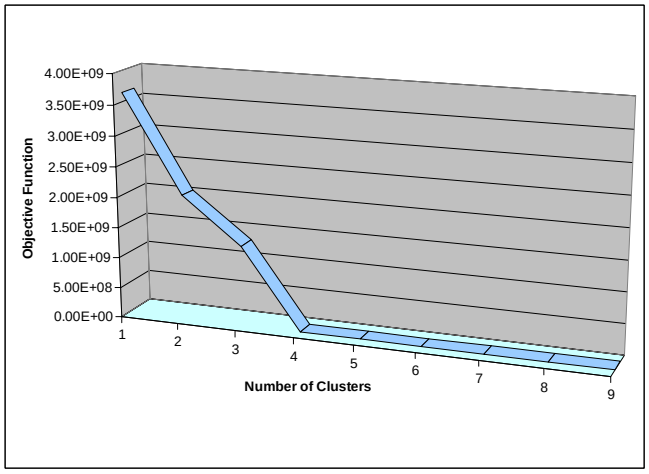


Figure 3.14. Curve produced by examining the objective function in k-means clustering.

Although the k-means clustering algorithm is extremely fast, the quality of the clustering results depends largely on the initial random set of clusters. One of the drawbacks of the algorithm is that it has to be told the number of clusters (k) before

clustering. This will force the user to try more than one initial number of clusters to reach the final result.

Fuzzy c-means (FCM) (Dunn, 1973) is one of the most widely used fuzzy clustering algorithms. The term 'fuzzy' denotes the fuzzy partitioning of the input data set. The algorithm will allow each of the input data to belong to more than one cluster at the same time. The algorithm is frequently used in image analysis problems and pattern recognition. The aim of the algorithm is to minimize the following objective function:

$$F_m = \sum_{j=1}^{n}\sum_{i=1}^{k} u_{ji}^{m}\left\|x_j - c_i\right\|^2 , \qquad (3.4)$$

where n is the total number of data points on one cluster and k is the total number of clusters. C_i is the cluster representative in cluster i that includes all data points x_j. m is a real number greater than 1 (m=2 in this research). In this objective function, a new matrix called degree of membership u_{ji} makes the difference between the k-means and FCM clustering algorithms. u_{ji} is the degree of membership of a data point x_j in the cluster i. The values on the membership matrix will go from 0 to 1. The FCM algorithm will optimize the objective function iteratively with updating (u_{ji}) the membership function and (c_i) cluster centres.

$$u_{ji} = \frac{1}{\sum_{r=1}^{k}\left(\frac{\left\|x_j - c_i\right\|}{\left\|x_j - c_r\right\|}\right)^{\frac{2}{m-1}}} , \qquad (3.5)$$

$$c_i = \frac{\sum_{j=1}^{n} u_{ji}^{m}\cdot x_j}{\sum_{j=1}^{n} u_{ji}^{m}} , \qquad (3.6)$$

I start the FCM algorithm by choosing the target number of clusters. Let us say the aim is to cluster the data into 4 groups. Then in step 1, a random 4 cluster representative is chosen from the input data. In step 2, the 4 random representatives will be used in equation 3.5 to substitute for $\left\|x_j - c_i\right\|$ (relationship between current data point x_j and current cluster representative c_i) and $\left\|x_j - c_r\right\|$ (relationship between current data point x_j

and all clusters from r = 1 to r = total number of clusters k). Step 2 will produce the membership matrix u_{ji}. In step 3, the new clusters' representatives will be calculated using u_{ji} and equation 3.6. Step 4 is carried out to compare the new clusters representative produces by equation 3.6 and the old representatives supplied to equation 3.5. If the difference between the new and old representatives is very small, then the algorithm stops. Otherwise, steps 2 and 3 will be repeated until the system converges. Figure 3.15 shows an illustration of the differences between k-means and FCM results. The blue dots under the x-axis are the input data points. In figure 3.15.a, k-means gives exact answers (0 or 1) when we try to find the relationship between all data points and the cluster II. On the other hand, the FCM in figure 3.15.b follows a curve or smoothed line to represent the different values of membership between the data and cluster II.

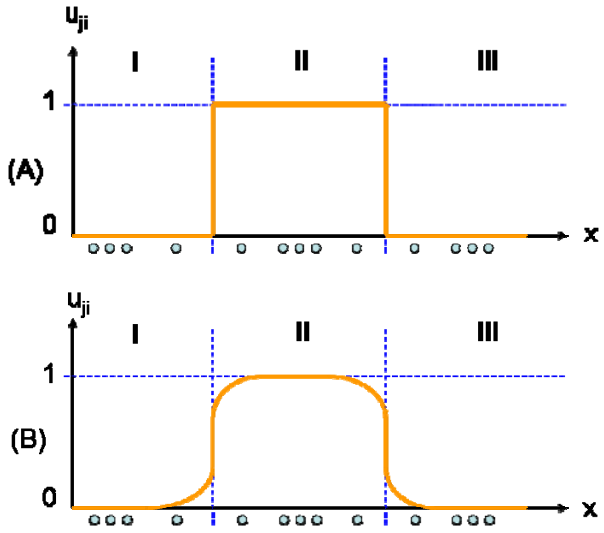


Figure 3.15. (a) K-means clustering gives membership values (u_{ji} = 0 or 1) between all data points and cluster number II. (b) FCM clustering gives smoother values for the membership with cluster number II.

The FCM clustering algorithm performs robustly and it was not very difficult in terms of programming implementation. One of the drawbacks of the system (as it was

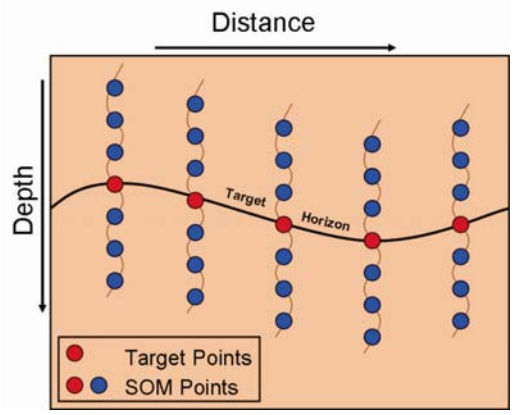
with k-means) is that it requires choosing the target number of clusters before starting the process. Also, FCM is sensitive to noise and outlying points on the data.

3.4 Application to carbonate data

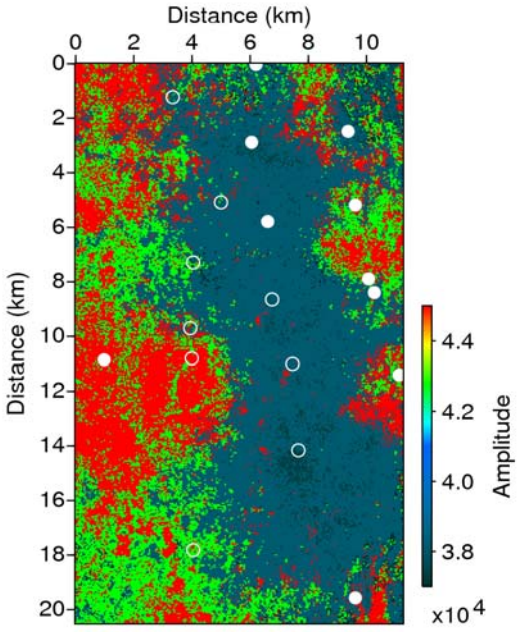
In this section, I apply the SOM algorithm to a real carbonate seismic data from Saudi Aramco Oil Company. Figure 2.8a (from chapter 2) shows the 3D seismic data with a horizon picked at the top of our target carbonate reservoir. A total of seven samples around the picked horizon are to be used in the SOM analysis. Seismic amplitude, seismic inversion, and 10 different seismic attributes are used as an input in the application phase. The inversion is preformed using the industry standard STRATA software.

The SOM performed very well in clustering the input data. The clustering was relatively fast and produced good results when applied to different data sets. The 3d seismic data under investigation cover the area of interest (1 sec) that showed overpressure information. In the following example, the SOM algorithm used the inversion seismic attribute and a horizon picked on top of the target formation to do the training. In the application stage, the SOM compared the input data vectors with the trained SOM vectors to do the clustering. This was direct compression using the Euclidean equation to establish how much difference the two traces have. Figure 3.16 shows the final clustering result after applying the trained SOM to all data points in the input 3d volume. Figure 3.16a shows the input to the SOM method during training and application. The input to the SOM method is not only the target horizon points but also 3 points above and below the target horizon. This gives the SOM engine enough data points with which to cluster and group the seismic inversion data. Figure 3.16b is the SOM clustering of the impedance data. Solid dots are wells with high pressure indication at the target horizon level. Hollow dots are wells with normal pressure. Noticed that figure 3.16b is not the final clustering result and the SOM will be clustered at the next step. The ANN is attending to find the nonlinear or random relationship between the seismic impedance velocities and the pore pressure values. Figure 3.16c shows a crossplot between the pressure and impedance values using 19 wells. I noticed that there

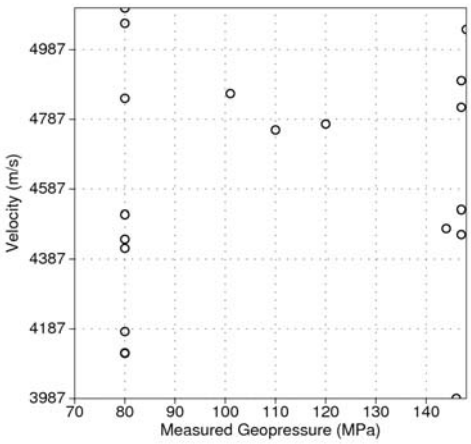
is no linear relation between the pressure and velocities in this data. For the pressure data around 80 MPa, these circles represent the normal pressure wells. The results shown in Figure 3.16 were good but not very reliable. Usually the carbonate reservoirs have very high velocities (6000 m/s). Also, if noise is present in the data, the trace-to-trace compression will not get the right answer.



(a)



(b)



(c)

Figure 3.16. (a) The target points are located directly in the target horizon. The SOM method uses both the red and blue points as the input for geopressure prediction. (b) Prediction of geopressure after using the SOM to cluster the impedance data. Note that this is not the final clustering and the SOM will be clustered again. (c) A crossplot showing the random relationship between the pore pressure and seismic impedance using 19 wells.

The next step is to cluster the SOM trained data before the application stage. The K-means and FCM were tested here. Both algorithms gave good results, and each one has its own way of clustering the data. The two algorithms were applied into the trained SOM after using the seismic inversion attribute as an input. Figure 3.17 shows indications of clustering. Figure 3.17a shows the different clusters inside the SOM lattice with colours representing each cluster. Figures 3.17b and 3.17c show the representative traces for each of the four clusters.

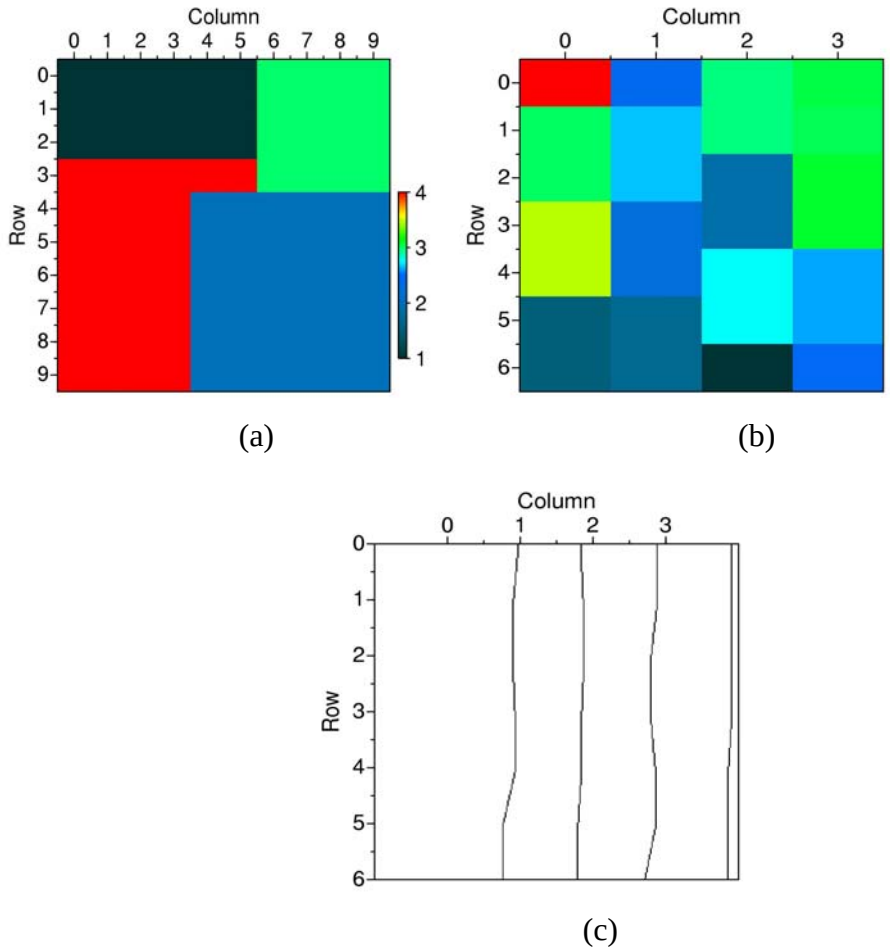
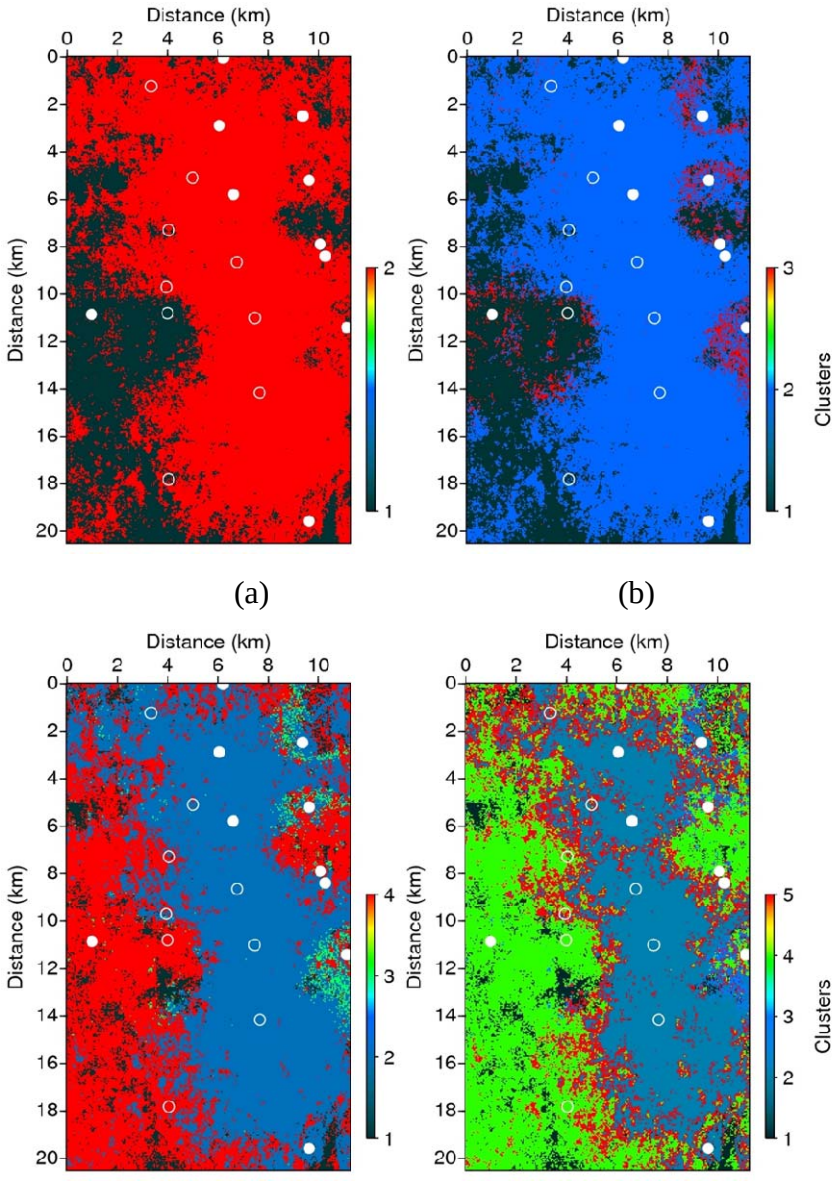


Figure 3.17 (a) different clusters inside the SOM lattice with colours representing each cluster. (b) And (c) show the representative traces for each of the four clusters.

Clustering was applied using 2 to 9 clusters. Figure 3.18 shows the 8 results. I then plotted the objective function (equation 3.3) versus number of clusters. Figure 3.19 shows the plot which indicates that the data have a limit of 4 clusters only. This is because the objective function tend to be stable when I used 4 clusters and above.


Distance (km)
Distance (km)
Clusters
(a)
(b)
(c)
(d)

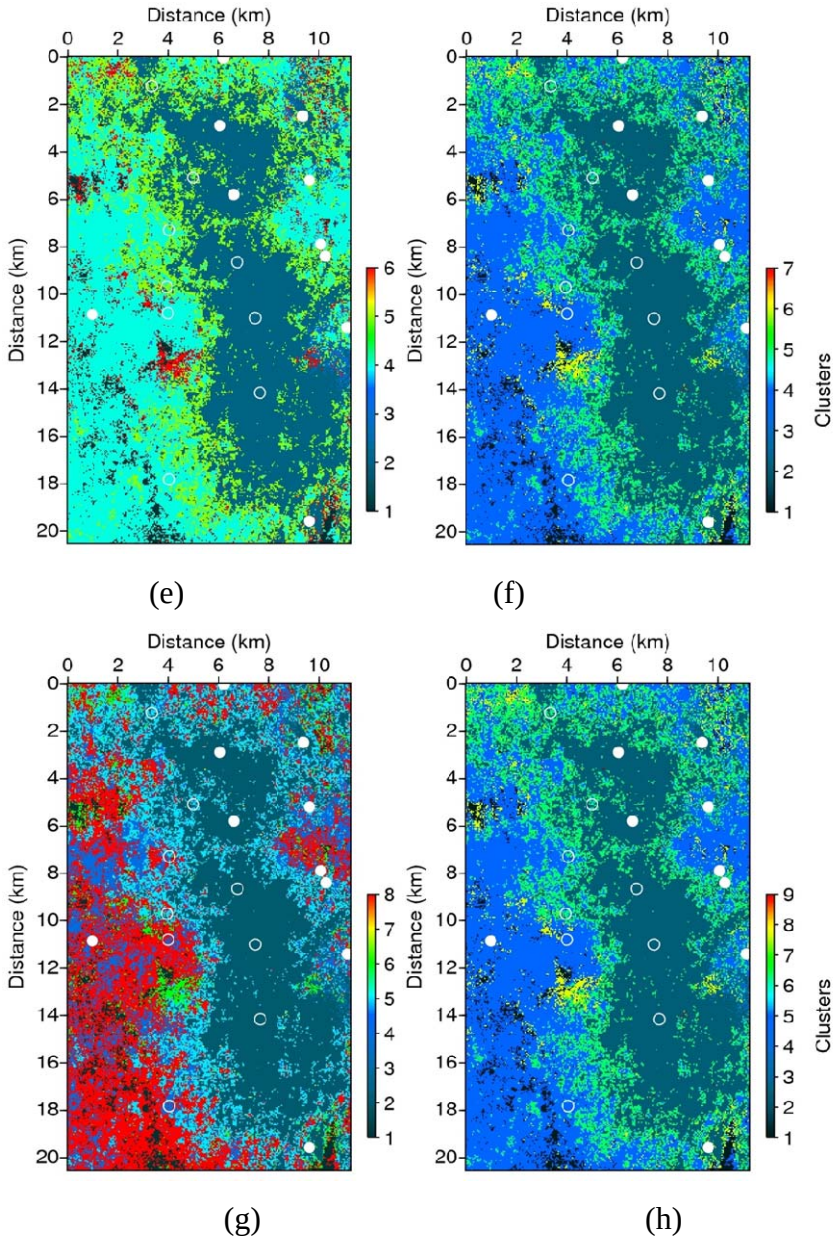


Figure 3.18 (a) Clustering of SOM using 2 clusters. (b) Clustering using 3 clusters. (c) Clustering using 4 clusters. (d) Clustering using 5 clusters. (e) Clustering using 6 clusters. (f) Clustering using 7 clusters. (g) Clustering using 8 clusters. (h) Clustering using 9 clusters.

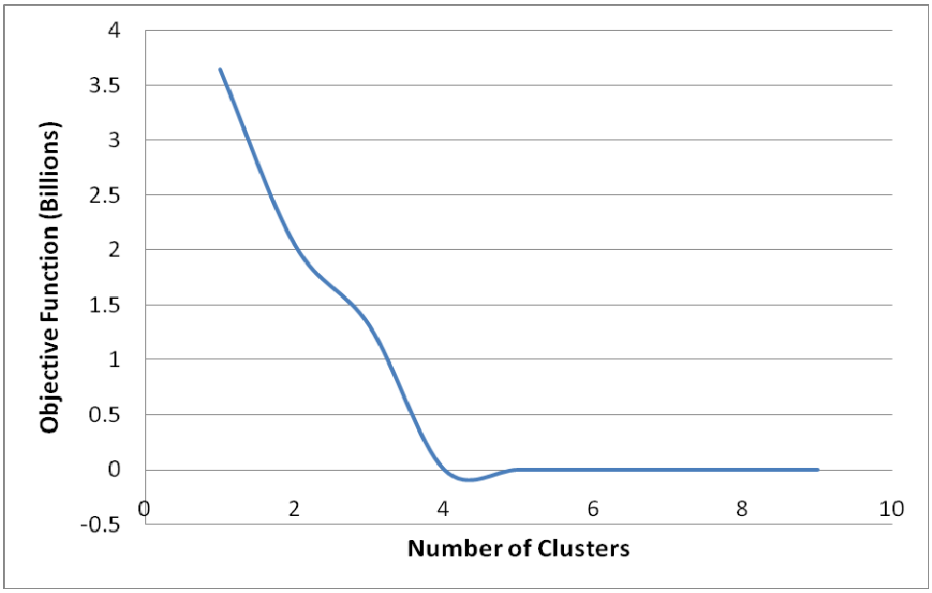
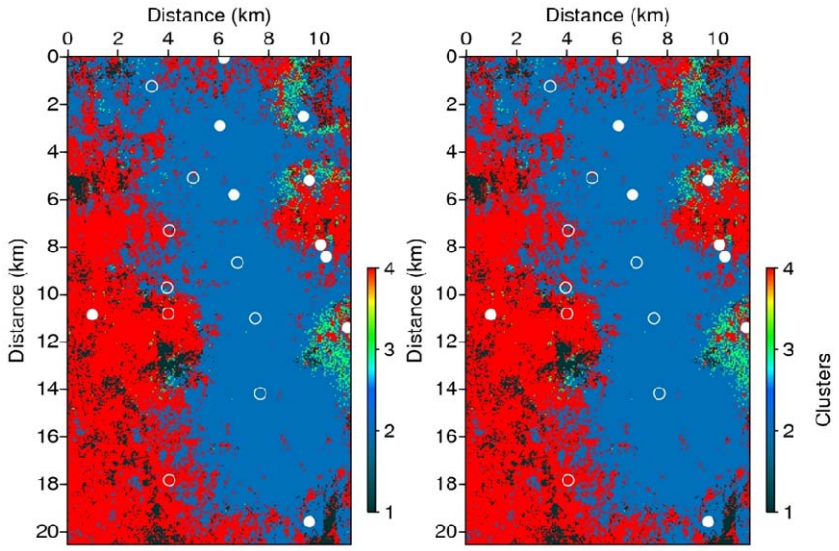


Figure 3.19 Curve produced by examining the objective function in k-means clustering with the carbonate data as an input.

Figure 3.20 shows the clustering of the k-means and FCM algorithms. The degree of membership that the FCM algorithm uses should show us more data clustering than the k-means. Both the k-means and FCM produced the same indication around the overpressure data (solid circles) and normal pressure data (hollow circles). This led me to conclude that in this specific dataset, using the k-means or FCM will not make a significant difference. Hence, one of them is sufficient to do the job.

From this experiment, seismic impedance showed some good relation to the high and low pressure. Normally, high pressured zones are more porous and hence have lower velocity and impedance. In Figure 3.20, the over-pressured wells (solid dots) are located under the same background or colour. Normal pressure wells (hollow dots) are located in a different segment or cluster. Figure 3.20 shows a good non-linear relation between the inversion and the geopressure obtained from the mud weights.



(a) (b)

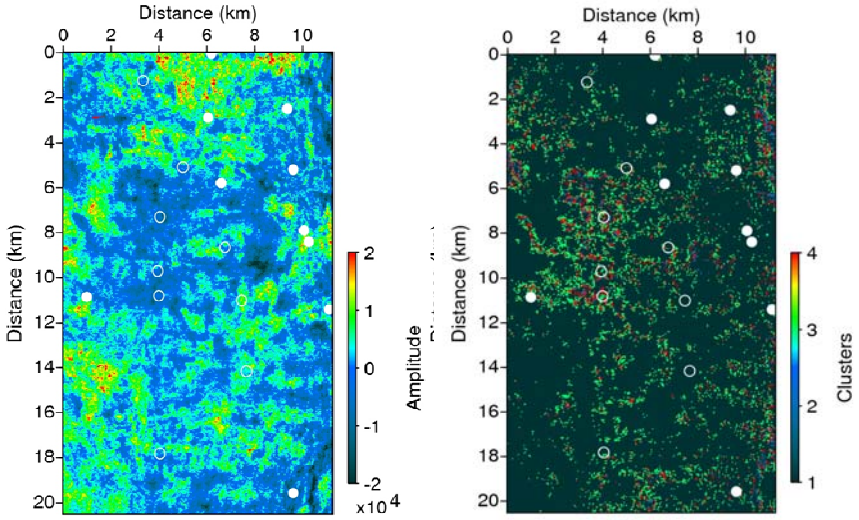
Figure 3.20. (a) Final clustering result of the k-means algorithm. (b) Final clustering result of the FCM algorithm.

3.5 Application of the SOM to other seismic attributes

In the previous section, SOM was applied into the seismic impedance. Here I will apply the SOM algorithm using more attributes including seismic coherence and seismic curvature. Coherence is a measure of similarity between two traces. It emphasizes discontinuities such as faults and stratigraphic changes. Curvature is a geometric tool that visualizes the faults and fractures of the reservoir on different scales. This is helpful in knowing the history of the tectonic deformation and depositional environment of the reservoir.

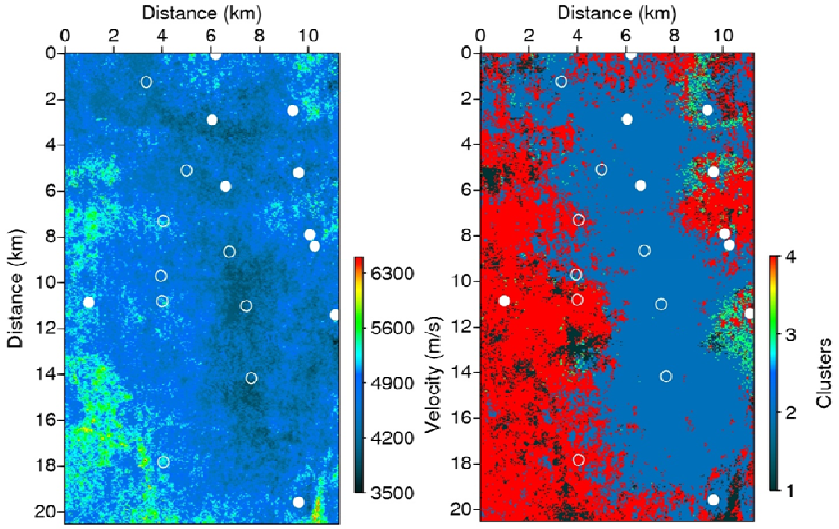
Figure 3.21 shows the result of clustering seismic amplitude and figure 3.22 is showing the clustering of seismic impedance. Figures 3.23 and 3.24 shows the result of clustering instantaneous amplitude and frequency weighted envelope attributes. Figures 3.25 and 3.26 shows the result of clustering thin-bed and attenuation attributes. Seismic amplitude and seismic impedance of the geopressured 3D data were used to produce the final coherence and curvature cubes. After that, the SOM algorithm was applied using both cubes, one at a time. Figures 3.27 and 3.28 show the final clustering results of the coherence and curvature. These two seismic attributes gave better consistency with the mud weight indications presented at the well locations (solid and hollow dots). The next

step is to find which seismic attributes gave the best indication of overpressure. The solution to this question is the final geopressure prediction of this dataset.



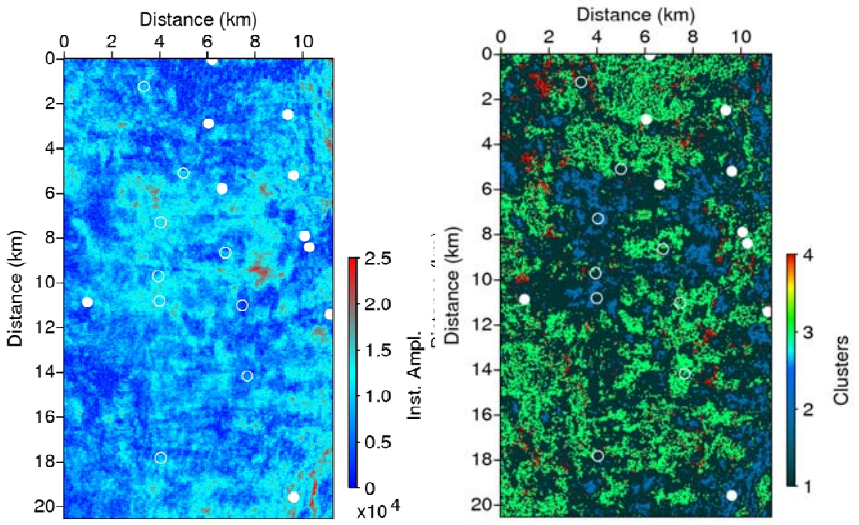
(a) (b)

Figure 3.21 (a) A horizon slice from the input seismic amplitude volume. (b) Final clustering after using the seismic amplitude as an input to the SOM code. Solid dots are wells with high pressure. Hollow dots are wells with normal pressure.



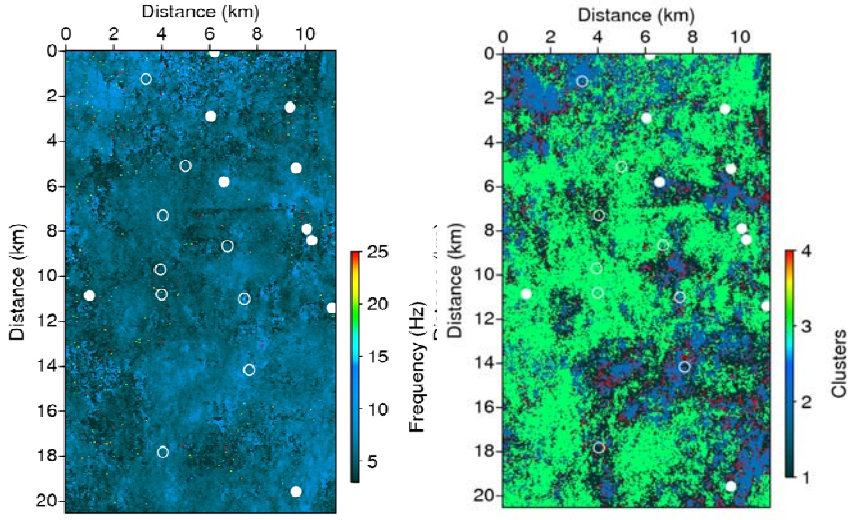
(a) (b)

Figure 3.22 (a) A horizon slice from the input seismic impedance volume. (b) Final clustering after using the seismic impedance as an input to the SOM code.



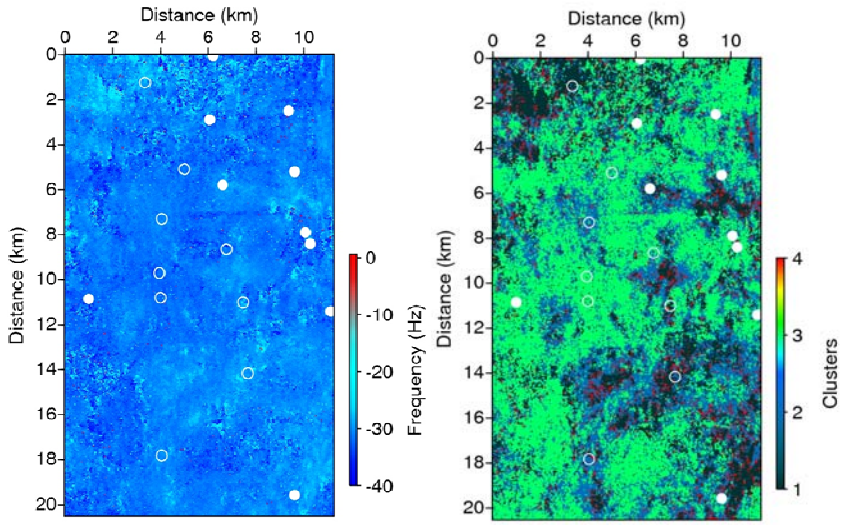
(a) (b)

Figure 3.23 (a) A horizon slice from the input instantaneous amplitude volume. (b) Final clustering after using the instantaneous amplitude as an input to the SOM code.



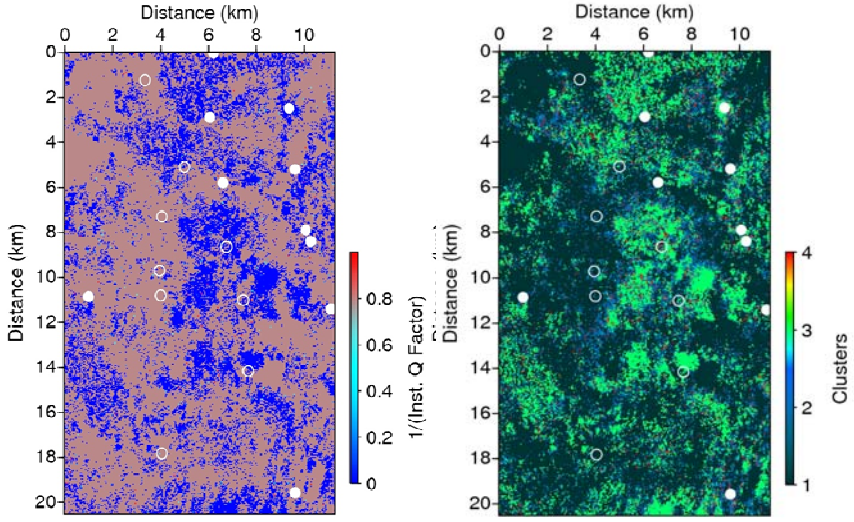
(a) (b)

Figure 3.24 (a) A horizon slice from the input frequency weighted envelope volume. (b) Final clustering after using the frequency weighted envelope as an input to the SOM code.



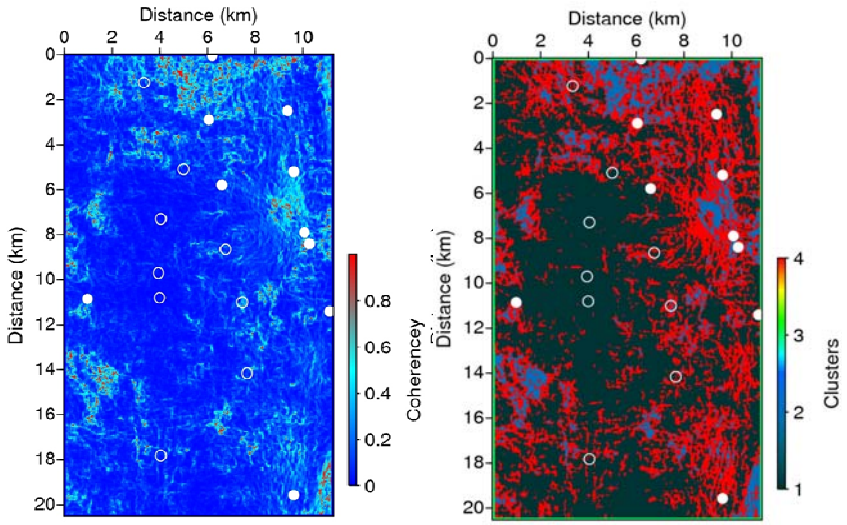
(a) (b)

Figure 3.25 (a) A horizon slice from the input thin-bed volume. (b) Final clustering after using the thin-bed as an input to the SOM code.



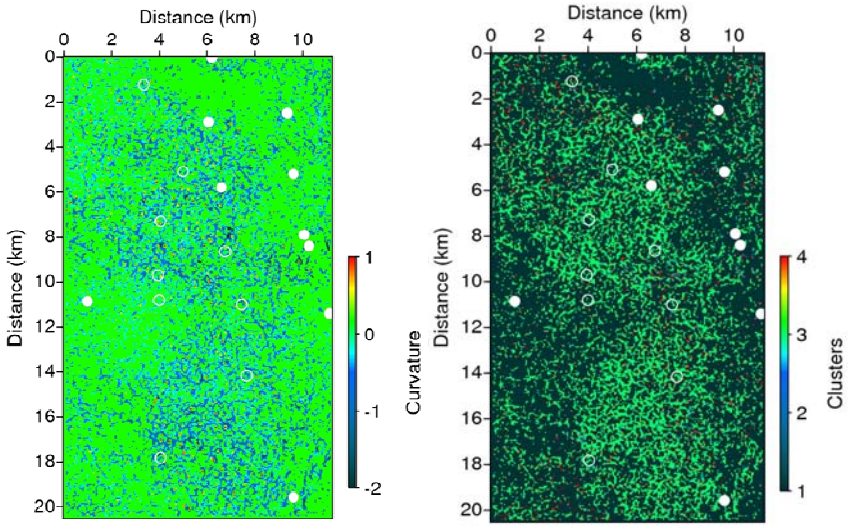
(a) (b)

Figure 3.26 (a) A horizon slice from the input attenuation (1/Q) volume. (b) Final clustering after using the attenuation (1/Q) as an input to the SOM code.



(a) (b)

Figure 3.27 (a) A horizon slice from the input coherence of seismic amplitude volume. (b) Final clustering after using the coherence of seismic amplitude as an input to the SOM code.



(a) (b)

Figure 3.28 (a) A horizon slice from the input curvature of seismic impedance volume. (b) Final clustering after using the curvature of seismic impedance as an input to the SOM code.

The science of seismic attributes depends most of the time on human visualization of which attribute might give the best indication. Curvature and coherence attributes, for example, are usually analyzed by checking if the extracted time slices have a better indication of fractures and faults. Here I will analyse the previous SOM clustering outputs visually, logically, and mathematically.

The first way to decide on which attribute prediction to use is to check the output horizon slice visually. The main purpose behind posting the high pressure and normal pressure wells was to check visually if the overpressured or normal wells lie under the same background. For example, having the same background colour under the overpressured wells (solid dots) might give a good indication that this background colour is showing overpressured zones. The same applies for the normal pressure wells. After applying the visualization check on the previous results (figures 3.21 to 3.28), I found that the seismic attributes that respond the most closely to overpressured and normal wells are the seismic impedance (figure 3.22) and seismic coherence (figure 3.27).

The second checkpoint is to think logically about the current prediction. Some of the results, like the seismic amplitude (figure 3.21), frequency weighted envelope (figure 3.24) and thin-bed (figure 3.25), show generally the same background colour everywhere on the area. This is logically not right, because we know that this area has some overpressured wells and some normal wells, while the predicted results cover the area widely with the same background colour.

The third checkpoint is to find a mathematical way to find the best seismic attribute used as an input. This is hard to achieve when using only a horizon slice of the data. Therefore, I will use the full trace that was used during the SOM training and application. The SOM used a trace of 7 samples in the training and application stages. As mentioned before (figure 3.16a), there is one sample at the horizon slice, 3 samples above, and 3 samples below the slice. Only the traces at the wells locations will be used. For each seismic attribute, each trace from a well (overpressured or normal pressure) will be compared with the four clusters in figure 3.17c. The comparison will be done using the Euclidean distance formula Equation 3.1. The purpose of this distance calculation is to find which clusters contribute the most to over-pressured or normal pressured wells. At the end of the calculation, I will have a table showing each attribute with the statistics of

clustering occurrence. For example, the seismic impedance gave 70% of overpressured wells under cluster number 2 and 22% of normal pressured wells under cluster number 4. Figure 3.29 shows a graph for the statistics of overpressured areas under each seismic attribute. In this graph, each seismic attribute will show 4 different bars. Each bar represents one cluster. The vertical axis is the occurrence (%) of geopressured wells in each attribute. Figure 3.30 shows the same statistics for the normal wells.

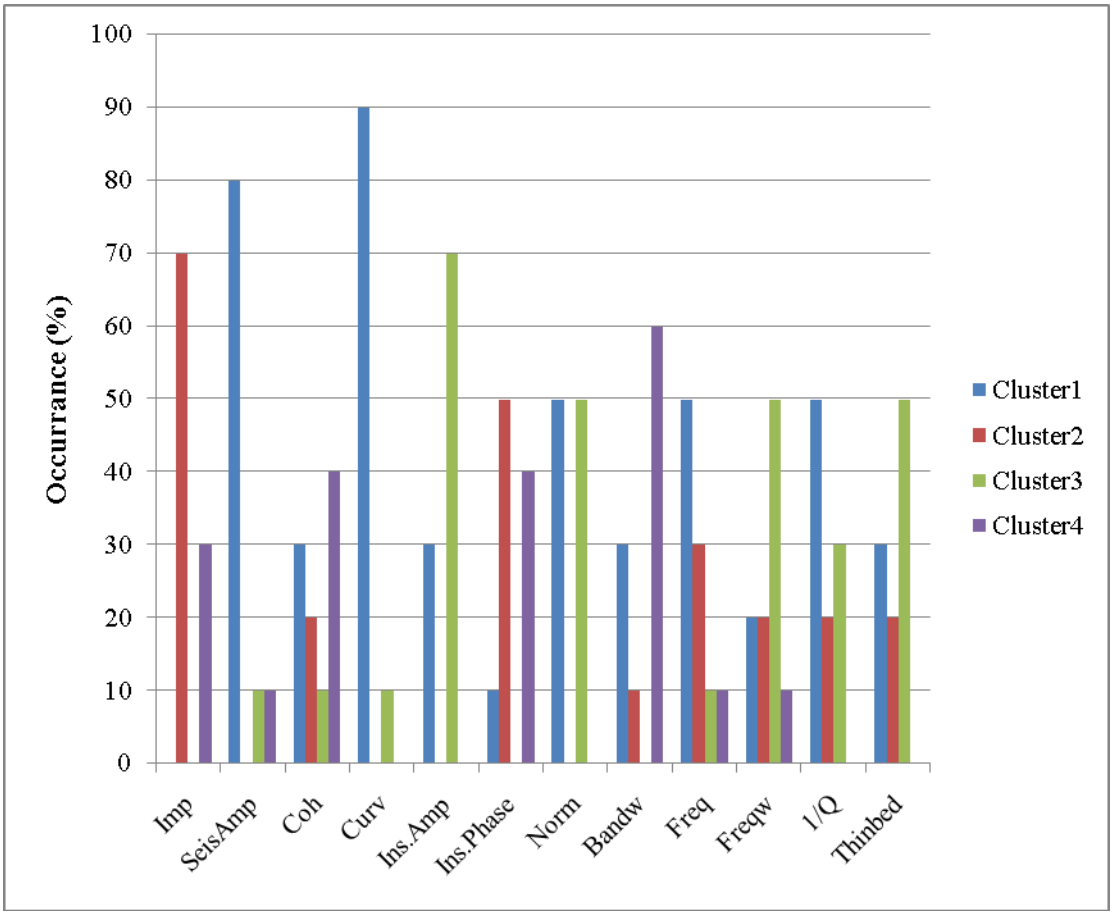


Figure 3.29 A graph for the statistics of overpressured areas under each seismic attribute. Colours are clusters (1-4).

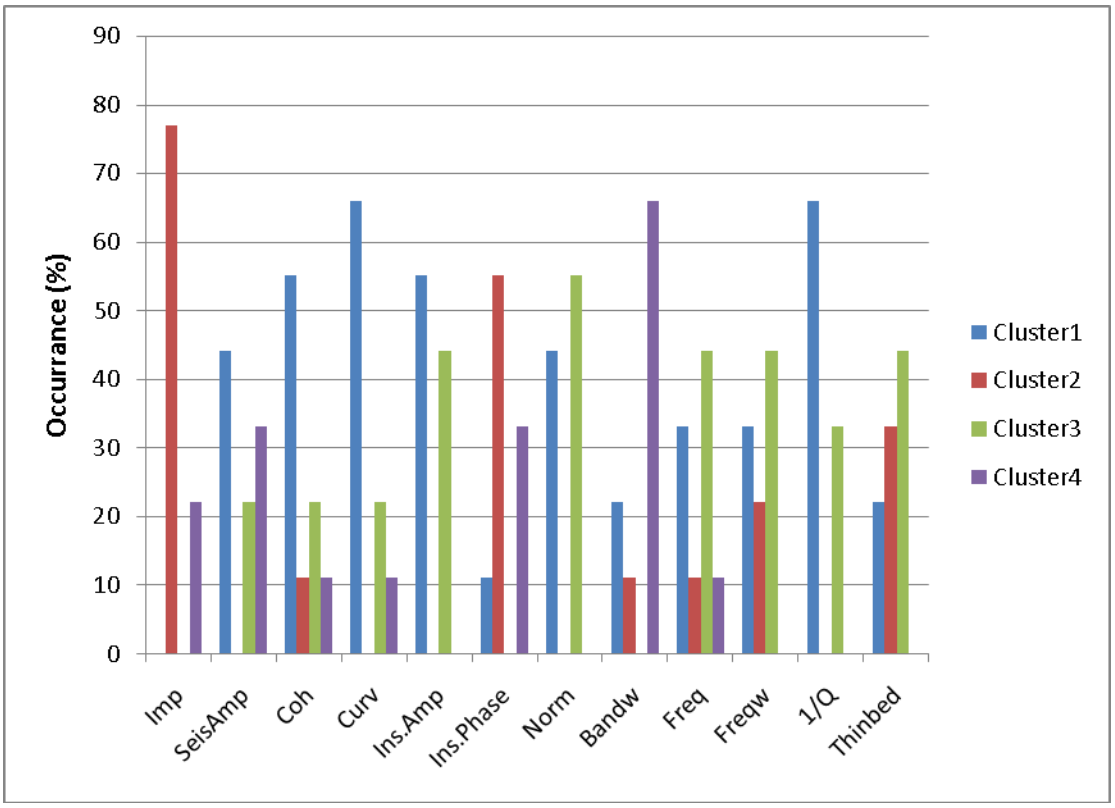


Figure 3.30 A graph for the statistics of normal pressured areas under each seismic attribute. Colours are clusters (1-4).

The last decision on which one of these attributes could win the prediction competition depends on the statistics of figures 3.29 and 3.30, and not just one of them. This is because some attributes gave strange occurrences or statistics. For example, some attributes like curvature, instantaneous amplitude, instantaneous phase, instantaneous bandwidth, and attenuation, gave a high occurrence at the same cluster location for both overpressure and normal wells. Curvature for instance, counts 90% occurrence of geopressured wells at cluster number 1, but also counts 66% for the same cluster number one with normal pressure wells. This is saying that both the overpressured and normal wells stay under the same background colour.

The seismic impedance statistical results showed that 70% of overpressured wells are in cluster 2. At the same time, 77% of normal wells are on the same cluster number 2. The coherence statistics showed that 40% of overpressured wells were under cluster 4,

and 55% of normal pressured wells were under cluster 1. Seismic impedance and coherence showed good indications visually and logically, but the coherence also showed a good indication statistically.

3.6 Summary

The main problem with the SOM neural network algorithm resides in choosing the correct parameters for the right input. There are very sensitive parameters, like the learning rate, the radius of the neighbourhood, number of iterations, size of input data, number of samples used, size of the SOM lattice (rows and columns) and the random initial weights of the SOM vectors.

(1) The choice of learning rate will affect the speed of learning and the stability of the final SOM weight vectors. A learning rate of zero results in slow learning by stable cluster positions. On the other hand, a learning rate of around one results in fast learning but unstable cluster position. One way to improve the performance of the SOM map is to start with an initial rate of one and reduce the rate toward zero during the learning. This will speed up the learning process.

(2) Another important parameter is the radius or size of the neighbourhoods. Choosing a high radius will slow the learning rate. To overcome this problem, I start with a large radius and reduce the radius size towards one during the learning process. The initial weights of the SOM vectors are randomly chosen at the beginning of the training. Choosing weight values around our input values will speed up the learning process.

The main objective of this study was to find a neural network method that can relate quantities that have non-linear relations for the purpose of predicting the reservoir parameters. The SOM classification is based entirely on the internal structure of the data, with no prior knowledge of the objects beforehand. The results have shown that the coherence and the impedance seismic attributes have a better response to pressure when compared with other seismic attributes. Pore pressure data were validated using 10 wells with overpressure information. Mud weights from the 10 wells helped, generally, in deciding which attributes are most sensitive to pore pressure.

The visualization property of the SOM algorithm may help us to separate the output data easily into small groups. However, when we want to produce quantitative

descriptions of the clustered data, other clustering methods like the k-means and FCM are needed. The K-means and FCM clustering algorithms produce good results when applied to the trained SOM data. Both of them are fast and robust. The FCM takes the advantage of assigning a degree of membership between all the input points and clusters' representatives. Seismic inversion data with high overpressure zones were used to test the SOM, K-means, and FCM. The K-means and FCM gave the same indications about high or low pressure zones.

Chapter 4

Supervised artificial neural networks

Artificial neural networks show high performance when dealing with uncertainty and non-linearity. They have the ability to continue operating as designed, despite internal or external changes which we call fault tolerance (Aminzadeh and Brouwer, 2006). Therefore, when we have incomplete or noisy input data, the results will not be deviated considerably from the true answer. Generalization is another feature that gives the artificial neural networks the ability to generalize from the input training data, so that the output is produced from samples that were never introduced to the network. These two characteristics above make the neural network methods in general superior to knowledge-based and rule-based expert systems (Saggaf *et al.,* 2003).

The main difference between the supervised and the unsupervised networks (such as the self organizing maps algorithm in the previous chapter) is the way the networks learn. In the former method, the network learns by introducing some examples to the algorithm and learning from them. In the latter method, the network is asked to learn by itself without having been given any prior information about the problem.

In this chapter, I implement a supervised neural network system based on the known multilayer perceptron method. First I introduce the method by showing the needed equations, topology, layering, and examples. Then I talk about training, learning, and the back-propagation technique. Thereafter, I discuss generalization and over-fitting or over-training problems.

The application section is divided into three parts. Part one (section 4.3) is about the data preparation and quality control (QC). Part two (Section 4.4) concerns the application to real seismic data using a supervised 2D trained operator. Part three (Section 4.5) is about the application of the algorithm using a supervised 3D trained operator.

Finally, I apply it to real data from Saudi Arabia and show the results, examples and figures for all applications.

4.1 Multilayer perceptrons

A multilayer perceptron is a feedforward artificial neural network model, which means that the process or flow of data goes in one direction, starting from the input neurons and ending at the output neurons. In this type of supervised neural network, we map sets of input data (such as seismic attributes) onto a set of appropriate output data (like porosity or pore pressure) (Haykin, 1999). A perceptron is the basic type of feedforward neural network, because the inputs feed directly into the output via weighted connections. Thus, the perceptron is the basic element of a neural network model, which has only two layers, namely the input layer and the output layer. In multilayer perceptrons, one or more layers (hidden layers) are added between the input layer and output layer. This gives the multilayer perceptrons the ability to solve more complicated problems. A three layers artificial network (input, hidden, and output) can model any arbitrary continuous function (Hornik *et al.,* 1988). Table 4.1 shows what each neural network layer might do during the process.

Table 4.1 The job of each layer in the MLP.

Network layer	Job
Input	Feeding the network with the input values in the training and application stages.
Hidden	Extracting and adjusting the non-linearity between the inputs and outputs.
Output	In the training: results from the hidden layer will be compared with the output values. In the application: delivering the final results.

Figure 4.1 shows a simple diagram of an artificial neuron. The most important part of this diagram is the processing element, which is composed of two operations. The two operations are summation and activation. The inputs are connected directly to the

processing element through weighted links or synapses. The weights in these links are equivalent to the strength of the corresponding link. These weights are randomly initialized at the beginning of the training, but at a later stage of the process, they change according to the training and learning of the network. The summation operator creates a weighted sum of all the inputs:

$$y_j = \sum_{k=1}^{n} w_{kj} x_k \quad , \qquad (4.1)$$

where y_j is the weighted sum at neuron j, x_k is the input sample k within the total of n samples in the input vector, and w_{kj} is the weight of the link between them.

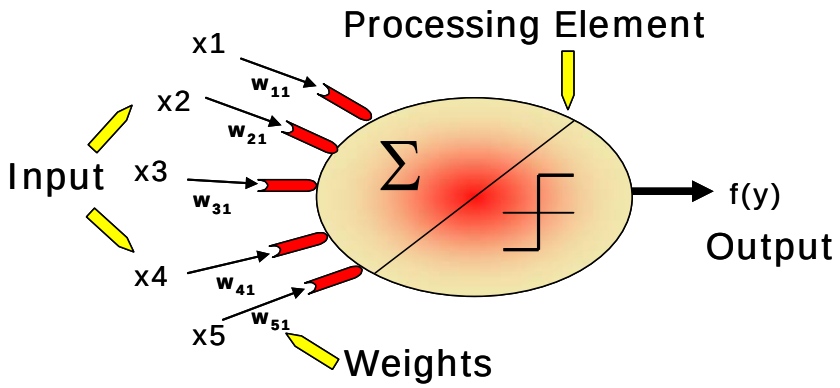


Figure 4.1 An artificial neuron (perceptron). The processing element is composed of summation which creates a weighted sum of all inputs and activation which is a non-linear operator to transfer the summation into a desired output.

The second important part of the processing element is the activation operator or function. This is a non-linear operator, also called a transfer function, transferring the summed input data into the desired output. Hence, after joining the two operators (summation and activation), we obtain the final output of the neuron:

$$\upsilon = f[\sum(w_i x_i)] = f[y] \qquad (4.2)$$

where υ is the final output of the neuron and y is the summation result of equation 4.1. The activation function (*f*) has the effect of squashing the infinite range of the

summation (*y*) into the desired operator range (example, -1 to 1). There are many activation functions commonly used, depending on the design of the artificial neural network problem.

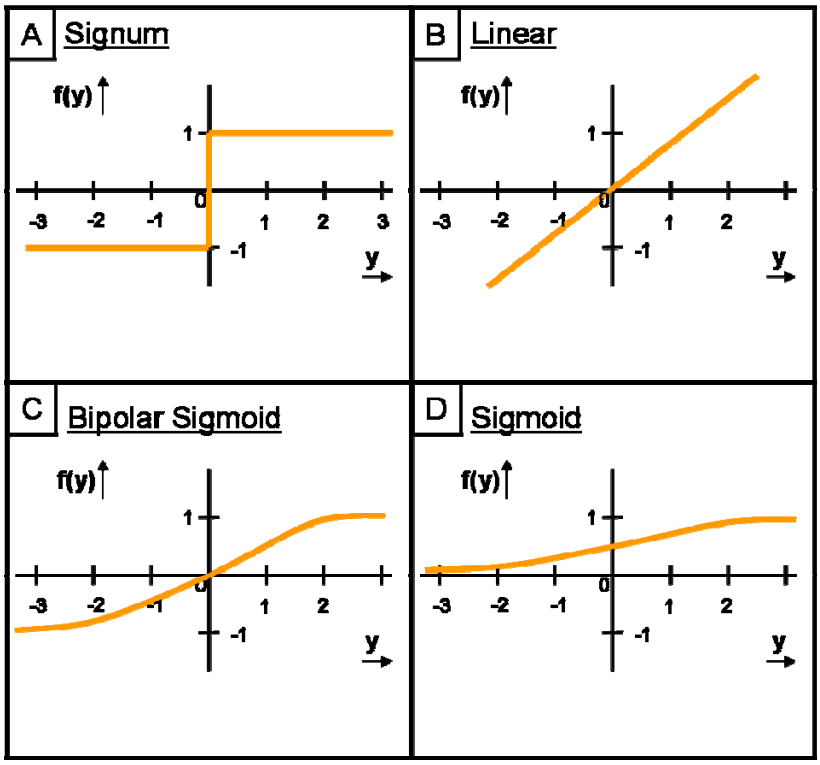


Figure 4.2 Four activation functions. (A) and (C) are producing a data limit between -1 and +1. (B) The linear function that does not change the summation values. (D) Sigmoid function that squashed the infinite range of input into the range 0 to +1

Figure 4.2 shows four examples of the activation functions.

(a) The signum function, which is a bipolar function (-1 < output < +1). When the input to this function is negative, the output is -1, and when the input is positive, the output is +1.

(b) The linear activation function which accepts the summation result of equation 4.1 without any change to it being required.

(c) A bipolar sigmoid, and the output results are between -1 and +1, depending on the input.

(d) Researchers usually use the sigmoid function when they deal with a multilayer perceptron algorithm. The sigmoid activation function is a unipolar operator which squashes the infinite range of input into the range 0 to +1. The sigmoid function has the convenient property whereby its derivative takes a simple form. This property will be very useful when we correct the weights of the links in the backward propagation.

Table 4.2 shows the mathematical expressions for signum, linear, bipolar sigmoid and sigmoid activation functions, respectively. Also, the derivative of the sigmoid activation function is shown. After showing the simple perceptron, Figure 4.3 shows the topology of the multilayer perceptron with three layers and the summation and activation operators.

Table 4.2: Mathematical expressions for some activation functions.

Activation Function	Equation
Signum	$f(y) = \begin{cases} -1 \xrightarrow{if} y < 0, \\ 0 \xrightarrow{if} y = 0, \\ +1 \xrightarrow{if} y > 0. \end{cases}$
Linear	$f(y) = y$
Bipolar Sigmoid	$f(y) = \frac{2}{1+e^{-y}} - 1$
Sigmoid	$f(y) = \frac{1}{1+e^{-y}}$
Derivative of Sigmoid	$f'(y) = \frac{1}{1+e^{-y}} * \left(1 - \frac{1}{1+e^{-y}}\right)$

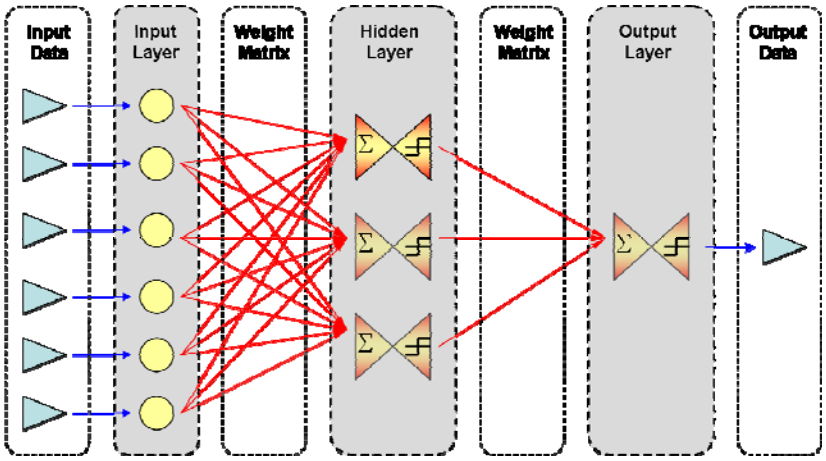


Figure 4.3 General topology of the multilayer perceptron. The input data are normalized for better data handling and then sent to the input layer. The weights between the input and hidden layers and between hidden and output layers are trained using a back-propagation method. Each perceptron is composed of two operations: summation and activation.

4.2 Learning, training and back-propagation

Learning is the main process that teaches the multilayer perceptron network how to deal with any new data. The networks are trained on a representative dataset. In this research, these examples can be pore pressure from mud weight or porosity from well logs. The corresponding input representatives are extracted from seismic data or any seismic attribute data, such as impedance, coherence, curvature, instantaneous amplitude, instantaneous frequency, etc. The training process starts from the input layer by multiplying the value of each input neuron (seismic data) by a certain connection weight. The weighted sum of all connections to each neuron in the hidden layer is then gathered. Then the activation function of each neuron in the hidden layer is calculated. Now the hidden layer is our new input layer toward the output layer. The same process that occurred between the input and hidden layers will be repeated between hidden and output layers. The final calculated result of the output layer will be compared with the desired output (pore pressure or porosity) and the error is calculated, after which the learning process will take place. Figure 4.4 shows the full process of training and learning. The

training process will continue to minimize the error between the predicted network result and the known output by adjusting the weights of the connections. One training cycle (or epoch) will occur when the entire training data is presented to the network (Haykin, 1999).

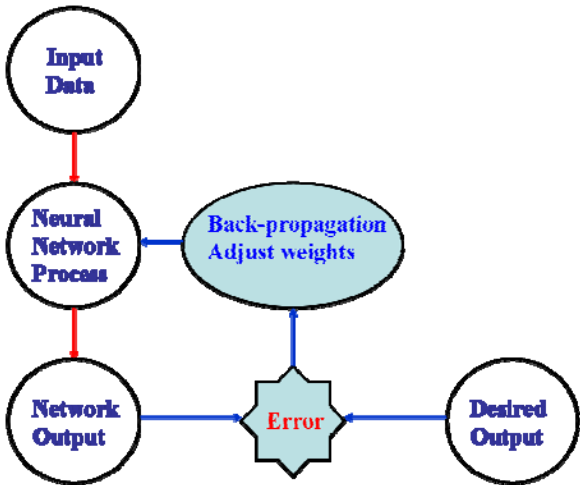


Figure 4.4 Learning process in the multilayer perceptron networks. The neural network output is compared to the desired output and the back-propagation will adjust the network weights.

A back-propagation learning algorithm is usually used with multilayer perceptron supervised neural networks. It is used to change the weights connected to either side of the neurons of the hidden layer. The algorithm uses a computed output error to change the weight values in a backward direction. The learning algorithm ends when all network outputs match their desired outputs, or the error is reduced to a specific threshold.

Table 4.3 shows my code to initialize the weights. This step is done only once at the beginning of the neural network process. The purpose of this step is to assign values randomly to the weights at each connection or link between the nodes. Choosing good initialization values for the connection weights can be of tremendous help in a successful network design (Haykin, 1999).

Table 4.3. My code for random initialization of the weights.

```
10      /* initialize WeightIH and DeltaWeightIH */
20      for( j = 0 ; j < NUMHID ; j++ ) {
30      for( i = 0 ; i < NUMIN ; i++ ) {
40      DeltaWeightIH[i][j] = 0.0 ;
50      WeightIH[i][j] = 2.0 * ( rando() - 0.5 ) * smallwth ;
60      }
70      }
80      /* initialize WeightHO and DeltaWeightHO */
90      for( k = 0 ; k < NUMOUT ; k ++ ) {
100     for( j = 0 ; j < NUMHID ; j++ ) {
110     DeltaWeightHO[j][k] = 0.0 ;
120     WeightHO[j][k] = 2.0 * ( rando() - 0.5 ) * smallwto ;
130     }
140     }
```

At the beginning of Table 4.3, I start the code by initializing the weights between the input layer and the hidden layer (WeightIH matrix). The loop will go over all input nodes (Numin) and all hidden nodes (Numhid). The weights will be initialized using a random function (rando), which will give a random number between 0 and 1. This limit will be then changed to make the value between -1 and 1 by subtracting 0.5 and multiplying by 2. Then the resultant random number will be adjusted to fall between a negative and positive small number (smallwth or samllwto). I give different initialization for the weights between the input and hidden layers in one side and the hidden and the output layers in another side. Smith (1997) recommends using different weights between different layers. This will place the initial values of the weights in the same order of

magnitude as the final weights. In my code, I initialized the weights between -0.0005 and 0.0005 for the weights between the input and hidden layers. For the weights between the hidden and output layers, the values were between -0.05 and 0.05. Lines number 40 and 110 are used to initialize the delta weights (DeltaWeightIH and DeltaWeightHO) which will be used later in the back-propagation algorithm.

Table 4.4. Computation of hidden layer activations.

```
10   /* compute hidden unit activations */
20   for( j = 0 ; j < NUMHID ; j++ )
30   {
40   SumH[p][j] =  bias ;
50   for( i = 0 ; i < NUMIN ; i++ )
60   {
70   SumH[p][j] += Input[p][i] * WeightIH[i][j] ;
80   }
90   Hidden[p][j] = 1.0/(1.0 + exp(-SumH[p][j])) ;
100  }
```

Next, I proceed with the neural network process by multiplying the input data from layer one to the randomly initialized weights from Table 4.3, in order to produce the weighted summation at each neuron of the hidden layer. This process is shown in Table 4.4, where SumH is the total weighted sum of pattern *p* at node *j* of the hidden layer. Sometimes I use a bias number to make the summation start from a specific number. The bias could be thought of as a weight, and usually it is equal to 1. This is helpful in some cases but can be omitted, as in my case (Hagan *et al.*, 1996). After doing the multiplication and summation, the resultant values might vary depending on the input and weight values. The next operator comes to compress these resultant values into the limit

of the chosen activation or transfer function. Line 90 shows the signum activation function which will do this job. This step will be done after finishing the inner loop (Numin), which does the multiplication and summation for all of the input values corresponding to one hidden node or neuron. After assigning the value of the signum function to node *j* of the hidden layer, the process will continue doing the same job to the other nodes in the hidden layer.

Table 4.5 Computation of output layer activations and errors.

```
10   /* compute output unit activations and errors */
20   for( k = 0 ; k < NUMOUT ; k++ )
30   {
40   SumO[p][k] =  bias ;
50   for( j = 0 ; j < NUMHID ; j++ )
60   {
70   SumO[p][k] += Hidden[p][j] * WeightHO[j][k] ;
80   }
90   Output[p][k] = 1.0/(1.0 + exp(-SumO[p][k])) ;
100  Error += 0.5 * (Target[p][k] - Output[p][k]) * (Target[p][k] - Output[p][k]) ;
110  DeltaO[k] = (Target[p][k] - Output[p][k]) * Output[p][k] * (1.0 -Output[p][k]) ;
120  }
```

The next step is to calculate the output nodes activations and errors. Table 4.5 shows the code. The same computation that was done in Table 4.4 will be repeated here, except for two equations. These two equations are responsible for the calculation of the computed output error. This is the error that we calculate while the network is

propagating in a forward direction. This error then will help us in the next step when we do the back-propagation algorithm to update and adjust the weights. The network learns by minimizing this error of the network's actual outputs compared with the desired outputs. Line 100 simply shows the calculation of the sum squared error overall output nodes *k* and all training patterns *p*. 0.5 is used here to simplify the algebra in deriving the learning process.

One way to minimize this error is to look for the global minima on the error curve, using an optimization algorithm. The basic and standard algorithm is called the gradient descent. By doing so, we can compute how much the error has changed by a small change in each weight. This could be done by computing the partial derivative of the error with respect to weight. Then the weights will be shifted by a small amount in the direction that reduces the error. Line 110 from Table 4.5 shows the gradient of the error with respect to the weight. The choice of a good activation function helped us here to take the derivative of the signum activation function easily. DeltaO[*k*] is the derivative of the error at output number *k*. Target[*p*][*k*] is the desired output at pattern *p* and output number *k*. Output[*p*][*k*] is the signum activation result of pattern *p* at output number *k*.

Table 4.6. Back-propagation algorithm and updating the weights.

```
10      /* back-propagate errors to hidden layer */
20      for( j = 0 ; j < NUMHID ; j++ ) {
30      SumDOW[j] = 0.0 ;
40      for( k = 0 ; k < NUMOUT ; k++ ) {
50      SumDOW[j] += WeightHO[j][k] * DeltaO[k] ;
60      }
60      DeltaH[j] = SumDOW[j] * Hidden[p][j] * (1.0 - Hidden[p][j]) ;
70      }
80      /* update weights WeightIH */
90      for( j = 0 ; j < NUMHID ; j++ ) {
100     for( i = 0 ; i < NUMIN ; i++ ) {
110     DeltaWeightIH[i][j] = etah * Input[p][i] * DeltaH[j] + alpha * DeltaWeightIH[i][j];
120     WeightIH[i][j] += DeltaWeightIH[i][j] ;
130     }
140     }
150     /* update weights WeightHO */
160     for( k = 0 ; k < NUMOUT ; k ++ ) {
170     for( j = 0 ; j < NUMHID ; j++ ) {
180     DeltaWeightHO[j][k] = etao * Hidden[p][j] * DeltaO[k] + alpha *
DeltaWeightHO[j][k] ;
190     WeightHO[j][k] += DeltaWeightHO[j][k] ;
200     }
210     }
```

Table 4.6 shows the learning algorithm which is called back-propagation. The first two loops prepare the error change so that it can be migrated in a backward direction. At the end of Table 4.5, I calculated the derivative of the error (gradient) to quantify the amount of change that we need to apply to the weights. Now in Table 4.6, I use that

quantity to migrate the change in the reverse direction. This is because when the activity of a node in the hidden layer changes, it affects the activities of all the nodes connected to it from the output layer. Therefore, to compute the overall effect on the error, in line 50 I add together all these separate effects on the output nodes. This was done by multiplying the gradient of the error DeltaO[*k*] at the output node k by the weight of the connections WeightHO[*j*][*k*] between the hidden node *j* and the output node *k*. In the same way, the error between the input layer and hidden layer can be defined by taking the gradient again to know how much change we need to apply on the weights. Line 60 shows DeltaH[*j*], which is the gradient at each hidden node *j*.

The loops in lines 90-140 show the code of adjusting the weights between the input layer and the hidden layer. The loops in lines 160-210 show the same code but for adjusting the weights between the hidden layer in the output layer. Lines 110 and 180 are showing the two main equations to do the weights adjustments. In these two equations, I use two important constants, which are the learning rate (eta) and the momentum term (alpha). The learning rate is the gradient descent contribution. A low value of learning rate will make the training very slow. A large value of the learning rate will accelerate the training, but might sometimes prevent learning. Finding a good value of the learning rate will depend on the problem, and this can be adjusted after some trial and error. I use 0.5 for my eta parameter. Note that my code can have two different training rates: etah (learning rate between input and hidden layers), and etao (learning rate between hidden and output layers). This will give the user the ability to speed up learning and training.

The other important constant is the momentum (alpha). This constant effectively keeps a moving average of the gradient descent weight change contributions and thus smoothes out the overall weight changes (LeCun *et al.*, 1990). This value is usually between 0 and 1, and it is frequently a matter of trial and error to get an accepted alpha value. It was noticeable that the momentum term (alpha) speeds up the training and improves the stability of the network.

In the learning stage, we start with some training examples and then we use a back-propagation algorithm to compute the weights at the connections between layers. By doing so, the aim is to make the neural network generalize. The goal of generalization is to make the network produce appropriate outputs for inputs it has never seen before. In

the training stage, we teach the neural network how to do that by having sufficient mapping between the inputs and outputs. This could be achieved by good non-linear interpolation of the input data (Wieland and Leighton, 1987). Figure 4.5 demonstrates how the generalization algorithm may occur. The generalization will happen after learning the points marked as “examples”. In Figure 4.4a, the good generalization results could be seen by looking at the point marked on the curve as “generalization”. A good indication of generalization is when the neural network learned from the examples and gave successful results. This should happen, even if the input data are slightly different from the examples used to train the network.

On the other hand, when the neural network learns many examples, an over-fitting might happen. This over-fitting or overtraining will happen when the neural networks start memorizing the input examples. As shown in Figure 4.5b, when the neural networks memorize the examples it will give good results only when the same examples are used as an input. This might happen because of noise or sometimes because of the use of a large number of nodes in the hidden layer. An example will be shown in the application stage.

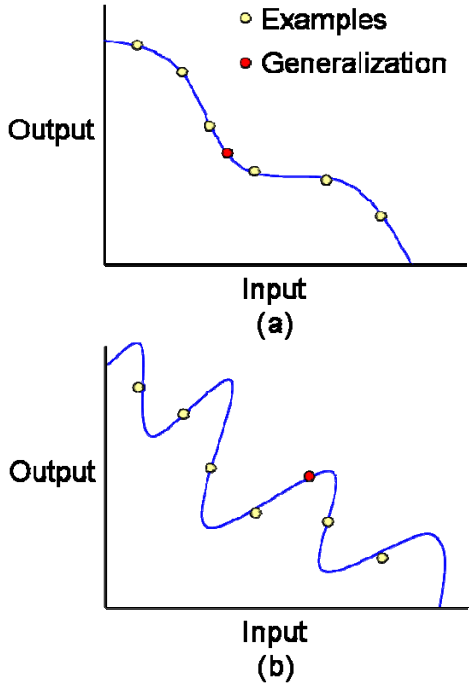


Figure 4.5. (a) A good generalization with proper non-linear interpolation between the examples (hollow circles). (b) A poor generalization with over-fitted nonlinear mapping.

4.3 Data preparations and quality control

After a successful training stage of the data using the multilayer perceptron method, now we come to the application stage. Many parameters that were changing in the training stage should be fixed and stable in the application stage. We stop using any learning rate, momentum term, or random weights initialization. The final weights that were generated and saved from the training stage are now ready for the new data in the application stage.

Table 4.7. Application code of multilayer perceptrons.

```
10      /* compute hidden unit activations */
20      for( j = 0 ; j < NUMHID ; j++ ) {
30      SumH[0][j] =  bias ;
40      for( i = 0 ; i < NUMIN ; i++ ) {
50      SumH[0][j] += Input[0][i] * WeightIH[i][j] ;
60      }
70      Hidden[0][j] = 1.0/(1.0 + exp(-SumH[0][j])) ;
80      }
90      /* compute output unit activations */
100     for( k = 0 ; k < NUMOUT ; k++ ) {
110     SumO[0][k] =  bias ;
120     for( j = 0 ; j < NUMHID ; j++ ) {
130     SumO[0][k] += Hidden[0][j] * WeightHO[j][k] ;
140     }
150     Output[0][k] = 1.0/(1.0 + exp(-SumO[0][k])) ;
```

Table 4.7 shows the application code of the multilayer perceptron method. Lines from 20 to 80 are used to compute the hidden nodes activations. The process uses an inner loop over the input nodes layer (Numin) and outer loop over the nodes in the hidden layer. Line 50 is used to compute the total weighted summation, and line 70 is used to compute the activation function at each neuron of the hidden layer. The final hidden layer values will be given to the next two loops (lines from 100 to 140) to compute the final output unit activations. Line 130 is used to compute the total weighted summation and line 150 to calculate the final activation function or the network output. In this section, I will show the real data application of the multilayer perceptrons neural networks method.

I will start with data preparation, and then I will apply the method using a 2D and 3D operators.

Data preparation is a very important step in any neural networks method. The data we use in the multilayer perceptron method are either the input vectors or the desired outputs. The desired outputs are used during training only. The input vectors are used as data representatives in the training stage and as regular inputs in the application stage. The same data scaling or standardization that we use in the training stage should be used in the application stage. This will make sure that the network gives the desired results. When we standardize the data before training, we tend to make the training process better and faster by improving the numerical condition of the optimization problem (Iglewicz, 1983).

However, standardization should be done with caution, because it might discard some necessary information. From this, one needs to know the data very well before doing any standardization. One example is when we have a different data range in the input values. This means that the data might have low values like 0.002 and high values like 2389 in the same input vector. In this case, we should choose the right standardization method to obtain better results. Standardizing the input data and forcing it to fit between 0 and 1 is one known method. However, in our case (the example above), this is not the right way to standardize the input data, because this might ignore some important information. In a real life example, a seismic attribute trace (impedance) could have values starting from 0 at the beginning of the trace but going up to 1400 meters per second in the lower parts of the trace. If we standardize the data between 0 and 1, the important high velocity numbers might be ignored. Therefore, another good standardization method is to obtain a standard normal random variable with mean 0 and standard deviation 1. This could be applied by subtracting the mean (sum of all input samples in the trace divided by number of samples) from each input vector value, and then dividing the result by the standard deviation (square root of summing all the differences between each input and the mean divided by number of samples). After trying both methods in my data, I found that the mean and standard deviation method was giving the right solutions.

4.4 2D operation

The 2D trained operator was built based on the multilayer perceptron neural network method. I called it a 2D trained operator, because the final result is only one horizon slice going above the target formation. The network in this method was trained by using 7 inputs from each representative trace at the well location. The corresponding desired output was only one sample at the horizon slice location. The goal behind using this simple method is to obtain a quick comparison of the results between the unsupervised method used in chapter 2 and the supervised method used here.

I use the same data (Saudi Aramco data) that was used in chapter 2. Geopressure data, in terms of occurrence or non-occurrence of abnormally high formation pressures across the target carbonate formation, are available from 19 wells. Among the 19 wells, only 10 wells have over-pressure and the remaining 9 wells have normal pressure.

The first application of the 2D trained operator was to predict geopressure and compare it with the results from chapter 2. The network was trained using 7 samples from each of the input 19 wells. The 7 samples were extracted from a seismic attribute (impedance). 3 samples were above the target horizon, one exactly at the horizon location, and 3 samples were below the horizon. Geopressure data (mud weight) were used as the desired outputs or targets at the horizon location. Figure 4.6 depicts the 2D trained operator. In this figure, the stars are the representative wells and the pink dots are the input acoustic impedance data. After training the network in all the representatives and reaching a minimum error between the network output and the desired outputs, we start the application stage. In the application stage, the full 3D volume was used as an input. 7 samples from each trace will be used to get a geopressure prediction using the trained network. Figure 4.7 shows the results. Figure 4.7a shows the geopressure prediction obtained by using the unsupervised neural network (SOM) that was discussed in chapter 2. Four clusters are shown in Figure 4.7a. Figure 4.7b shows the prediction of the supervised neural network method using the 2D trained operator. Note from the legend of both pictures that the supervised method is now using real geopressure values and not clusters. The supervised method clustered and segmented the data, while the supervised method just tries to give the real results that were used in training. This supervised

method attempts to give the real values of geopressures (Figure 4.7b), while the unsupervised method segments and groups the data (Figure 4.7a). These two prediction results are in a good agreement for some areas and are not in others.

In Figure 4.7 the solid dots indicate an over-pressure well location and the hollow dot indicates a well with normal pressure; this information was supplied during drilling the wells. SOM (Figure 4.7a) classifies the input based on seismic impedance, as high or low impedances produce different seismic signatures. The supervised neural network (Figure 4.7b) utilizes also mud weight values, in addition to seismic impedance, to predict the non-linear relationship. But mud weights values can only be collected (while drilling) at the horizon location and not as a complete well log below and above the target horizon. Figure 4.7c is a crossplot of the predicted geopressure using supervised neural network versus the measured pressure from mud weights. In Figure 4.7a and b, although most of the over-pressured wells are located under the same background or color, the figure cannot show a clear relationship between the seismic impedance and the geopressure we able to get from the mud weights.

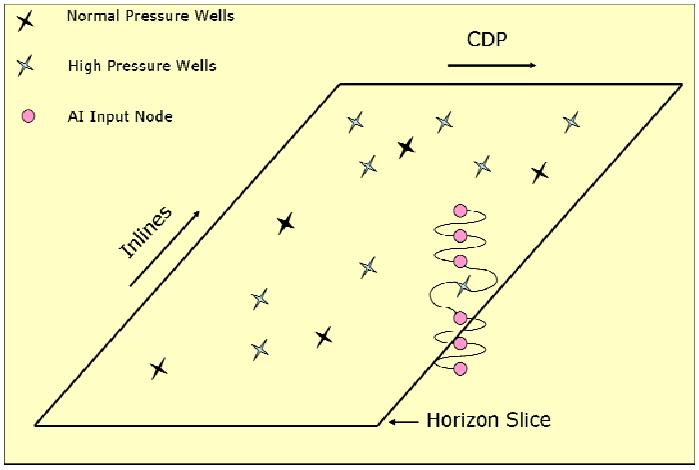


Figure 4.6. 2D trained operator used for geopressure prediction. The network is trained using values of geopressure from the wells. One data sample of geopressure is used from each well.

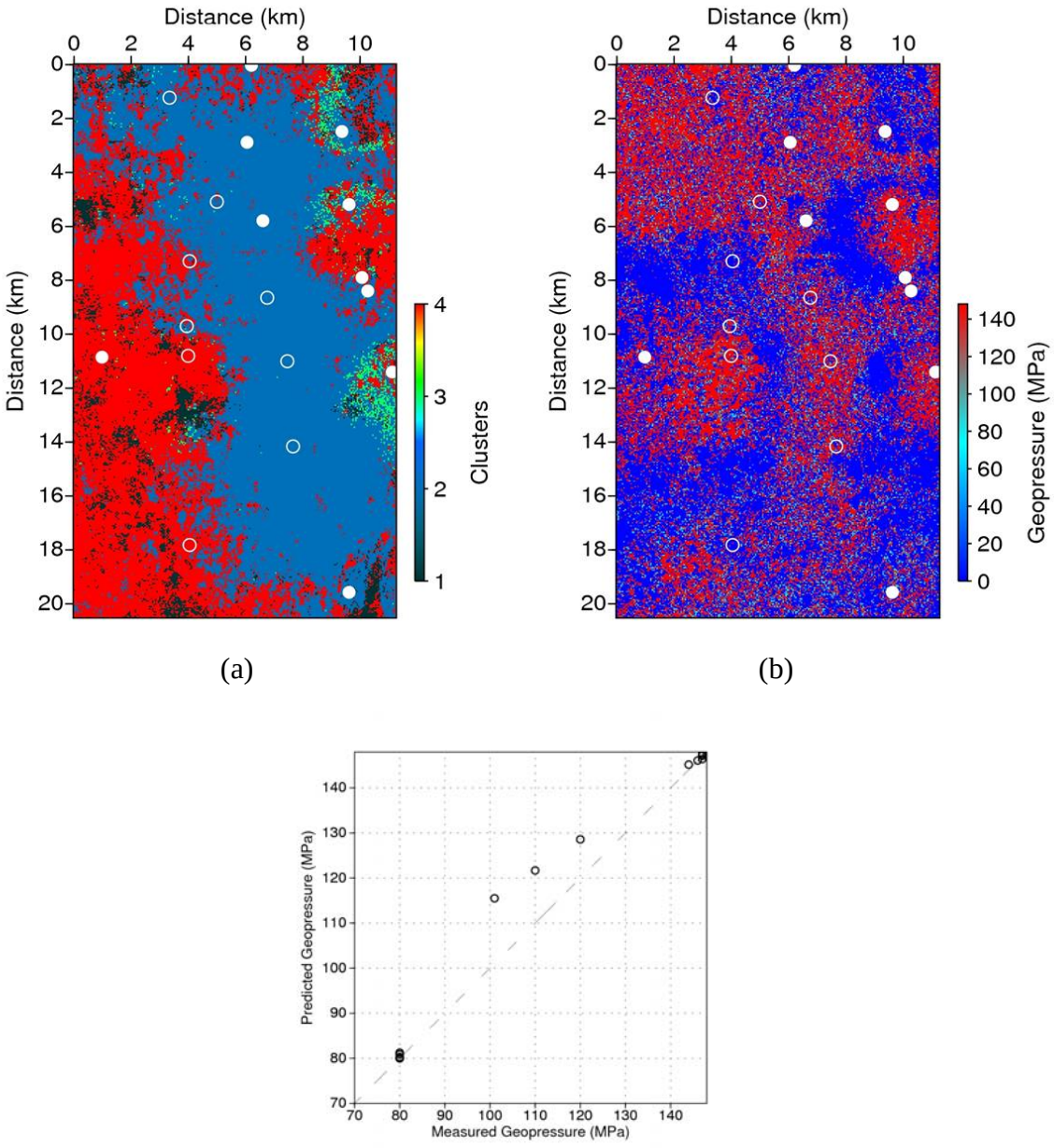


Figure 4.7 Geopressure prediction based on seismic impedance. (a) The result of unsupervised neural network SOM. (b) The result of supervised neural network multilayer perceptions. Solid dots indicate a well location with an over-pressure and the hollow dot indicates a well with normal pressure. (c) cross-plot of predicted pressure using supervised neural network MLP vs. pressure data (mud weights) at 17 wells

The second application of the 2D trained operator was used to predict porosity using the existing well logs information. The network was trained using 7 samples from each of the input 19 wells and a porosity value at the target horizon. Figure 4.8 depicts the porosity 2D trained operator. Figure 4.9 shows the final porosity result after the application. Also, for the sake of comparison, Figure 4.9a is the same as 4.7b to show the relation between porosity and geopressure.

The 2D trained operator performed the task of learning the target pressure values but the target values were not enough. The ANN algorithm needs more data to obtain better results. The same idea applies for the porosity prediction. Using a mud weight log or porosity log during training should teach the ANN to have better prediction than using only few input values.

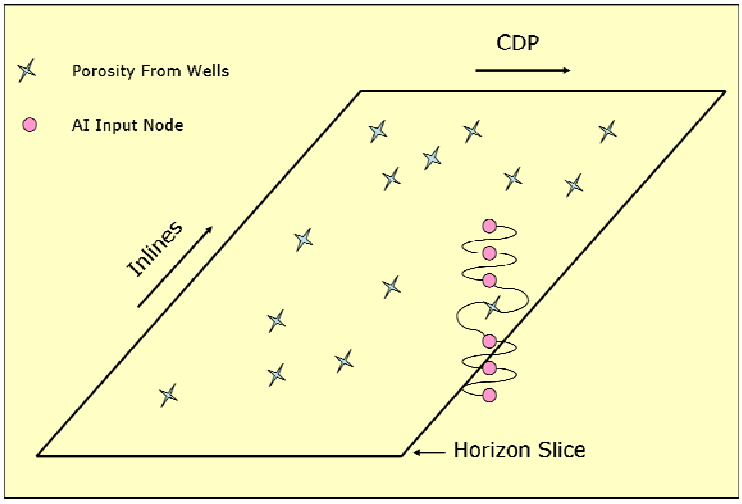
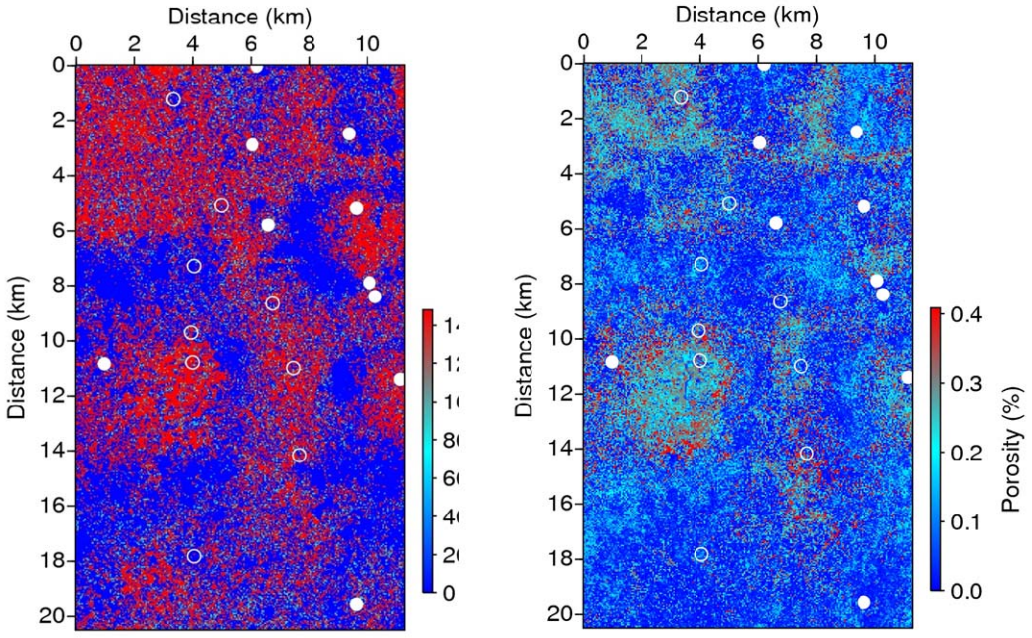


Figure **4.8.** 2D trained operator used for porosity prediction. The network is trained using values of porosity from porosity logs.



(a) (b)

Figure 4.9. (a) Geopressure prediction using a supervised neural network. (b) Porosity prediction using a supervised neural network.

4.5 3D operation (and comparison with OpendTect)

Applying the 2D trained operator in all of the volume or all samples of the input trace will upgrade the operator to a 3D trained operator. While the 2D trained operator was simple and quick, the 3D trained operator is robust and produces better results. Figure 4.10 shows the 3D trained operator. One can think about this operator as producing multiple horizon slices as the final outputs. The input to the 3D trained operator neural network is a full trace or part of the trace covering the formation of interest. The network is trained by using 7 inputs from the representative trace at the well location, and it keeps moving these 7 inputs down to cover the target formation. Well logs like porosity will be used for desired outputs. Figure 4.10 shows an example of 74 targets.

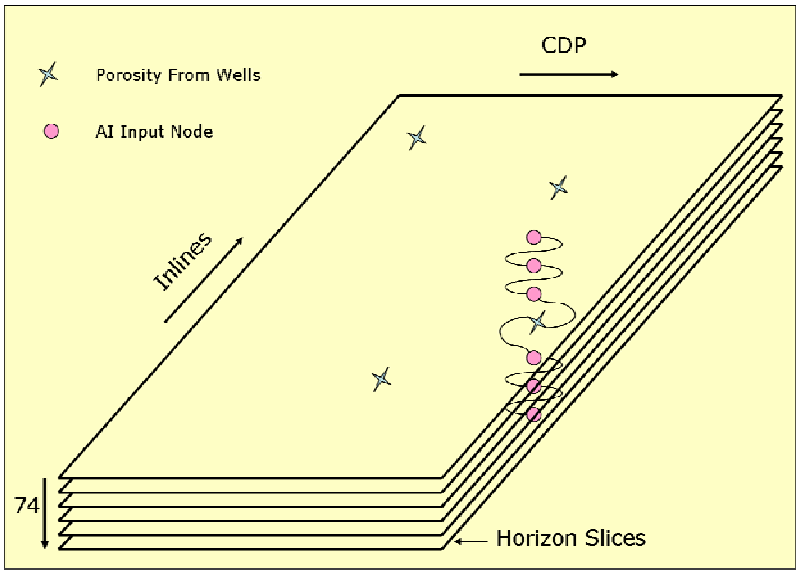


Figure 4.10. 3D trained operator used with multilayer perceptrons. The 2D slice is now moving to cover 74 horizon slices.

In order to make sure that my code is working well, I used the same data to run the multilayer perceptron in my code and a commercial software. OpendTect (the same software used in chapter 2) was used for the testing. OpendTect uses 3D seismic data that were acquired to explore for oil and gas in the Upper-Jurassic-Lower Cretaceous strata (Aminzadeh & Groot, 2006). The seismic data were inverted to acoustic impedance using the industry standard STRATA software. There are 4 available well data with porosity logs. Figure 4.11 shows the supervised neural network workflow that is used by OpendTect software.

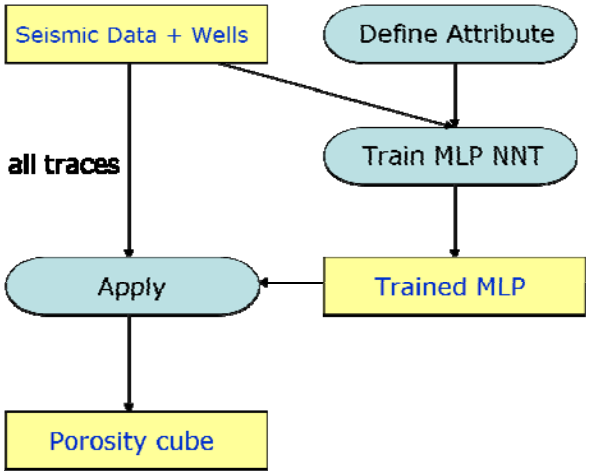
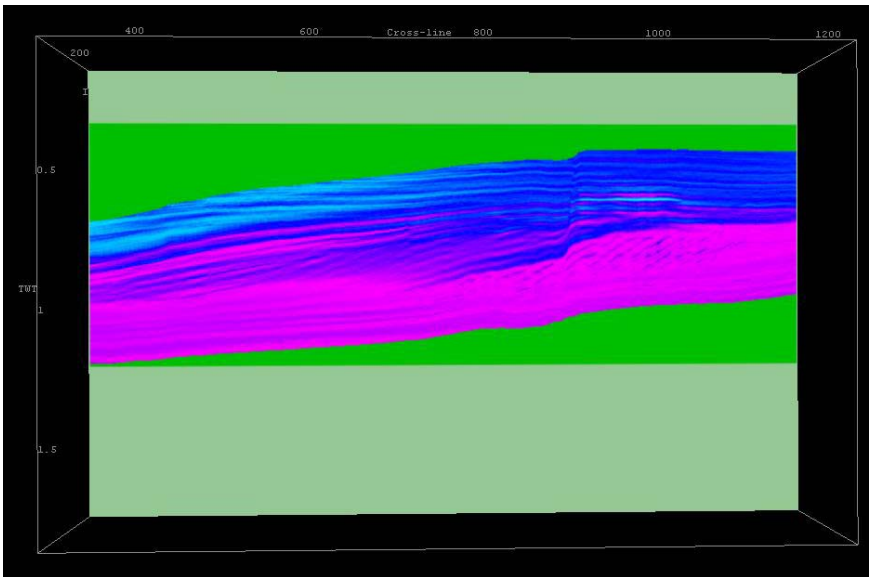
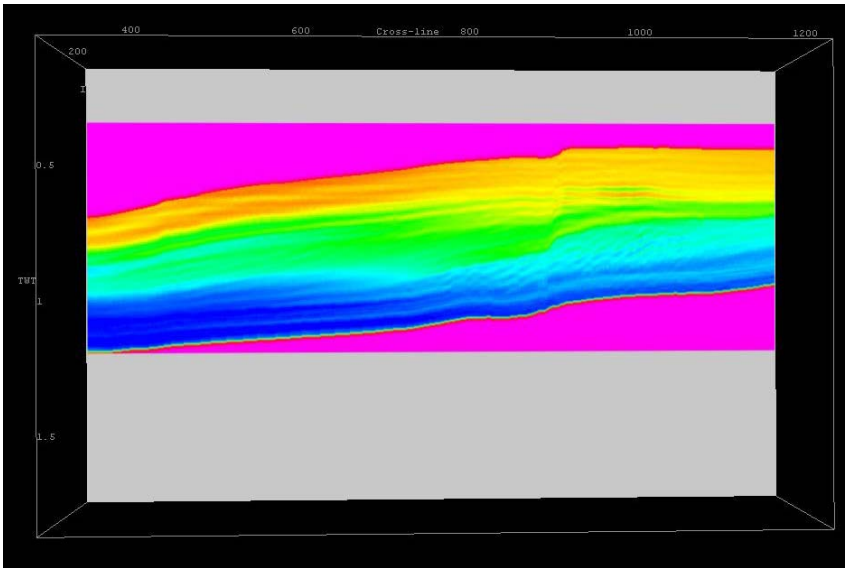


Figure 4.11. Workflow of supervised neural network used by OpendTect software. (modified after (Aminzadeh and Groot, 2006)

The workflow shows that the network is trained using the input seismic data and wells. Then the trained multilayer perceptron is used in the application stage. Figure 4.12a shows the acoustic impedance data that were used as an input. The target porosity values were taken from 3 wells in the area. Figure 4.12b shows the final OpendTect porosity prediction after applying the trained network to the acoustic impedance data.



(a)



(b)

Figure 4.12. (a) Input acoustic impedance used by OpendTect. (b) Predicted porosity of OpendTect multilayer neural network.

The acoustic impedance data were exported and prepared for use by my multilayer perceptron code. In my code, I tried to have the same parameters that OpendTect was using. The parameters have 7 samples in each input vector, one porosity desired output, 3 wells, 3 layers for the neural network, and 4 nodes or neurons used for the hidden layer. The other parameters, such as the learning parameter, were adjusted after trying many different values to produce the right values. These parameters were not shown by the OpendTect software. Figure 4.13 shows the error versus one of the learning parameters (Alpha). This analysis suggests that we set alpha to 0.5.

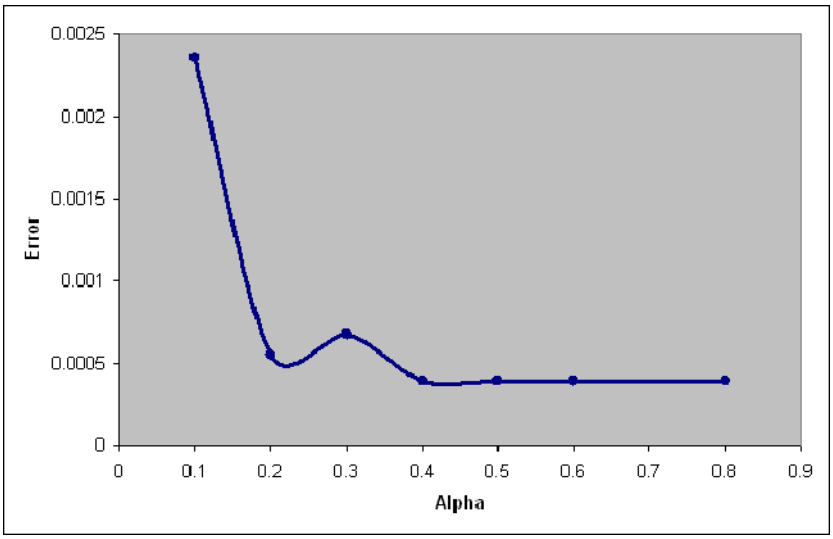


Figure 4.13 Error versus learning parameter alpha. During coding, the learning rate is changed many times to come up with the best and stable learning rate that produce the lowest network error.

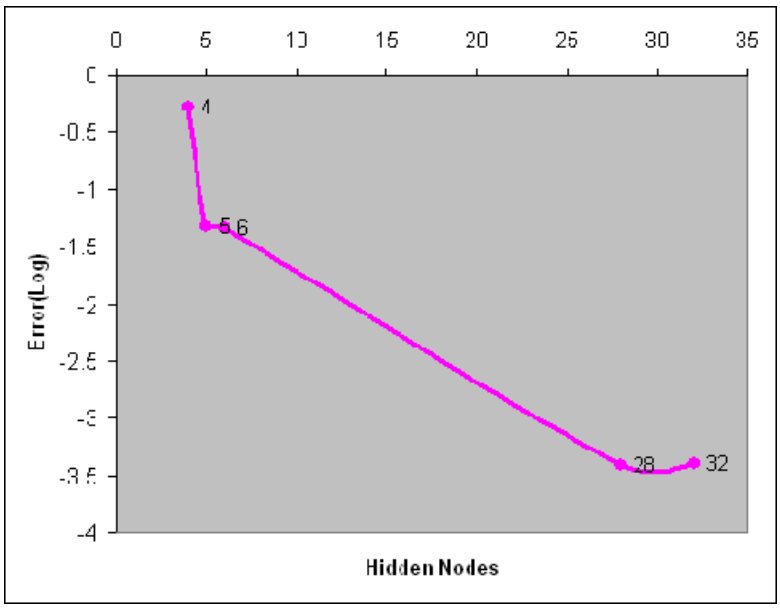


Figure 4.14. Error versus number of nodes in the hidden layer. As the number of hidden nodes increase, the error decreases. This is not always good because using too many hidden nodes will misguide the network and over-fitting will occur.

Figure 4.14 shows the error of the multilayer perceptron versus the number of hidden nodes in the hidden layer. The graph indicates that, whenever we increase the number of nodes in the layer, we obtain lower network errors. This is true but with two problems. The first problem is that for the sake of comparison, I should use the same number that OpendTect used with these data. The second problem is that the experiment showed a very good fit of curves when I used 28 nodes in the hidden layer, but this did not give me good final porosity results. The reason for getting bad final porosity results is that the network memorized just the input vectors that were used in the training stage. This is what I mentioned above when I talked about generalization. Figure 4.15 shows a comparison of the final porosity prediction. The figure shows just a trace or curve extracted from the same position in one of the wells that were used at the training stage. The figure shows the original porosity log (black), the porosity predicted by OpendTect system (blue), and the porosity predicted by my multilayer perceptron code (red). Figure 4.15b shows the case when I used 28 nodes in the hidden layer. The network memorized the original porosity log, and produced just the exact curve in the application stage. This is called overfitting or overtraining. Figure 4.15c shows the right estimation of the porosity after using 4 nodes in the hidden layer.

Figure 4.16a shows the input acoustic impedance cross section. Figure 4.16b shows the porosity prediction of OpendTect software. Figure 4.17a shows the prediction after applying my code and using 28 nodes in the hidden layer. Note that the porosity prediction is wrong because of the overfitting and memorizing of the input vectors only. Figure 4.17b shows the final prediction of my neural network method after fixing the overfitting problem. Note that my results are similar to the results predicted by the commercial software OpendTect. This means that the code is giving good results. Figure 4.18 shows a seismic amplitude line in the background and my predicted porosity section in the foreground.

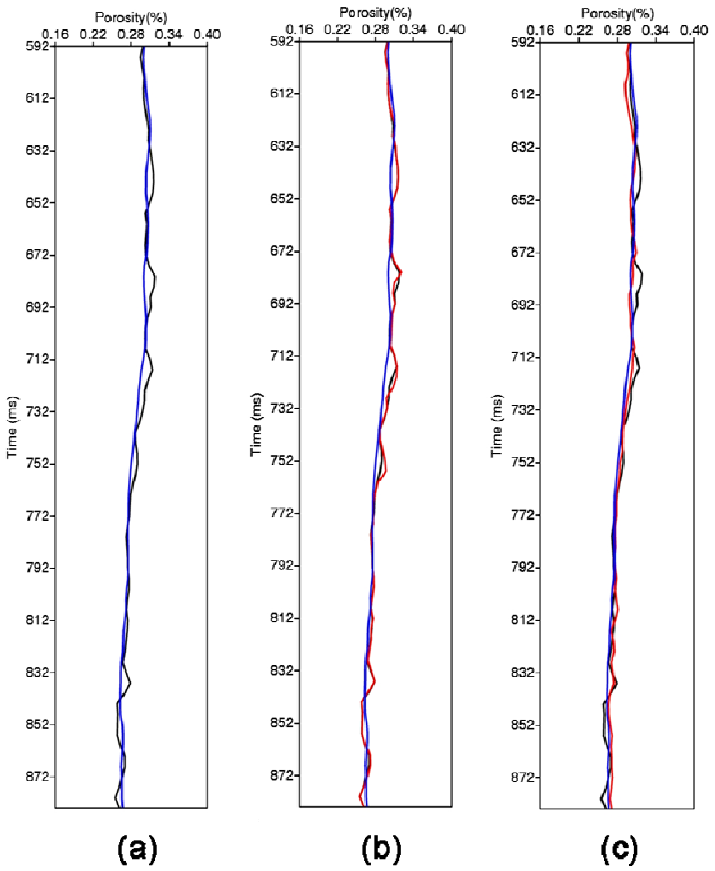
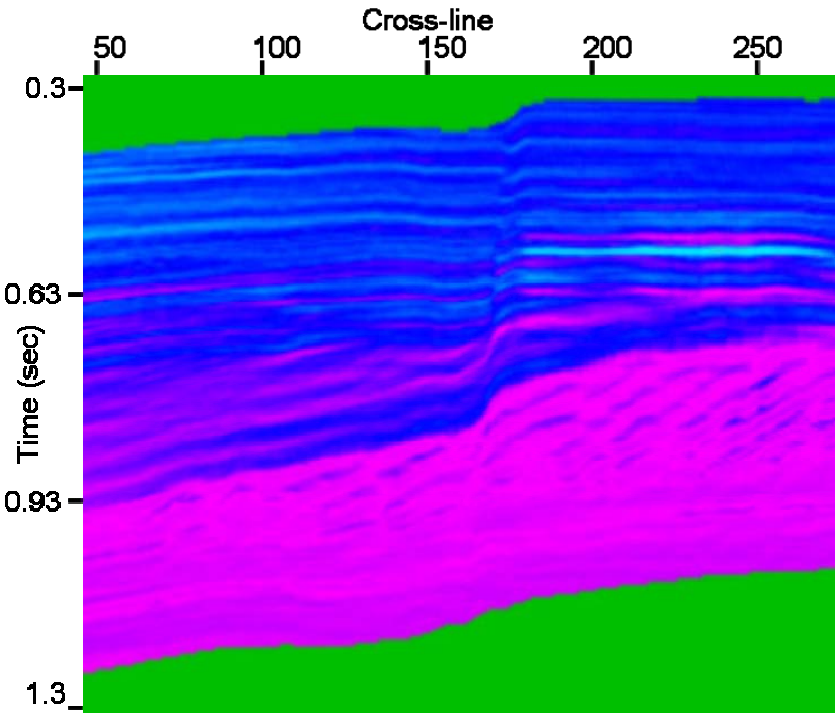
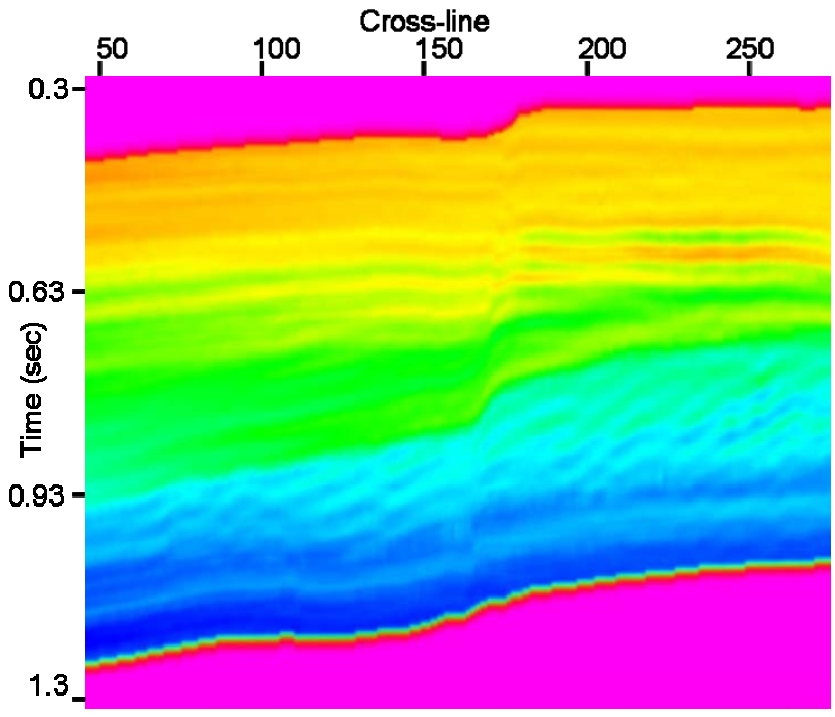


Figure 4.15. Comparison of original porosity log (black), porosity predicted by OpendTect (blue), and my porosity prediction (red). (b) Over-fitting or overtraining problem. (c) Final acceptable result.

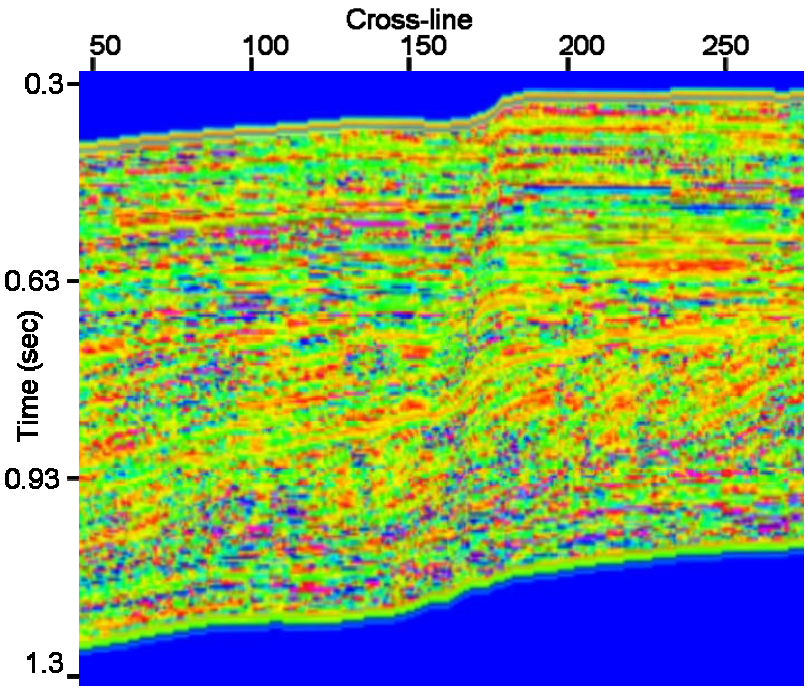


(a)

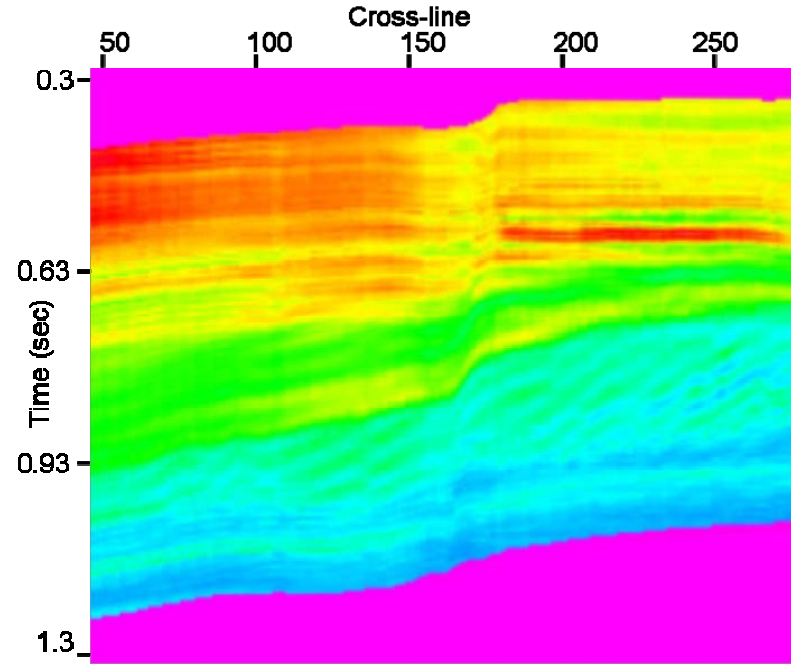


(b)

Figure 4.16. (a) Acoustic impedance input data. (b) Porosity prediction using OpendTect.



(a)



(b)

Figure 4.17. (a) My porosity prediction with over-fitting. (b) My porosity prediction of the final result.

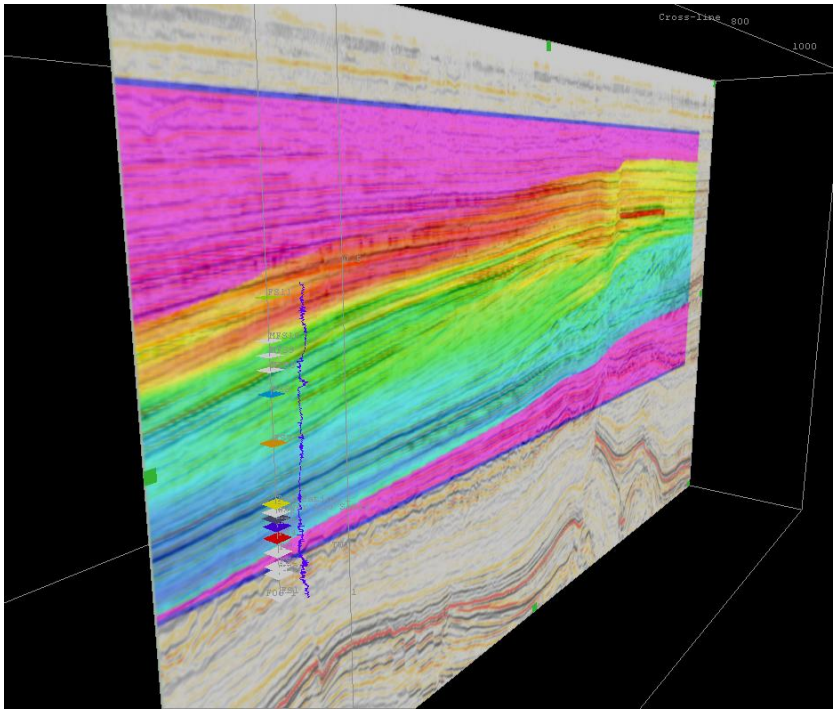


Figure 4.18. 3D view of seismic amplitude data in the background and the in-house code result of porosity in the foreground.

4.6 Application result to Saudi Aramco data

Figure 4.19 shows an inline extracted from the acoustic impedance data that were used as an input. The input to the 3D operation is a part of the trace covering the formation of interest. The network is trained by using 7 inputs from the representative trace at the well location, and it keeps moving these 7 inputs down to cover the target formation. The parameters used in my code have 7 samples in each input vector, one porosity desired output, 7 wells, 3 layers for the neural network, and 4 nodes or neurons used for the hidden layer. The other parameters, such as the learning parameter, were adjusted after trying many different values to produce the optimal values.

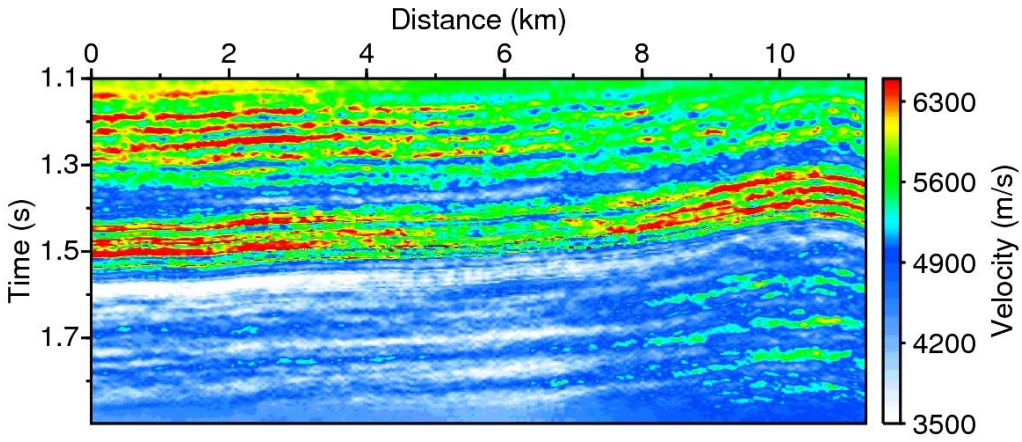


Figure 4.19 Seismic acoustic impedance profile. This 2D line is extracted from the 3D volume to show the impedance events and layers.

At the beginning of the training, I trained the network with 3 wells only. This gave me an error estimation of about 0.210163. Figure 4.20 shows a comparison between the original porosity logs (black) and the predicted logs (red) using 3 training wells. Unfortunately, the prediction results were not good enough when we tried to display 3 testing wells that had not been used during training. Figure 4.21 shows 3 testing wells with original porosity (black) and predicted porosity (red). Note that well number (a) has some problems in the shallow parts. Also, the predicted porosity is not smooth. I then used 7 wells in my training, and that gave me an error estimation of about 0.660944. The error increased here (compared with 3 wells) because the number of wells increased. Figure 4.22 shows the new porosity comparison. Figure 4.23 shows the 3 testing wells which show smoother prediction and better estimation. So, using 7 training wells gave me better porosity estimation than using 3 wells only.

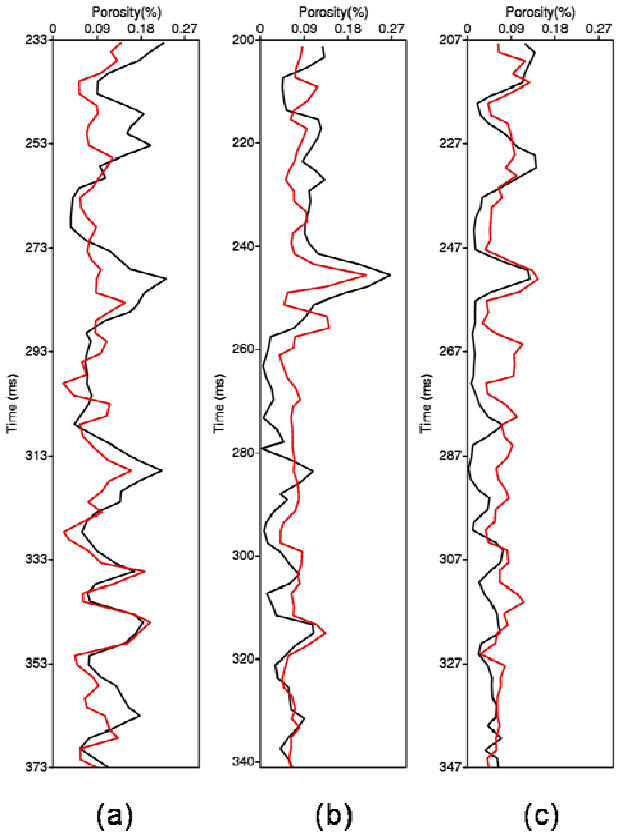


Figure 4.20 The original porosity (black curves) and predicted porosity (red curves) after using 3 wells (a, b, and c) as training wells.

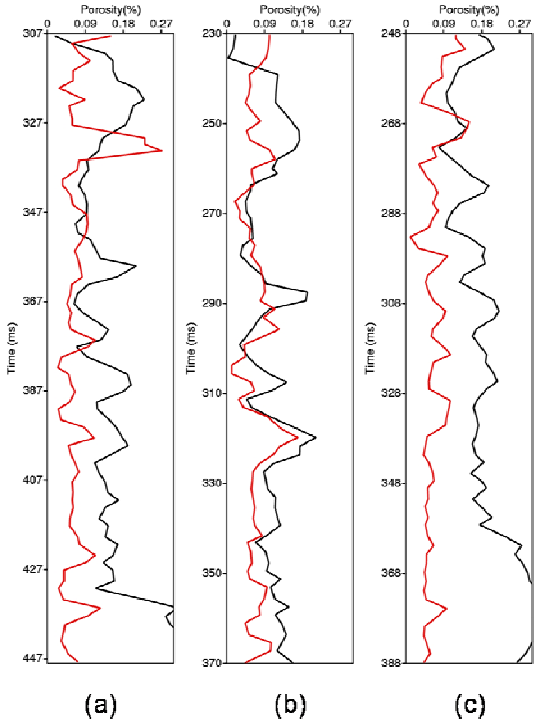


Figure 4.21 3 wells to test the porosity prediction after using 3 training wells. The original porosity (black curves) and predicted porosity (red curves).Note the misprediction of shallow values of well (a).

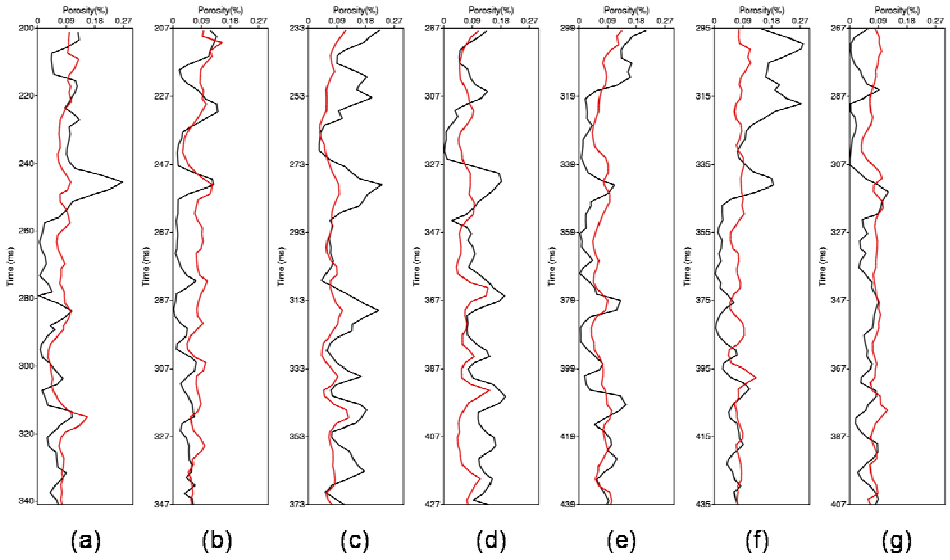


Figure 4.22 The original porosity (black curves) and predicted porosity (red curves) after using 7 wells (a to g) as training wells.

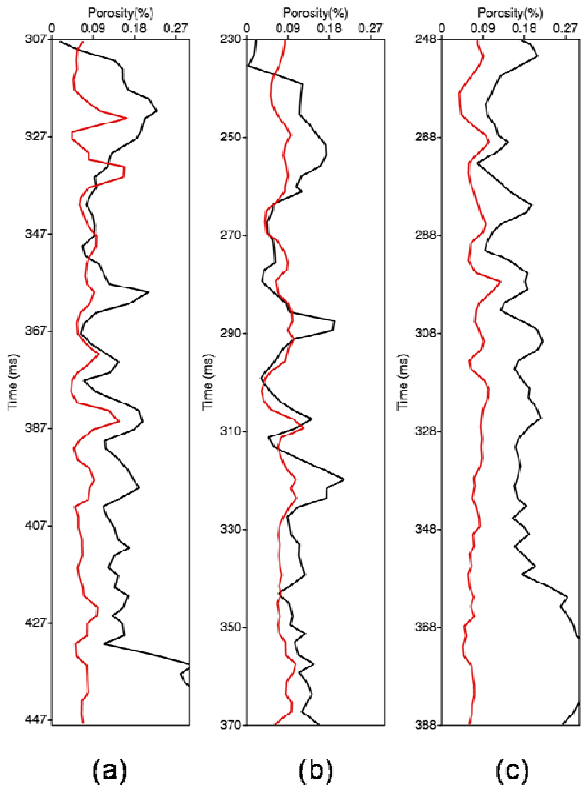
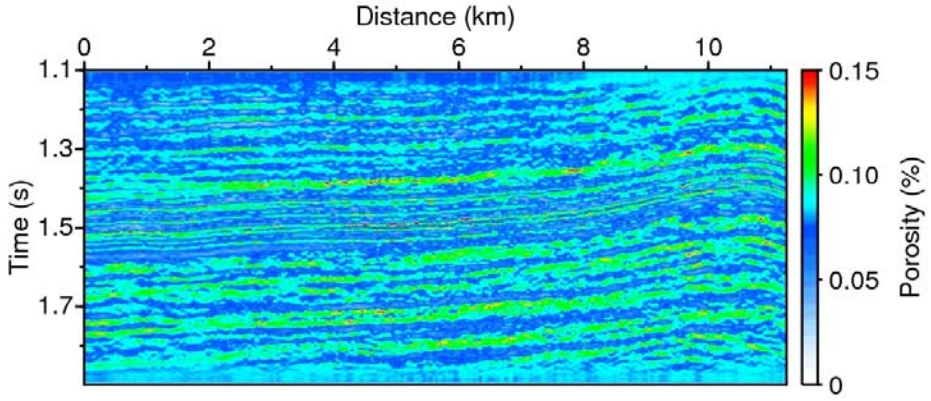
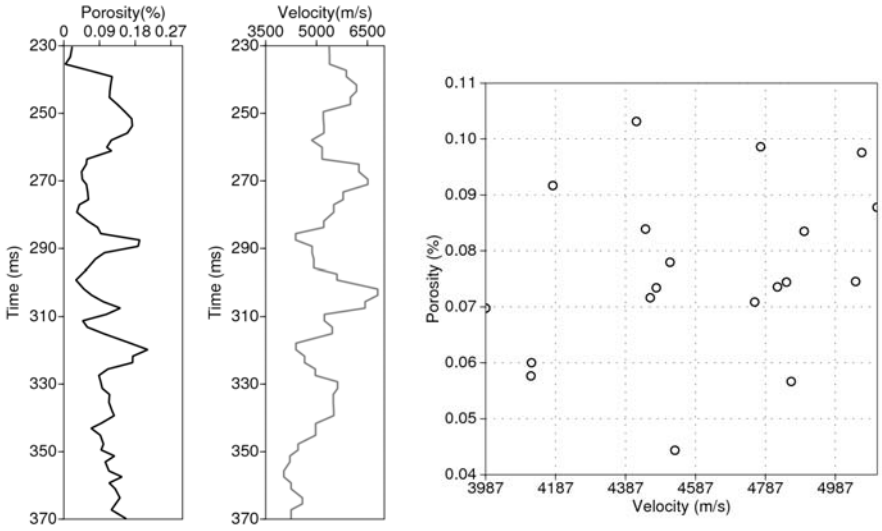


Figure 4.23 3 wells to test the porosity prediction after using 7 training wells. The original porosity (black curves) and predicted porosity (red curves).

I apply now the 3D slice trained operator successfully to the volume along time samples. It produced multiple horizon slices as the final output. After applying the trained multilayer perceptron to the seismic data (impedance), I obtain a porosity volume that estimates porosity in the area. Figure 4.24a is a 2D line extracted from the new predicted porosity volume. The porosity is shown from 0.0% to 0.15%, and the data coverage is from 1.1 sec to 1.9 sec. Note from the colour bar that red zones have more porosity than others. Figure 4.24b shows a plot of impedance trace and porosity trace at one well location. Figure 4.24c is a crossplot showing the nonlinear relationship between the seismic impedance velocities and the porosity values using 19 wells. Figure 4.25 is a 3D view of the area, with one inline showing the original impedance data (right) and one cross-line porosity in the foreground and seismic amplitude in the background (left).



(a)



(b) (c)

Figure 4.24 (a) Predicted porosity profile. This is the porosity output when I used the seismic impedance 2D line from figure 4.19. (b) A plot of impedance trace and porosity trace at one well location. (c) A crossplot showing the random relationship between the porosity and seismic impedance using 19 wells.

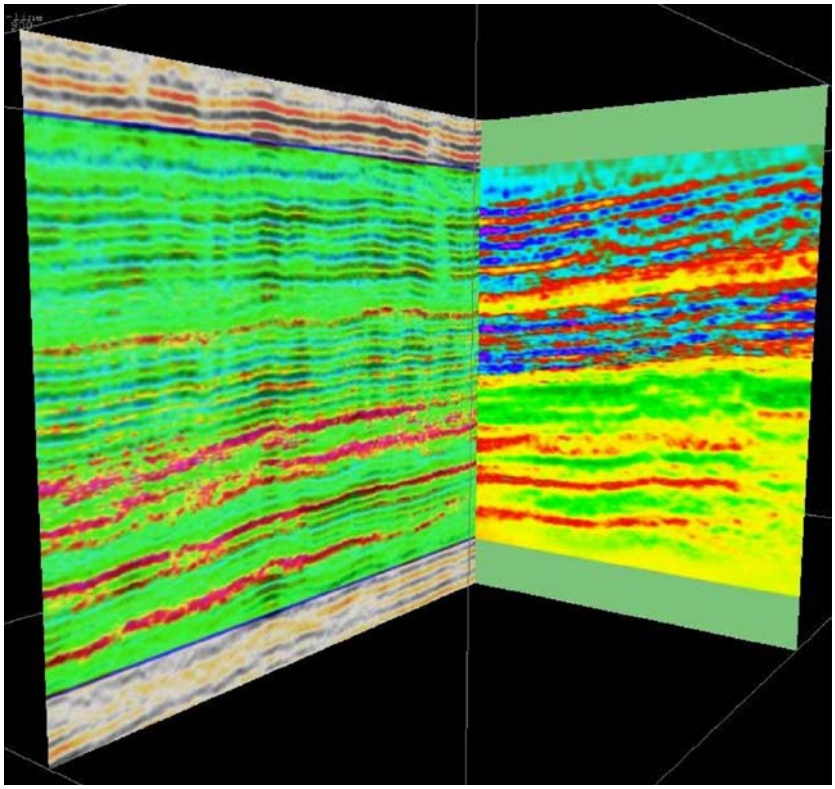


Figure 4.25 A 3D view showing one impedance cross-line (right) and one inline that has both porosity and seismic amplitude (left).

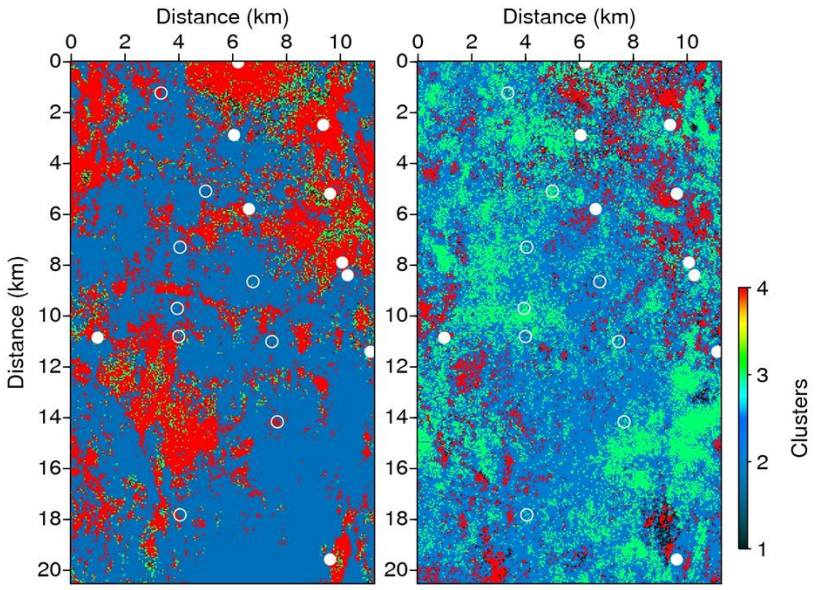
4.7 Relationship between geopressure and porosity

The next step is to relate the predicted porosity to pore pressure. One way for direct prediction or estimation of the pore pressure is to use the unsupervised neural network method (SOM) to cluster the seismic impedance data. The geopressure result I obtained from the previous chapter (Figure 4.7a) gives a good indication of high pressure in the areas that have wells with high pressure indication. It was a horizon slice that passes through the top of our zone of interest. Solid circles in the figure are wells with high pressure indication in the zone of interest, while hollow circles are wells with normal pressure. The red colour area shows more high pressure wells.

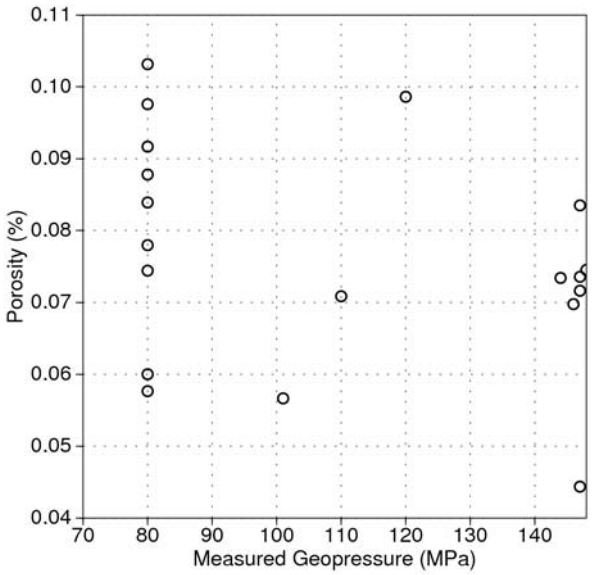
Now I can also cluster the porosity volume using the SOM method to find a structure or segment in the porosity data, and thereby extract the relevance between pore pressure

and porosity. Figure 4.26a is the clustering result of predicted porosity volume. Here I attempt to find zones with high porosity, which could relate more to high pressure because of hydrocarbon accumulation in these target zones. I have found that the clustered porosity volume can be combined with the clustered impedance volume to predict high pressure zones more accurately. I could say from this result that the clustered porosity (red in Figure 4.26a) selected specific locations from the clustered impedance (red in Figure 4.7a).

Furthermore, I upgrade the operator used in the SOM method to allow it to do unsupervised vector quantization using multiple attributes. This time the SOM network was fed with two attributes, namely impedance and porosity. Since the two attributes have different ranges in their values, the input in this case must be normalized such that the two attributes have fairly similar ranges. Otherwise, the impedance attribute will dominate the analysis when distance norms are used, because it has higher absolute values and thus the norms would be disproportionally affected by the impedance relative to the porosity. The normalization was done by adjusting all the input values (impedance and porosity) to have a mean of zero and a variance of one. Figure 4.26b shows the result of the SOM after using multiple attributes input. I observe that roughly 8 out of 10 (80%) overpressured wells are shown in one cluster, and 7 out of 9 (78%) normal pressured wells are shown in another cluster number. Figure 4.26c is a crossplot showing the nonlinear relationship between the pore pressure and the porosity values using 19 wells. For the pressure data around 80 MPa, these circles represent the normal pressure wells.



(*a*) (*b*)



(c)

Figure 4.26. (*a*) The pressure prediction after applying the SOM to the predicted porosity. (*b*) The prediction picture after applying the SOM to the attributes, including the impedance and the porosity estimate. (c) A crossplot showing the random relationship between the porosity and pore pressure values using 19 wells.

4.8. Summary

A multilayer perceptrons neural network method is used for finding a non-linear relation between seismic attributes (impedance) and pore pressure or porosity. The operation is slice based, and the 2D slice result shows some consistency with the results of the unsupervised method. The operation is performed to 3D volume successfully slice by slice, producing a robust generalization. The supervised neural network method was applied to real 2D data as well as 3D data. The method was also applied to seismic carbonate data for the sake of reservoir characterization. We use neural network methods for finding a non-linear relation between seismic impedance and pore pressure/porosity and the relation between seismic impedance/porosity and pore pressure. Comparing the four pore pressure results (Figures 4.7a, 4.7b, 4.26a and 4.26b), we can have the follow observations:

(1) Unsupervised neural network method SOM has done a very good job in clustering seismic impedance (Figure 4.7a) and the porosity (Figure 4.26a). The results have shown some consistency with high and normal pressure wells.

(2) Because of lack of data, the supervised method multilayer perceptrons has not done a good job in finding the areas with high and low pressure zones (Figure 4.7b). Multilayer perceptrons neural network has been trained by only one sample from each well. But it can estimate porosity robustly as we do have porosity information in well locations.

(3) The porosity data have shown segments related to overpressured when clustered using the SOM method (Figure 4.26a). In addition, using both seismic impedance and porosity attributes, SOM has given us an acceptable indication of high and low pressure areas on the final maps (Figure 4.26b).

Chapter 5

The group method of data handling (GMDH)

The relationship between the seismic data and the reservoir properties can be modelled using statistical approaches. Regression is simply the statistical fitting of trend lines to a data set. Linear regression is an analysis in which the relationship between one or more independent variables and a dependent variable is modelled. Linear regression and multi-linear stepwise regression were used for optimal attribute selection (Hampson *et al.,* 2001). Nonlinear regression techniques such as multilayer ANN were also used to predict reservoir properties (Saggaf *et al.,* 2003). The group method of data handling (GMDH) is another nonlinear predictor which has been proven to perform better than ANN.

GMDH is not a regular feed-forward multilayer networks (presented in the previous chapters), in which the number of layers and links between layers are known before training. During training, the GMDH network uses a genetic component to decide how many layers are used in the final network structure (Farlow, 1984b). The predicted output at the output layer of the GMDH network can be represented as a polynomial function of some or all of the inputs. GMDH prediction is significantly robust and more accurate than those of the ANN (Campbell and Johnson, 1993; Montgomery and Drake, 1991; Shastri *et al.,* 1998).

One important characteristic of GMDH is its ability to automatically synthesize the network from a database of inputs and outputs representing a training set of solved examples. This automation can be called the self-organization of input models (Ivakhnenko, 1971). Thus the user is not required to specify the network architecture in advance, which protects the final generated model from human bias and misjudgements.

Also, this reduces the need for lengthy analysis in the model synthesizing process and reduces the computing time. GMDH can be classified as a special class of biologically inspired networks with artificial intelligence (Barron *et al.,* 1984). GMDH is related to human behaviours by dealing with a limited number of inputs at a time, and then summarizing the input information and passing the summarized information to a higher level (Lee *et al.,* 1999).

GMDH has been used in the classification of difficult pattern recognition problems, in determining which inputs are more important to the modelled system, and in predicting the outputs of complex systems. In the industry, GMDH was used in predicting tool life in drilling, predicting temperature distributions, predicting an acceptable product on T-shape tube hydro-forming process, radar applications, missile guidance, multisensory signal processing, biomedical modelling, econometric forecasting, environmental systems, and many other uses (Abdel-Aal, 2003; Barron *et al.,* 1984; Farlow, 1984b; Fulcher and Brown, 1994; Lee *et al.,* 1999; Lin and Kwan, 2004). From my reading while doing the literature review, I never came across prediction of geophysical reservoir properties from seismic networks using GMDH.

In this chapter, I start by introducing the GMDH theory and equations. Then, I discuss the learning and training of the GMDH algorithm. In the application stage, I show a synthetic and real data examples. I used the following seismic attributes as input to the GMDH: seismic amplitude, seismic impedance, porosity, instantaneous amplitude, instantaneous phase, instantaneous frequency, instantaneous bandwidth, first and second envelope trace derivative, and average frequency. I also make a result comparison of the GMDH and supervised ANN.

5.1 Group method of data handling

The GMDH builds a polynomial function which behaves in such a way that the predicted output will be as close as possible to the actual output. With the use of the polynomial functions, GMDH is capable of representing complex and uncertain relationships between input and output variables. Original inputs are used to generate a network of layers and nodes. Each layer feeds its outputs to the next layer. In this section, I study the basic aspects of the GMDH.

The topology of the GMDH network is determined using a layer-by-layer pruning process based on a predefined criterion of what are the best nodes at each layer. Figure 5.1 shows the basic scheme of the GMDH topology. Ivakhnenko (1976) started the algorithm by keeping in mind that many types of mathematical models require the modeller to know some system variables that are generally very difficult to find. The modeller will be forced to guess these variables, but this method is time consuming and might produce unreliable prediction models. GMDH uses an iterative polynomial regression procedure in order to synthesize any model. The polynomial regression equations are able to produce a high order polynomial model using the effective predictors. Ivakhnenko started by computing the quadratic polynomial regression equation:

$$y = a + bx_i + cx_j + dx_i^2 + ex_j^2 + fx_i x_j , \tag{5.1}$$

where y is the output sample, (x_i, x_j) are pair of input samples, and a, b, c, d, e, f are the polynomial coefficients to be determined by the training data set. For each pair of input samples (x_i, x_j), a quadratic polynomial equation is used to produce an output sample y. Note that each input and output contains of N elements, and the coefficients are shared by all of these N elements. Figure 5.2 illustrates the basic propagation scheme of a GMDH algorithm. The first layer is the input data with m number of attributes or variables. The second generation (or second layer) has m(m-1)/2 higher-order variables produced from the original x_m input variables. After solving the quadratic polynomial equations, we keep the equations that best describe the target dependent variable (or real output), and discard all other equations. This is done by using an objective function like the root mean squared error to determine which polynomial equations are applicable to go to the next layer. This will be described more in the next section.

The remaining set of quadratic polynomial equations (each produced by using a pair of input variables) now are used to produce new variables (x_{m1}) which will replace the original input variables x_m. This is done by using each pair of data from the original input variables and the polynomial coefficients produced by that pair to generate new variables. Equation (5.1) will be used again with the new variables x_{m1} to produce the third generation which is $m_1(m_1-1)/2$. The procedure will continue generating new independent

variables and replacing the old ones, using only one pair of input variables with equation (5.1) until the polynomial equations start to give poor predictions compared with previous generations or layers. This could happen when the network starts to memorize the training variables or when overtraining happen. After stopping, the best quadratic polynomial equations that were picked from all generations will be used to give the final dependent variable *y*. By doing the required algebraic substitutions using all these best quadratic polynomial equations, a complicated polynomial equation will be formed. Ivakhnenko found that the data can be modelled by using an approximation of the Kolmogorov-Gabor polynomial:

$$\mathbf{y} = a + \sum_{i=1}^{m} b_i \mathbf{x}_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij}\mathbf{x}_i\mathbf{x}_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} d_{ijk}\mathbf{x}_i\mathbf{x}_j\mathbf{x}_k + \cdots , \qquad (5.2)$$

where $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ are the inputs, $\mathbf{y}$ is the final output, m is the number of variables in the layer, and a, b_i, c_{ij}, d_{ijk} are the coefficients of the polynomial functional nodes. Expression (5.2) is called the 'Ivakhnenko polynomial' (Farlow, 1984a).

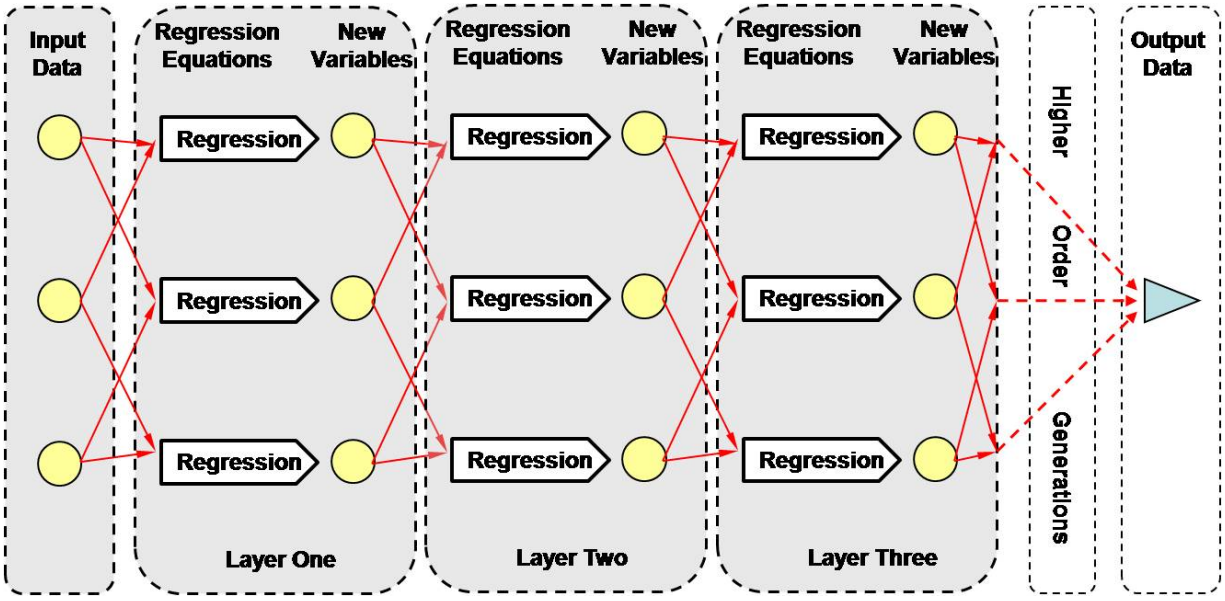


Figure 5.1. Basic scheme of the GMDH topology. Each pair of input data forms a regression equation. The outputs of the regression equations form new inputs to the next layer. The final output consists of selected regression equations from all layers.

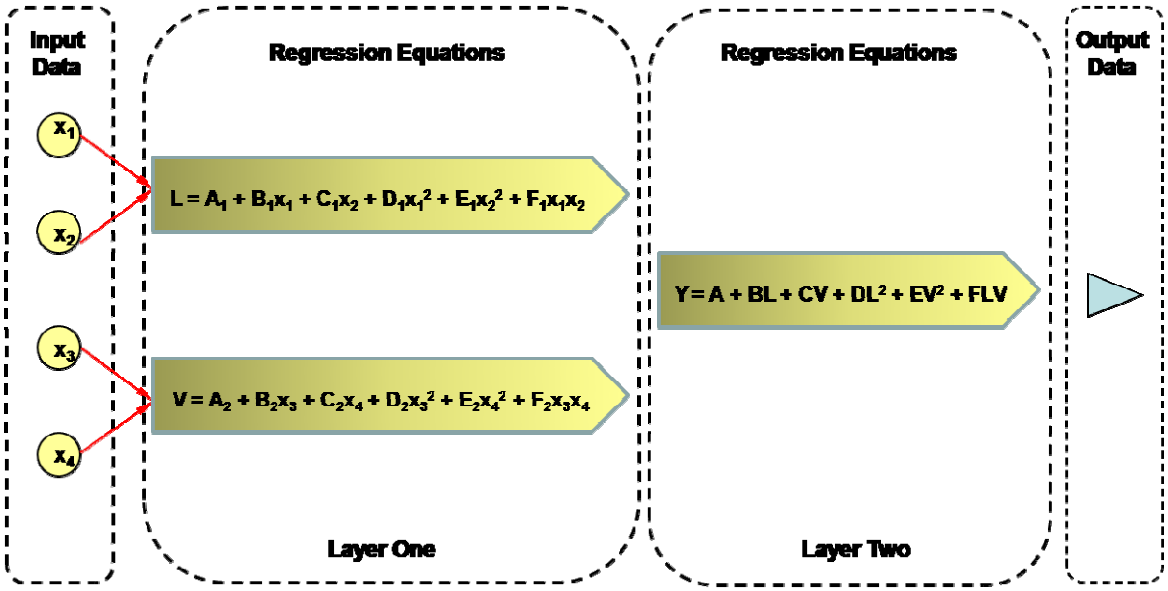


Figure 5.2. Basic propagation scheme of GMDH algorithm. Layer one is showing the relation between two input pares as regression equations. Layer two is showing the relation between the variables from layer one.

5.2 Learning and training

Training of the GMDH is done iteratively in order to synthesize the final model. The learning process teaches the GMDH network how to deal with new data that have never been presented to the network. The network is trained on a representative dataset. Each dataset consists of *n* number of observations and each observation consists of *m* number of inputs. A classical GMDH divides the number of observations (*n*) into two datasets. One dataset (n_t) will be used for training and the other one (n_c) for testing or checking the regression equations. The reason behind dividing the data into two datasets is to prevent overtraining. The training dataset will be used to calculate the polynomial coefficients of the quadratic polynomial equations, and the checking dataset will be used to evaluate the quality of these polynomial coefficients (Ivakhnenko, 1971). GMDH will use an input matrix of *n* observation (rows) and *m+1* inputs (columns). These columns are the independent variables (x_1, x_2, x_3, ..., x_m) and one dependent variable *y*.

The training iterations start by taking all the independent variables (two at a time) and then constructing the quadratic regression polynomial (equation 5.1) that best fits the dependent variable y_i. These iterations will use only the training dataset n_t. The first layer is now constructed using the input variables (n_t) together with the dependent variables to get m_1 ($m_1 = m(m-1)/2$) regression polynomials. For example, if the independent input variables were $m = 4$ in each observation row, then $m_1 = 6$ regression polynomial equations will be generated. New input variables ($z_1, z_2, z_3, ..., z_{m1}$) are now constructed in layer one using the resulting $m(m-1)/2$ polynomial functions and data from the pair of x variables used to generate it. The new generated variables are supposed to describe the dependent variable y better than the original input variables. Since the number of variables increases each time, we do the iterations using ($m(m-1)/2$); one needs to keep the variables that best describe the dependent variable y and discard all others. GMDH does this job by screening all the values of ($z_1, z_2, z_3, ..., z_{m1}$) and eliminating the least effective ones. This could be done by using the regularity criterion or the root mean squared error (r_s) in equation (5.3) over the checking dataset n_c. Equation (5.3) takes the sum of the squared differences between the real output and the prediction divided by the sum of the squared real outputs. This should be done using the checking dataset.

$$r_s^2 = \sum_{l=1}^{n_c} \left(y_l - z_{sl}\right)^2 / \sum_{l=1}^{n_c} y_l^2 , \qquad s = 1, 2, 3, ..., m_1, \qquad (5.3)$$

After computing the values of r_s at each z polynomial equation and using the checking dataset, the matrix columns of z will be arranged according to increasing r_s. Only r_s and z_{m1} values that satisfy some prescribed number chosen in advance will be chosen to replace the original columns of x_m. The minimum r_s value (r_min) will be saved for the sake of comparing it with the minimum r_s values in the next layer or generation. Training and iterations will continue to produce new z_{mi} values and corresponding r_s values until the minimum value of r_s (r_min) is no more less than the previous minimum r_s value from the previous layer. Increasing minimum r_s means that the data model has become complex and overtraining is happening (Barron *et al.,* 1984). Figure 5.3 shows the stopping criterion of the classical GMDH algorithm.

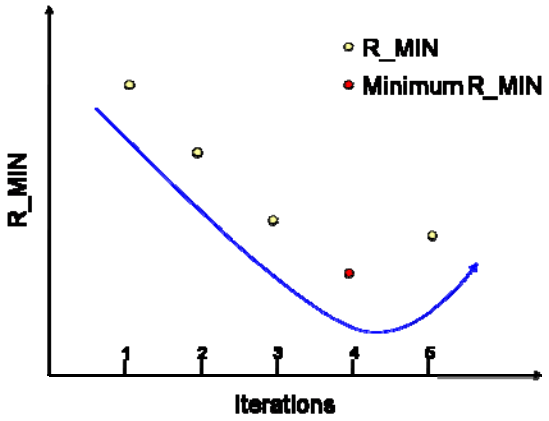


Figure 5.3. Stopping criterion of GMDH. (Edited after Farlow 1984a)

One important step in each training iteration is to compute the coefficients of the quadratic polynomial regression equation (5.1). The six polynomial coefficients are solved by minimizing the objective function,

$$\phi = \sum_{n} [y(n) - z(n)]^2 \quad , \tag{5.4}$$

where $y(n)$ is the *n*th samples of the predicted output sample, and *z*(*n*) is the *n*th sample of the desired output. To get the coefficients, get the partial derivatives of equation (5.4) with respect to each constant value (*a, b, c, d, e, and f*) and set equal to zero. The following system of equations is solved using the training data set:

$$\frac{\partial r}{\partial A} = \sum_{l=1}^{n} y_l = \sum_{l=1}^{n} A + Bx_i + Cx_j + Dx_i^2 + Ex_j^2 + Fx_i x_j = 0 \quad , \tag{5.5}$$

$$\frac{\partial r}{\partial B} = \sum_{l=1}^{n} y_l x_i = \sum_{l=1}^{n} Ax_i + Bx_i^{\,2} + Cx_j x_i + Dx_i^3 + Ex_j^2 x_i + Fx_i^{\,2} x_j = 0 \quad , \tag{5.6}$$

$$\frac{\partial r}{\partial C} = \sum_{l=1}^{n} y_l x_j = \sum_{l=1}^{n} Ax_j + Bx_i x_j + Cx_j^{\,2} + Dx_i^2 x_j + Ex_j^3 + Fx_i x_j^{\,2} = 0 \quad , \tag{5.7}$$

$$\frac{\partial r}{\partial D} = \sum_{l=1}^{n} y_l x_i^2 = \sum_{l=1}^{n} Ax_i^2 + Bx_i^{\,3} + Cx_j x_i^2 + Dx_i^4 + Ex_j^2 x_i^2 + Fx_i^{\,3} x_j = 0 \quad , \tag{5.8}$$

$$\frac{\partial r}{\partial E} = \sum_{l=1}^{n} y_l x_j^2 = \sum_{l=1}^{n} Ax_j^2 + Bx_i x_j^2 + Cx_j{}^3 + Dx_i^2 x_j^2 + Ex_j^4 + Fx_i x_j{}^3 = 0 \quad , \tag{5.9}$$

$$\frac{\partial r}{\partial F} = \sum_{l=1}^{n} y_l x_i x_j = \sum_{l=1}^{n} Ax_i x_j + Bx_i{}^2 x_j + Cx_i x_j{}^2 + Dx_i^2 x_j{}^2 + Ex_i x_j^3 + Fx_i{}^2 x_j{}^2 = 0 \quad , \tag{5.10}$$

which can be simplified as follows,

Assume that $\mathbf{Y} = (1 \quad x_i \quad x_j \quad x_i{}^2 \quad x_j{}^2 \quad x_i x_j)$ (5.11)

and assume that $\mathbf{a} = \mathbf{Y}^T \mathbf{Y}$ (5.12)

The matrix ***a*** can be written as

$$a = \begin{bmatrix} 1 & x_i & x_j & x_i^2 & x_j^2 & x_i x_j \\ x_i & x_i^2 & x_i x_j & x_i^3 & x_i x_j^2 & x_i{}^2 x_j \\ x_j & x_i x_j & x_j^2 & x_i^2 x_j & x_j^3 & x_i x_j{}^2 \\ x_i^2 & x_i^3 & x_i^2 x_j & x_i^4 & x_i^2 x_j^2 & x_i{}^3 x_j \\ x_j^2 & x_i x_j^2 & x_j^3 & x_i^2 x_j^2 & x_j^4 & x_i x_j{}^3 \\ x_i x_j & x_i{}^2 x_j & x_i x_j{}^2 & x_i{}^3 x_j & x_i x_j{}^3 & x_i^2 x_j^2 \end{bmatrix}, \tag{5.13}$$

Now assume that matrix **Z** holds the coefficients,

$\mathbf{Z} = (a \quad b \quad c \quad d \quad e \quad f)$. (5.14)

Now the right hand side of equations (5.5-10) is simplified into $\sum_{l=1}^{n} aZ$

While y is the real output, assume that:

$\mathbf{b} = (y\ \mathbf{Y})^T = (y \quad yx_i \quad yx_j \quad yx_i{}^2 \quad yx_j{}^2 \quad yx_i x_j)^T$ (5.15)

The left hand side of equations (5.5-10) is simplified into $\sum_{l=1}^{n} b$

Then the final simplification is

$$\sum_{l=1}^{n} aZ = \sum_{l=1}^{n} b \quad , \tag{5.16}$$

This leads to the following matrix-vector system:

$$\sum_n \begin{bmatrix} 1 & x_i & x_j & x_i^2 & x_j^2 & x_i x_j \\ x_i & x_i^2 & x_i x_j & x_i^3 & x_i x_j^2 & x_i^2 x_j \\ x_j & x_i x_j & x_j^2 & x_i^2 x_j & x_j^3 & x_i x_j^2 \\ x_i^2 & x_i^3 & x_i^2 x_j & x_i^4 & x_i^2 x_j^2 & x_i^3 x_j \\ x_j^2 & x_i x_j^2 & x_j^3 & x_i^2 x_j^2 & x_i^4 & x_i x_j^3 \\ x_i x_j & x_i^2 x_j & x_i x_j^2 & x_i^3 x_j & x_i x_j^3 & x_i^2 x_j^2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \\ f \end{bmatrix} = \sum_n \begin{bmatrix} y \\ y x_i \\ y x_j \\ y x_i^2 \\ y x_j^2 \\ y x_i x_j \end{bmatrix}, \qquad (5.17)$$

Equation (5.17) is used for evaluating all observations of x_i, x_j, and y in matrices **a** and **b** using the training dataset to solve for the coefficients matrix Z. Six unknowns in matrix Z and six equations using equation (5.17) will compute the coefficient values. This process of finding the coefficients will be repeated for each z value in the new layer.

5.3 Enhancements to the GMDH algorithm

Over time, there are some enhancements that give the GMDH algorithm the ability to produce a better prediction. Two important enhancements are the use of only one dataset and the use of better stopping criteria. The classical GMDH algorithm was using two datasets, as shown earlier in equation (5.3). Some other methods used three datasets. Using two datasets will allow the GMDH to predict better properties, but it will be sensitive to noisy data. Using three datasets will solve the problem of sensitivity in the data, but might not have a good final prediction (Farlow, 1984a). In this research, I made the GMDH work using one dataset only. This was implemented using the "hold one case out" technique. The polynomial regression equation (5.1) will be used to fit the data using all of the training cases, but holding one case or observation out. Then, the prediction error will be done for the holdout case. At the end of this fitting and holding out procedure, the sum of squared errors across all holdout cases will be calculated (using equation (5.3)) and used as the final prediction error.

The adaptive learning network (ALN) approach is an enhanced algorithm that uses a stopping criterion that minimizes the squared error while avoiding model overtraining. ALN uses a predicted square error (PSE) to avoid dividing the data into two datasets and model overtraining. This gives the GMDH the ability to produce accurate and least complex networks (Barron *et al.,* 1984). PSE is the sum of two terms: the average squared error (FSE) and the overtraining or complex penalty (KP), as

$$\text{PSE} = \text{FSE} + \text{KP}, \tag{5.18}$$

The FSE term performed well in the initial stages of the prediction, but gave poor estimation for future observations. Therefore, FSE will decrease in time when the model becomes more complex and overtraining occurs. The KP term comes here to correct for a decrease in the FSE term when more coefficients are added to the model. The two terms perform a trade-off between model accuracy and complexity by obtaining the minimum expected squared difference between the estimated model and the true model for future datasets (Barron *et al.,* 1984). This will ensure that the final model is simple with low errors. Equations (5.19) show the FSE and KP terms.

$$\text{FSE} = \frac{1}{N}\sum_{n}[z(n) - y(n)]^2, \qquad \text{KP} = \text{CPM}\,\frac{2\sigma_p^2 K}{N}, \tag{5.19}$$

z (n) and y(n) is the real output and predicted output values respectively, N is the number of observation points in a data vector, K is the number of polynomial coefficients which increases in each layer, σ_p^2 is an estimate of the model error variance, and CPM is a complex penalty multiplayer. The prior estimate of model error variance σ_p^2 is set to be to the standard deviation calculated from the real output data. Large CPM values will produce simple but less accurate models. Small CPM values will produce more accurate models but might overfit the data. Figure 5.4 shows the trade-off that was used in equation (5.18). In figure 5.4, KP is increasing while FSE is decreasing. The network training and learning will stop when PSE reaches the optimum stopping point, as in Figure 5.4. PSE was used in my code to give a better prediction than the classical GMDH selection and stopping criterion method.

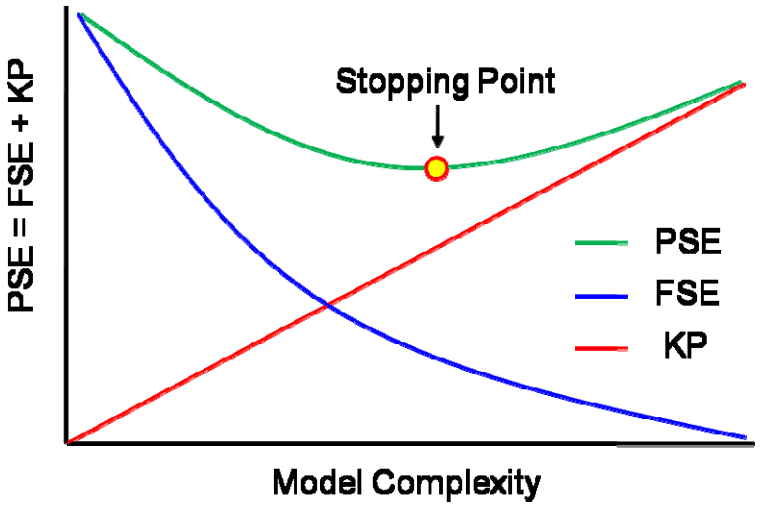


Figure 5.4. The predicted square error PSE. KP is increasing and FSE is decreasing with increasing complexity of the model. (Edited after (Montgomery and Drake, 1990)).

5.4 Application of the GMDH algorithm - Synthetic Example

Application is performed after a successful synthesizing of the GMDH network to reach the final trained model. A "c" code was written to join the GMDH algorithm and all enhancements in one code. The code reads all inputs and real outputs, and processes them to obtain the predicted outputs. Next I will train and apply the code using synthetic and real examples. The real examples will use seismic attributes data to predict reservoir properties such as geopressure and porosity. A comparison between the results of ANN and GMDH will be shown here.

Farlow (1984) applied the classical GMDH code to synthetic data obtained from a book called "Applied Regression Analysis" (Draper and Smith, 1966). The synthetic data consist of 27 observations. Farlow used about 52% (14 observations) of the data for training and the rest (13 observations) for checking and testing. The input to Farlow's code consists of 4 columns, which are the real output (y_r) and 3 variables (m). The first variable (x_1) is the radiation in relative gram calories/min during the preceding half-day of sunlight. The second variable (x_2) is the average soil moisture tension. The third variable (x_3) is the air temperature in degrees Fahrenheit. To simplify the problem, x_1 and

x_2 were divided by 100 and x_3 was divided by 10. Table 5.1 shows the input data that were used. Farlow used the classical GMDH algorithm to find the model that predicts the real output y. The root mean square (equation (5.3)) was used as a stopping criterion. The classical GMDH algorithm converged after one step or layer, with the minimum root mean square equal to 0.00525. This was achieved by using the input variables x_2 and x_3. Farlow calculated the square of the multiple correlation coefficients over the training samples, which happen to be equal to 0.82. This means that 82% of the variation of the real output y was described by the resultant quadratic Ivakhnenko polynomial equation (5.20).

$$y = 209.1 + 360.5x_2 - 26.36x_3 - 1263x_2^2 + 1.118x_3^2 + 27.69x_2x_3, \qquad (5.20)$$

I used the same data from Table 5.1 with my enhanced GMDH code. My code uses all of the training data with no specific part for checking or testing. All of the inputs were normalized to have a mean of zero and a variance of one. Normalization is important in making all the input data consistent. At the end of the prediction process, a unitizer is responsible for transforming the output values back to the range of real output values. The stopping criterion I used is the predicted square error from equation (5.17). The Barron equation (5.17) was applied on one observation which was not used during the training cycle. This is the method of leaving one observation out during the training. The final PSE will be the sum of the results produced from all of the observations that were left out. The enhanced GMDH algorithm converged after one step or layer, with the minimum PSE equal to 0.12163. This was achieved by using the input variables x_2 and x_3. x_2 and x_3 were also the same selected variables from Farlow's result. I calculated the square of the multiple correlation coefficients over the training samples which equal 0.92. This is 10% better than the classical GMDH method applied by Farlow. It also means that 92% of the variation of the real output y was described by the resultant quadratic Ivakhnenko polynomial equation:

$$y = 2.54 + 0.98x_2 - 0.19x_3 - 2.70x_2^2 + 0.07x_3^2 + 0.17x_2x_3, \qquad (5.21)$$

Table 5.1 Input to the GMDH algorithm. Y is the real output. X_1, X_2, and X_3 are the input variables. (From Draper and Smith, 1966)

Observations	**Real Output**			
	Y	**X_1**	**X_2**	**X_3**
1	110.4	1.76	0.07	7.8
2	102.8	1.55	0.07	8.9
3	101	2.73	0.07	8.9
4	108.4	2.73	0.07	7.2
5	100.7	2.56	0.07	8.4
6	100.3	2.8	0.07	8.7
7	93.7	1.84	0.07	8.7
8	96.6	1.98	0.02	7.6
9	96.2	0.8	0.02	6.7
10	99	0.8	0.02	6.2
11	88.4	1.05	0.02	7
12	75.3	1.8	0.02	7.3
13	77.1	2.3	0.02	8.2
14	65.7	1.91	0.47	8.3
15	56.8	1.91	0.47	8.2
16	62.1	1.91	0.47	6.9
17	53.2	2.13	0.47	7.6
18	59.4	2.13	0.47	6.9
19	58.7	1.51	0.47	7.5
20	58	2.05	0.47	7.6
21	102.8	2.8	0.07	7.4
22	98.9	2.16	0.07	8.8
23	99.4	0.59	0.02	6.5
24	92	1.8	0.02	6.5
25	82.4	1.77	0.02	7.6
26	74	2.03	0.47	7.6
27	61	0.76	0.47	7.4

To illustrate the differences between the classical and the enhanced code, Table 5.2 shows the detailed prediction of both algorithms. In Table 5.2, Y is the real output values, YPC is the predicted output values using the classical method, and YPE is the predicted output values using the enhanced method.

Table 5.2 Prediction of classical and enhanced GMDH algorithms. Y is the real output, YPC and AEC are the prediction and absolute error of the classical method. YPE and AEE are the prediction and absolute error of the enhanced method.

Observations	Real Output Y	Classical Prediction YPC	Enhanced Prediction YPE	Absolute Error of Classical AEC	Absolute Error of Enhanced AEE
1	110.4	105.699	104.023	4.701032	6.377041
2	102.8	99.37532	99.48747	3.424683	3.31253
3	101	99.37532	99.48747	1.62468	1.512527
4	108.4	110.2889	108.5847	1.888848	0.184656
5	100.7	101.9142	100.935	1.214233	0.235009
6	100.3	100.3238	99.94367	0.023777	0.356329
7	93.7	100.3238	99.94367	6.623783	6.243677
8	96.6	84.28122	83.86097	12.31878	12.73903
9	96.2	93.11639	92.90079	3.083603	3.299204
10	99	98.80759	99.35571	0.192413	0.355713
11	88.4	89.97005	89.51908	1.570045	1.11908
12	75.3	87.02499	86.5058	11.72499	11.2058
13	77.1	79.39751	79.67658	2.297509	2.576578
14	65.7	65.7002	62.15455	0.000198	3.545445
15	56.8	65.18997	61.8009	8.389973	5.000896
16	62.1	60.59239	60.92861	1.507606	1.17139
17	53.2	62.59836	60.53862	9.398357	7.338622
18	59.4	60.59239	60.92861	1.19239	1.528606
19	58.7	62.24472	60.47152	3.544719	1.771521
20	58	62.59836	60.53862	4.598358	2.538623
21	102.8	108.6694	106.9003	5.869417	4.100341
22	98.9	99.83848	99.69511	0.938478	0.795103
23	99.4	95.32579	95.35995	4.074212	4.040057
24	92	95.32579	95.35995	3.32579	3.359945
25	82.4	84.28122	83.86097	1.881217	1.460963
26	74	62.59836	60.53862	11.40164	13.46138
27	61	61.91341	60.44536	0.913406	0.554641

I plotted the curves shown in Table 5.2 to verify the enhancements. Figure 5.5 shows a scattered plot with 5 curves. The blue line represents the real data values Y. The red line represents the plot of the classical GMDH results YPC. The dotted red line represents the absolute error produced using the classical method AEP. The green line represents the prediction using the enhanced GMDH method YPE and the dotted green line represents the absolute error AEE produced from the enhanced method. Notice that both the classical and enhanced GMDH predictions follow the same trend. However, for

the observations from 14 to 22, the enhanced method showed better predictions. Also, the enhanced method showed lower absolute error for the range from 14 to 22 in the plot.

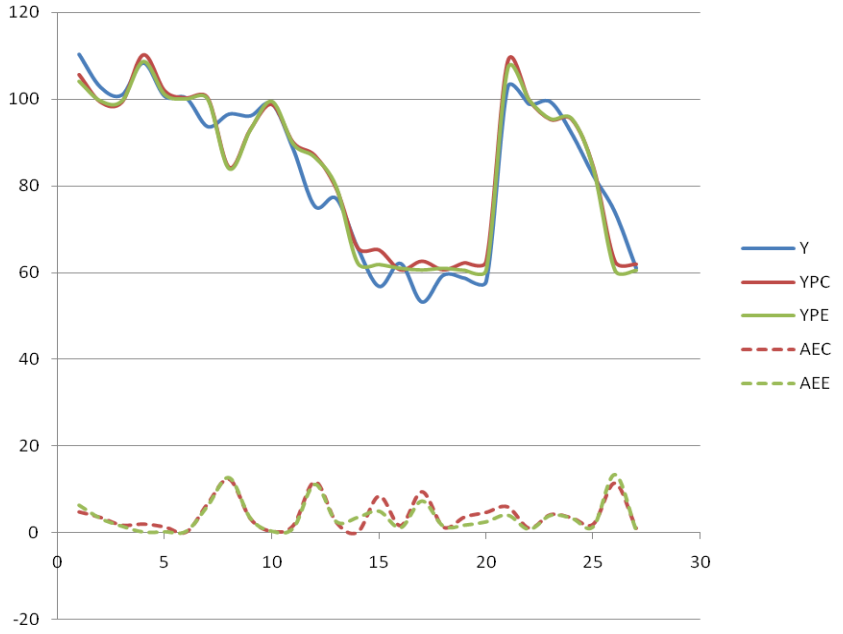


Figure 5.5 A comparison between the classical and the enhanced GMDH methods. Y is the real output, YPC and AEC are the prediction and absolute error of the classical method. YPE and AEE are the prediction and absolute error of the enhanced method.

5.5 Clastic reservoir porosity prediction (comparison to ANN)

I use a supervised ANN and GMDH algorithms here to predict porosity volume using information from well logs as an input. The supervised ANN is called multilayer perceptron (MLP). The data of porosity logs from 3 wells were used in the training stage. These 3 wells penetrate through the target reservoir covering about 290 ms of seismic data. With 74 samples of porosity from each well, the algorithms were trained on a total of 222 data samples. These porosity well logs values were used as the dependent variables or real outputs. The same total number of samples (222) were extracted from each of the input independent variables (seismic attributes). For each seismic attribute, I

extract the same number of samples from the same locations, to make an input vector. I have total 10 input vectors, as I use 10 seismic attributes to train the system. These 10 attributes are

$\mathbf{x}_1^{(0)}$: Seismic amplitude; $\mathbf{x}_2^{(0)}$: Seismic impedance.

$\mathbf{x}_3^{(0)}$: Coherence; $\mathbf{x}_4^{(0)}$: Instantaneous amplitude.

$\mathbf{x}_5^{(0)}$: Instantaneous bandwidth; $\mathbf{x}_6^{(0)}$: First envelope trace derivative.

$\mathbf{x}_7^{(0)}$: Instantaneous frequency; $\mathbf{x}_8^{(0)}$: Normalized phase.

$\mathbf{x}_9^{(0)}$: Instantaneous phase; $\mathbf{x}_{10}^{(0)}$: Second envelope trace derivative.

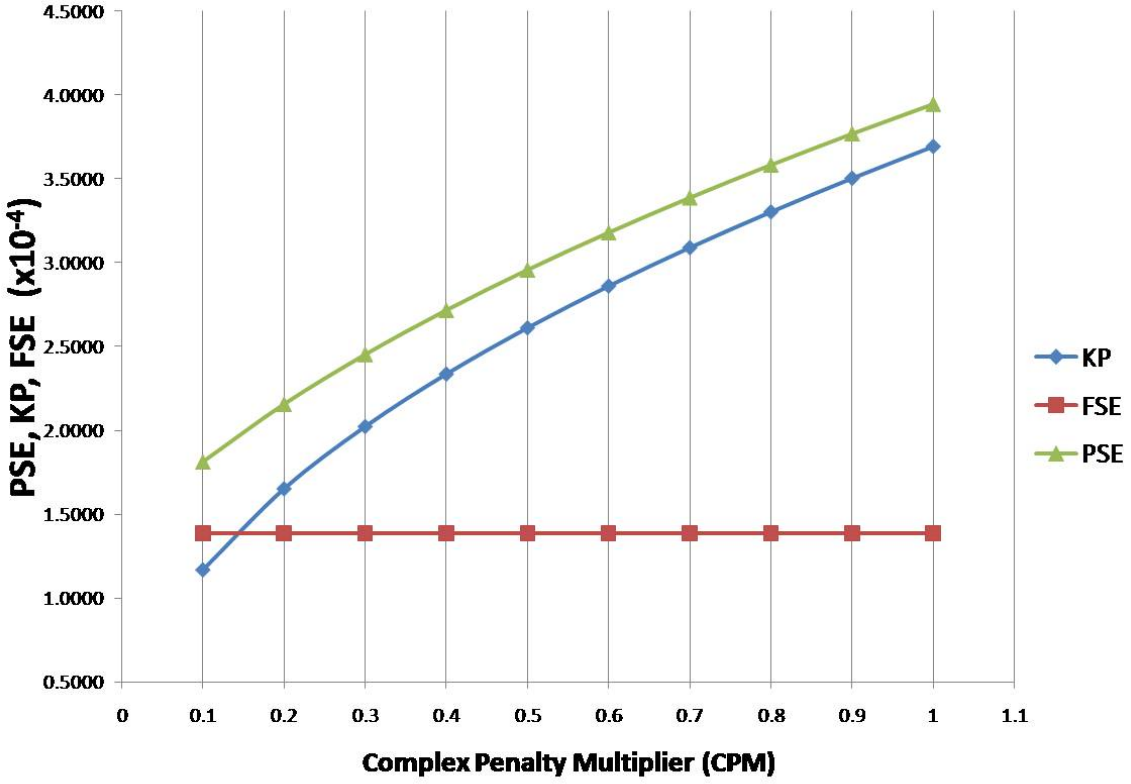
The superscript 0 means the layer 0, i.e. the original input. Accordingly, $\mathbf{x}_i^{(\ell)}$ means the ith input vector in the ℓth layer.

The training began with normalizing the input and output values by adjusting the seismic attributes to have a mean of zero and a variance of one. Normalization of the real outputs is needed and will be used to transfer the predicted values back to the real outputs range. 3 layers were used for the MLP algorithm. An input layer, a hidden layer, and an output layer. The weights between the input and hidden layers and between hidden and output layers are trained using a back-propagation method. Each node or perceptron is composed of two operations: summation and activation. The activation function that I used is called the sigmoid function. The parameters used during the MLP training were 10 samples in each input vector, one porosity desired output, and 6 nodes or neurons used for the hidden layer. The 10 input samples were extracted from the 10 seismic attributes at the same time interval. The other parameters, such as the learning parameter, were adjusted after trying many different values to produce the right values.

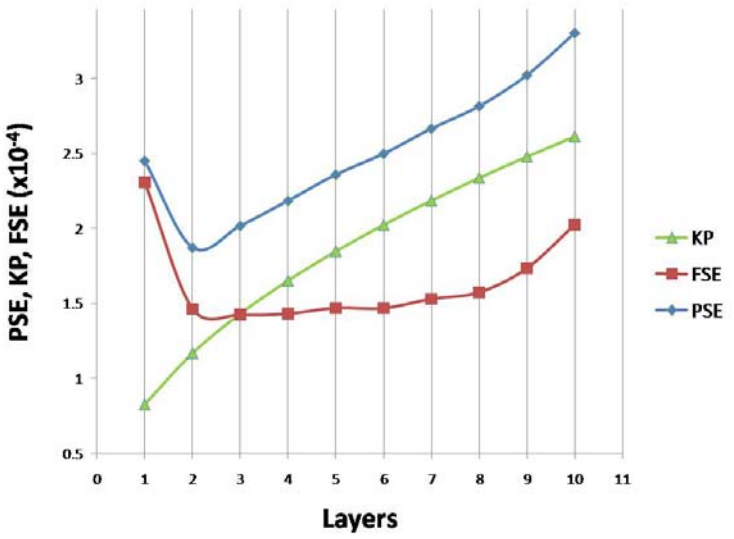
On the other hand, GMDH training converges after 2 layers. GMDH selected the most interesting seismic attributes among all the input 10 attributes. Instead of selecting only one output as in conventional implementation, we keep a group of outputs and take the median as the final prediction. In this way, we effectively include all of the available attributes in the prediction process and obtain an unbiased estimation.

I used many CPM values that range between 0.1 to 1.0. Figure 5.6a summarizes the analysis from real data. The horizontal axis is the CPM values we used for a real data set, and the vertical axis is the actual values for FSE, KP and PSE obtained from tests corresponding to different CPM. I noticed from Figure 5.6a that the lowest error of PSE can be achieved by using a low KP value. This guided me to use lower CPM values in the calculation.

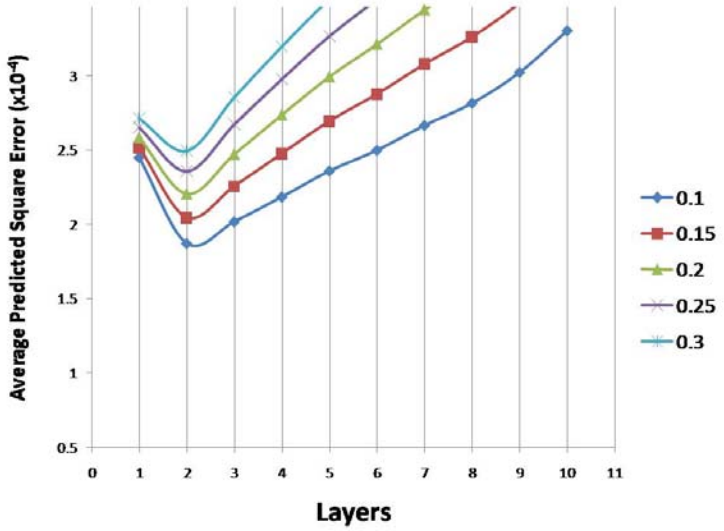
Figure 5.6b plots the average of the predicted square errors, average FSEs, and average KPs versus the layers for CPM equal to 0.1. Figure 5.6c shows the effect of CPM values on the values of PSE. It is found that setting CPM equal to 0.1-0.3 will enable layer two to have the minimum PSE. At this point, the GMDH predicts 72% of the variation in the real porosity logging data.



(a)



(b)

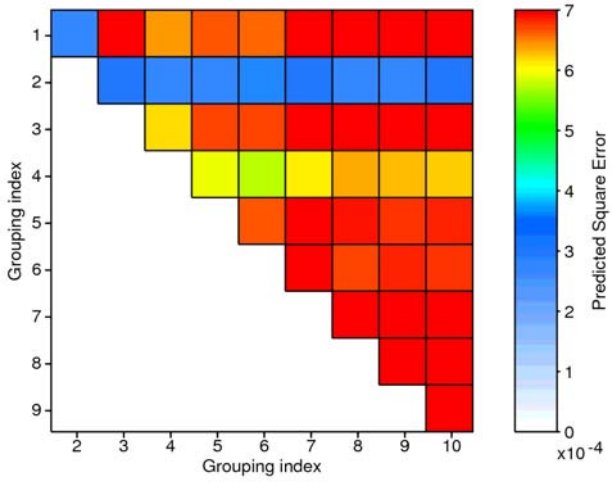


(c)

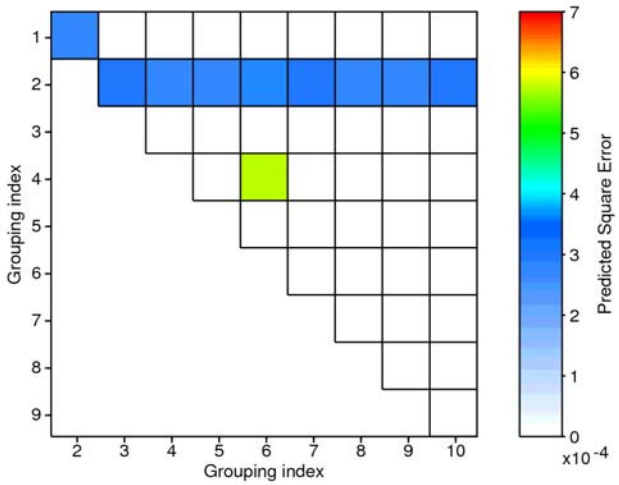
Figure 5.6 (a) The Effect of changing CPM values on the values of FSE, KP, and PSE. The horizontal axis is the CPM values we used for a real data set, and the vertical axis is the actual values for FSE, KP and PSE obtained from tests corresponding to different CPM. (b) The average PSE (black circle) for each layer (after using CPM = 0.1). The lowest PSE is at layer 2, where the training stops. (c) The effect of changing CPM values (0.1-0.3) on the values of PSE.

Figure 5.7a shows the relationship between the predictions in the first layer and the original input vectors: each pixel with index i in the vertical axis and j in the horizontal axis means this prediction originated from a pair (x_i, x_j). We have 10 attributes and thus 45 different combinations. The colour represents the predicted square error for each prediction. Figure 5.7b shows the 10 predictions with the lowest predicted square errors. We use these 10 predictions as the input to the next layer.

Figure 5.8a is the relationship in the second layer, and Figure 5.8b is 20 relations that have the lowest predicted errors. Since both the values of the PSE and the standard deviation for all of 20 relations are small, we take the median of 20 outputs as the final result.

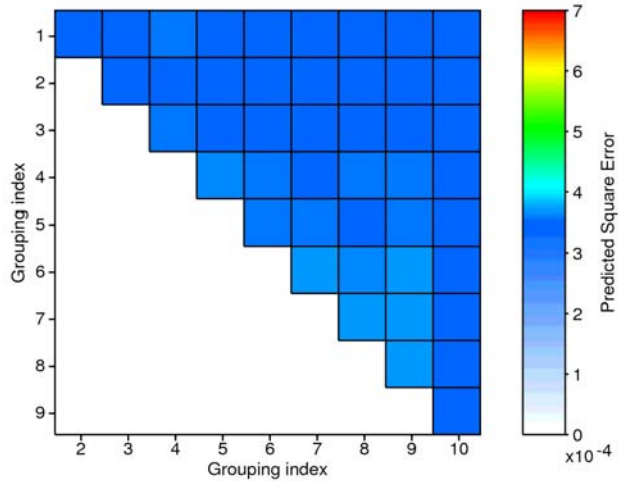


(a)

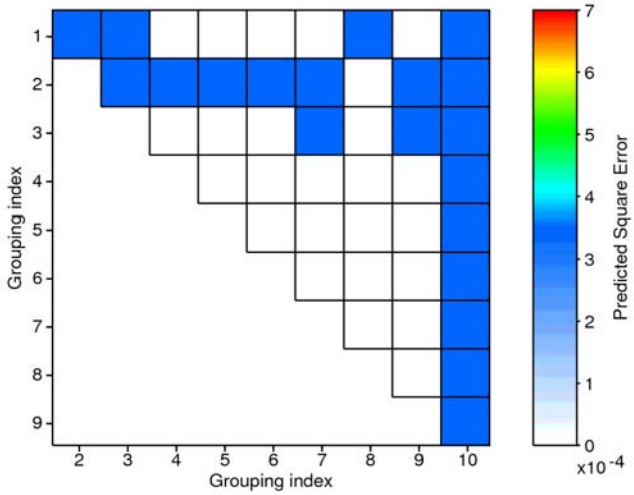


(b)

Figure 5.7. Predicted square errors for layer one. The vertical index *i* and the horizontal index *j* indicates that the output in the first layer is the combination of the *i*th and the *j*th vectors of the original input. (a) 45 relationships in layer one. (b) Ten relationships with the lowest predicted square errors at layer one. These 10 outputs will be used as the input to layer two.



(a)



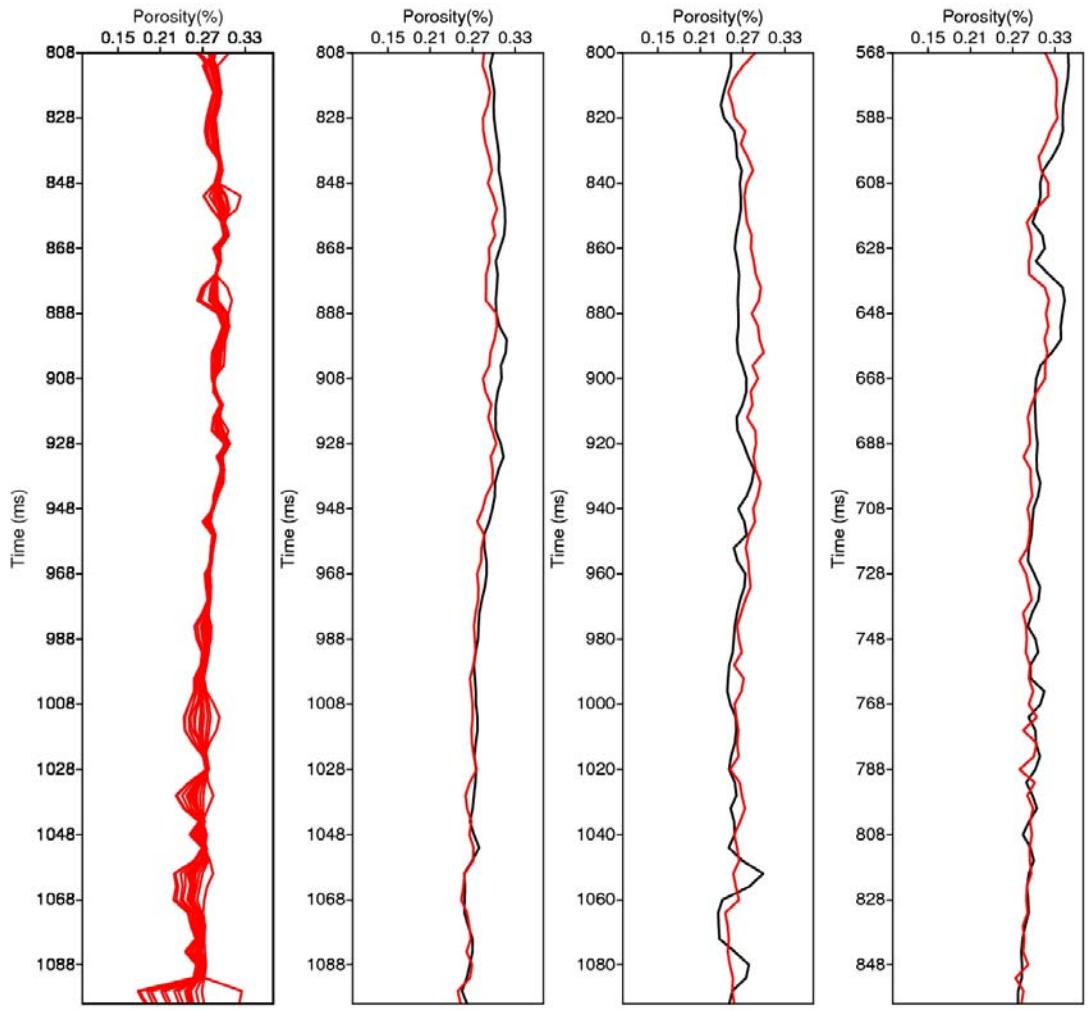
(b)

Figure 5.8. Predicted square errors for layer two. The vertical index *i* and the horizontal index *j* indicates that the output in the second layer is the combination of the *i*th and *j*th vectors in layer one. (a) 45 relationships in layer two. (b) 20 relationships with the predicted square errors less than 0.35×10^{-3} from layer two.

Figure 5.9a is a comparison between the porosity results of 20 relationships from layer two (using CPM equal to 0.1). Figure 5.9b is the median curve of the 20 curves from Figure 5.9a, and is compared to real porosity log. Figures 5.9c and 5.9d also show well-logging porosity and the GMDH predictions at other two well locations. The comparison suggests that the GMDH predicted porosity (red curves) is very reasonable, and follows the trend of the original porosity logs (black curves). The correlation coefficients of the three wells are 0.95, 0.50 and 0.83 respectively. The second well (Figure 5.9c) has a low correlation coefficient because of the spikes on the lower part of the real porosity log. The overall correlation coefficient for all 3 wells together equals to 0.85.

Figure 5.10 shows the standard deviation and the predicted square errors of 20 relations with lowest PSE from layer two. As the PSEs and standard deviations of these 20 relations are all very small, we use all of them from layer two and take the median as the final result. This is different from conventional implementation in which only one of the outputs is selected as the final result. By doing so, we effectively include all of the 10 original seismic attributes in the prediction process. Figure 5.11 clearly shows that the 20 relations from layer two are connected to the 10 original variables.

For the purpose of comparing the MLP and GMDH results, I extracted some traces from the two predicted porosity volumes. Figure 5.12 shows a well that was used by MLP and GMDH during the training process. The horizontal axis is the porosity values and the vertical axis is time (in ms). The black curve in Figure 5.12 is the original porosity from the well logs. The dotted red curve is the predicted porosity using the supervised ANN algorithm (MLP). The solid red curve is the predicted porosity using the GMDH algorithm. These red curves were extracted from the predicted porosity volumes. The overall prediction was quite good, and the red curves (dotted and solid) follow the trend of the original porosity logs (black curves). We noticed from Figure 5.12 that the GMDH predicted porosity (red curves) is very reasonable when compared with the original porosity logs (black curves) and the MLP predicted porosity (dotted red curves).



(a) (b) (c) (d)

Figure 5.9. (a) 20 predicted porosity results based on 20 relationships from layer two. (b) The median value curve (in red) of those 20 outputs (as the final prediction), which is compared with the true porosity curve (in black). (c-d) Comparison between the prediction and true curves from the other two porosity logs. The correlation coefficients between two curves in (b), (c) and (d) are 0.95, 0.50, and 0.83 respectively.

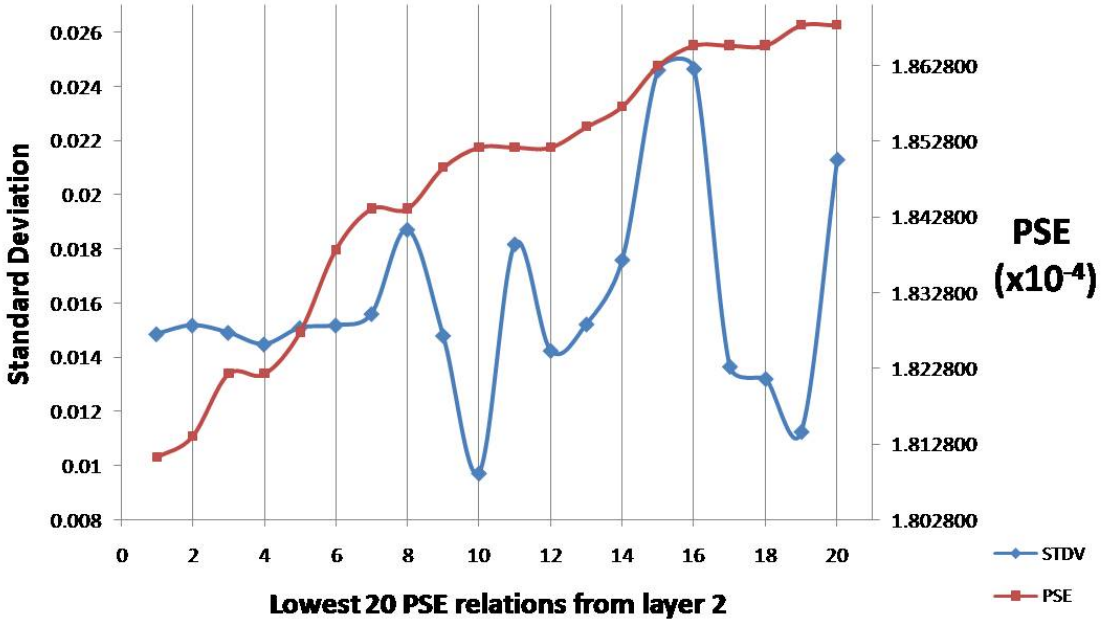


Figure 5.10. The standard deviations and the predicted square errors of 20 relations with lowest PSEs from layer two. As all of these 20 outputs have very small PSEs and deviations, the median values of them are selected to form the final prediction.

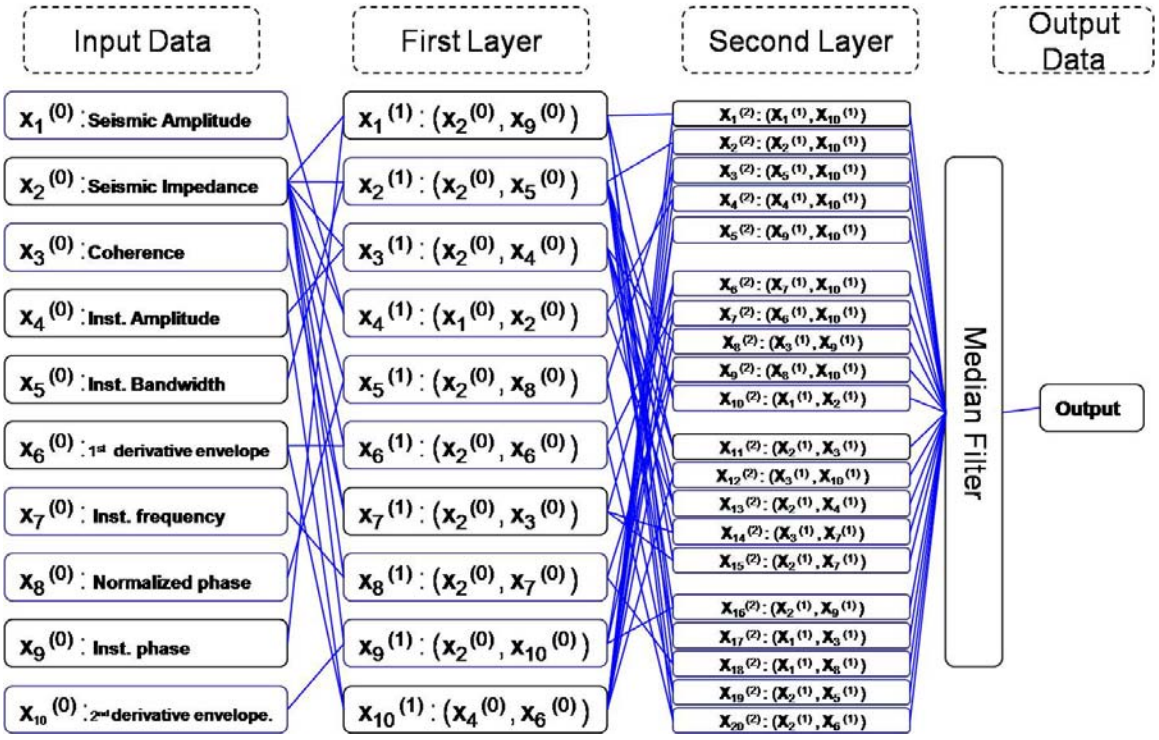


Figure 5.11. The structure of the GMDH network for porosity prediction. The input consists of 10 seismic attributes, and the output is the median filtered result of 20 combinations in layer two; in this way GMDH effectively uses all of the 10 seismic attributes in the prediction process.

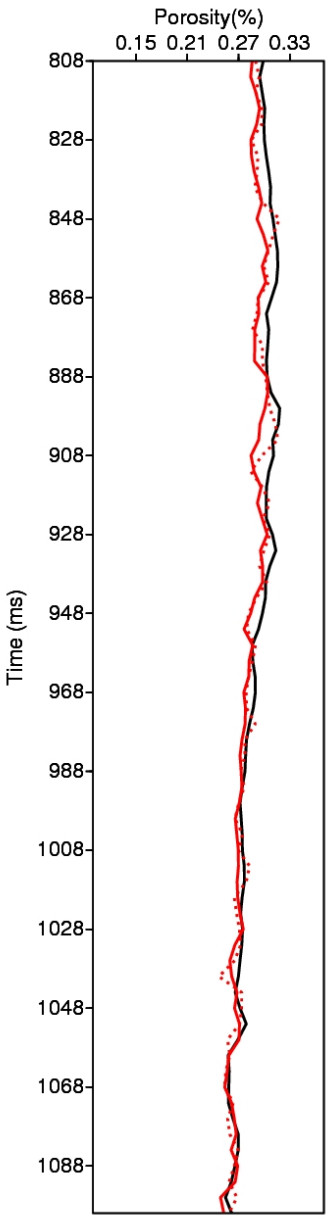
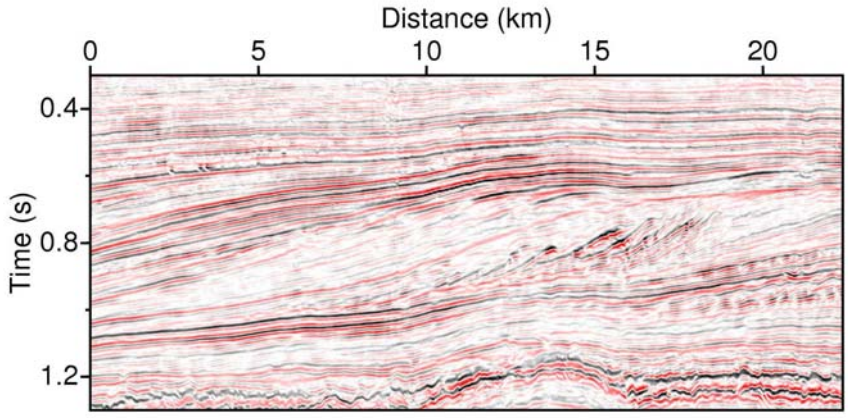
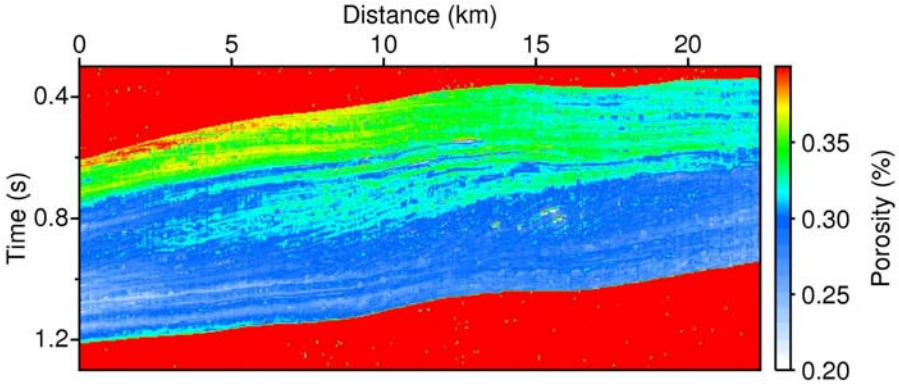


Figure 5.12. Cross section of a porosity logs. Black curve is the original porosity log. Dotted red curve is extracted from the porosity volume that was predicted using supervised ANN or (MLP).Red curve is the predicted porosity extracted from the GMDH porosity volume.

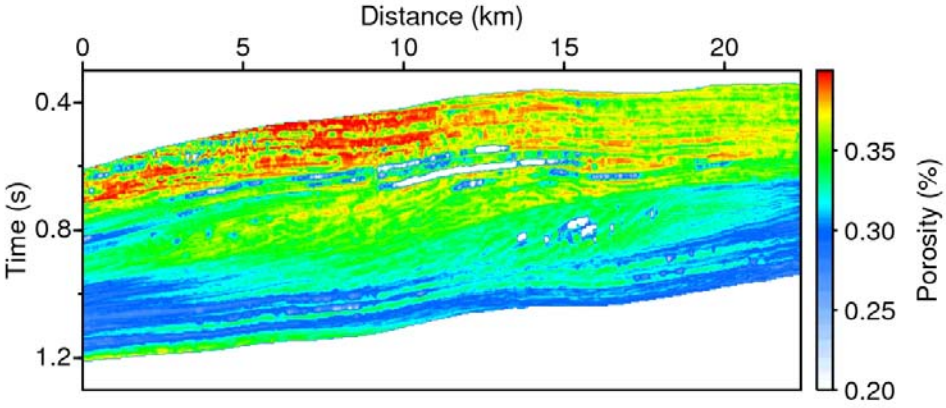
Figure 5.13a is a 2D line extracted from a seismic volume. The length of the 2D line is about 22 km and the time shown is from 0.3 to 1.3 s. Figure 5.13b is showing the same 2D line extracted from the porosity volume that was predicted using MLP neural network algorithm. Figure 5.13c is the GMDH algorithm porosity prediction on the formation of interest which is shallow on the right side of the line (0.4 to 1.0 s) and deep on the left side of the line (0.6 to 1.2 s). The overall prediction is very good. It shows a high porosity formation in the upper part of the section (0.6 s). It also shows a gradual decrease in porosity from shallow to deep part. Both the MLP and GMDH perform a good porosity prediction and showed a consistency of results.



(a)



(b)



(c)

Figure 5.13. (a) A 2D line extracted from the original seismic amplitude volume. (b) The same 2D line extracted from the porosity volume that was predicted using MLP algorithm. (c) The porosity section, predicted using the GMDH algorithm.

5.6 Carbonate reservoir geopressure prediction

In this section, I use the GMDH algorithm to predict geopressure. As mentioned before, the GMDH algorithm should have the power of both unsupervised and supervised ANN. This is because the GMDH is a self-organized method and uses the multilayer supervised method in the training and learning process. I use the same data (Saudi Aramco data) that were used in the previous chapters. As a summary of what was mentioned in chapter 1, the total thickness of the formation is about 900 feet. Throughout the field, the depth of the target formation varies between 9700 and 10500 feet. The study area encompasses a 3-D seismic volume consisting of 411 inlines and 226 CDPs. The extent of the field is about 200 km^2. Data were processed at 2ms. The zone of interest is equal to 100 ms on the seismic time section. Geopressure data, in terms of occurrence or non-occurrence of abnormally high formation pressures across the target carbonate formation, are available from 19 wells. Among the 19 wells, only 10 wells have over-pressure and the remaining 9 wells have normal pressure. The geopressure data from these 19 wells were used to represent the real outputs of the GMDH model.

The input variables to the GMDH algorithm were 14 seismic attributes. These attributes are:

1. Seismic amplitude.
2. Seismic impedance.
3. Porosity.
4. Coherence.
5. Instantaneous amplitude.
6. Instantaneous phase.
7. Instantaneous frequency.
8. Frequency weighted envelope (average frequency).
9. Thin-bed (instantaneous frequency – average frequency).
10. Instantaneous bandwidth.
11. Normalized phase.
12. First envelope trace derivative.
13. Second envelope trace derivative.
14. Instantaneous Q factor.

The values of these seismic attributes were extracted at the top of the overpressured formation. In this way, each of the 19 training wells will have 14 values (one value from each seismic attribute). In other words, the inputs to the GMDH algorithm were 19 observations (19 wells) and each observation has one dependent variable (mud weight value) and 14 independent variables (seismic attributes). Since the input data are from a different range of values because of different seismic attributes, I started by normalizing the input and output values. After normalization has been done, each seismic attribute is adjusted to have a mean of zero and a variance of one. The mean and variance of the real output values will be used to de-normalize or unitize the predicted outputs.

After the completion of the training, GMDH was smart enough to select the most interesting seismic attributes among all the input 14 attributes. As mentioned earlier, GMDH is very effective in selecting which inputs are important to the modelling system. This will save the application time, because GMDH will be applied to the selected seismic attributes only. Among the 14 seismic attributes input, only 5 attributes won the competition. These attributes are porosity, coherence, instantaneous amplitude, instantaneous frequency, and thin-bed. Figure 5.14 shows the structure of the GMDH network for predicting geopressure. In this figure, the system converged after 3 layers with the minimum PSE equal to 1.2, and the square of the multiple correlation coefficients over the training samples equal to 0.85. In Figure 5.14, N_i is the normalization equation used for this specific node, F_i is the quadratic polynomial function, and U is the unitizer equation. These equations are shown in table 5.3. Figure 5.15 shows the final result of the GMDH prediction. This is an estimation of the geopressure on the area extracted from the top of the target formation. The dots in the figure represent the real geopressure values. Solid dots indicate the overpressure encountered while drilling this specific well and hollow dots indicate normal pressure wells at the reservoir level. Figure 5.15 shows some good indication of the broader picture of geopressure in this area. The red areas on the figure represent high pressure and blue areas indicate normal pressure. Not all high pressure wells fall under the high pressure zones and not all normal pressure wells fall under the normal pressure zones. This is because of the limitation we have when training the GMDH algorithm. This could be avoided by having more information about the geopressured wells. Mud weight well logs might give a better

prediction when used in the training because of more information having been added to the system.

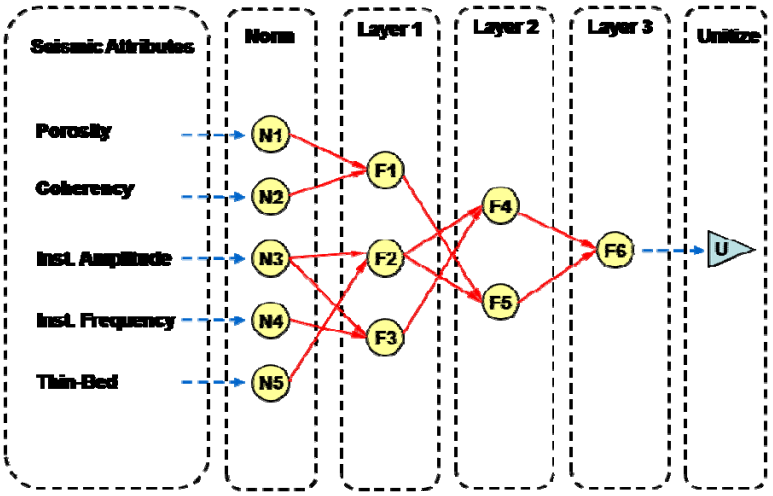


Figure 5.14 The final structure of the GMDH network for predicting geopressure. Explanations of N, F, and U are given in Table 5.3.

Table 5.3 Equations used to predict geopressure.

Equations used in GMDH application

Normalization

N1 = -4.77 + 63.4 Porosity
N2 = -0.855 + 12.3 Coherency
N3 = -2.31 + 3.5 Inst. Amplitude
N4 = -4.5 + 0.133 Inst. Freq.
N5 = 18.42 + 0.07 Thin-bed

Quadratic Polynomial Equations

$F1 = 0.078758 + 0.002123N1 + 0.614662N2 + 0.140287N1^2 - 0.039299N2^2 + 0.798846N1N2$
$F2 = -0.144612 - 0.514880N3 - 0.252699N5 + 0.447174N3^2 - 0.062710N5^2 + 0.354011N3N5$
$F3 = -0.083830 - 0.627092N3 - 0.227673N4 + 0.527762N3^2 - 0.206325N4^2 + 0.413367N3N4$
$F4 = -0.206805 - 0.739910F2 - 0.255891F3 + 9.258914F2^2 + 4.959558F3^2 - 13.435809F2F3$
$F5 = -0.375864 + 0.650669F1 - 1.521130F2 + 0.887426F1^2 + 0.751430F2^2 - 1.392114F1F2$
$F6 = -0.303773 + 0.490995F4 - 0.296667F5 - 0.052240F4^2 + 1.083465F5^2 - 0.583500F4F5$

Unitizer

U = -1.96 + 0.02 F6

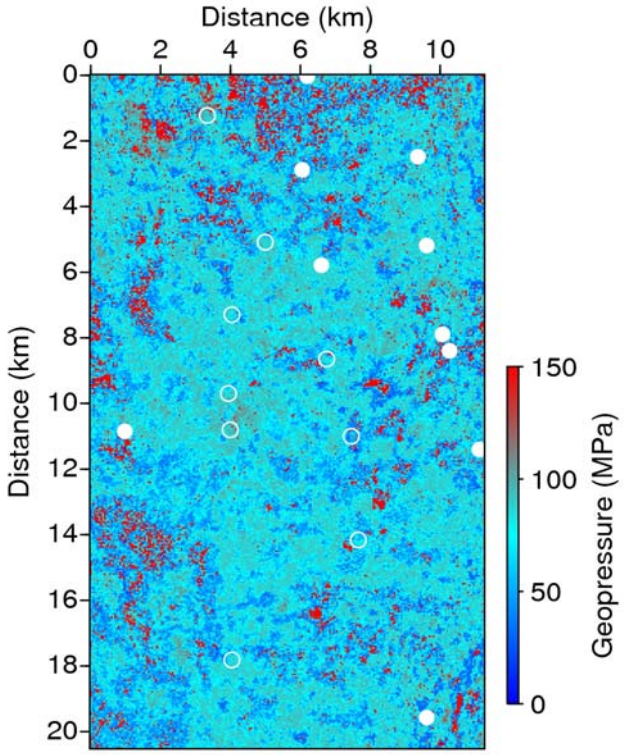


Figure 5.15. Geopressure prediction by using the GMDH algorithm

5.7 Porosity prediction in carbonate reservoirs

I use the GMDH algorithm here to predict porosity volume using information from well logs as an input. Porosity logs from 7 wells are used in the training stage. These 7 wells penetrate through the target carbonate reservoir covering about 140 ms of seismic data. With 71 samples of porosity from each well, the GMDH algorithm was trained on a total of 497 data samples. These porosity well logs values are used as the dependent variables or real outputs. The same total number of samples (497) is extracted from each of the input independent variables (seismic attributes). I use the following 13 seismic attributes in the training stage:

1. Seismic amplitude.
2. Seismic impedance.
3. Coherence.
4. Instantaneous amplitude.
5. Instantaneous phase.
6. Instantaneous frequency.
7. Frequency weighted envelope (average frequency).
8. Thin-bed (instantaneous frequency – average frequency).
9. Instantaneous bandwidth.
10. Normalized phase.
11. First envelope trace derivative.
12. Second envelope trace derivative.
13. Instantaneous Q factor.

The GMDH training began with normalizing the input and output values by adjusting the seismic attributes to have a mean of zero and a variance of one. Normalization of the real outputs is needed and will be used to transfer the predicted values back to the real outputs range. At the end of training, GMDH selected the most interesting seismic attributes among all the input 13 attributes. Among the 13 seismic attributes used, GMDH selected only 6 attributes to represent the prediction of porosity. These attributes are seismic amplitude, seismic impedance, coherence, instantaneous amplitude, instantaneous bandwidth, and normalized phase. Figure 5.16 shows the structure of the GMDH network for predicting porosity. The GMDH system for predicting porosity converged after 3 layers, with the minimum PSE equal to 414 and square of the multiple correlation coefficients over the training samples equal to 0.18. In Figure 5.16, N_i means the normalization equation used for this specific node, F_i is the quadratic polynomial functions, and U is the unitizer equation. These equations are shown in Table 5.4.

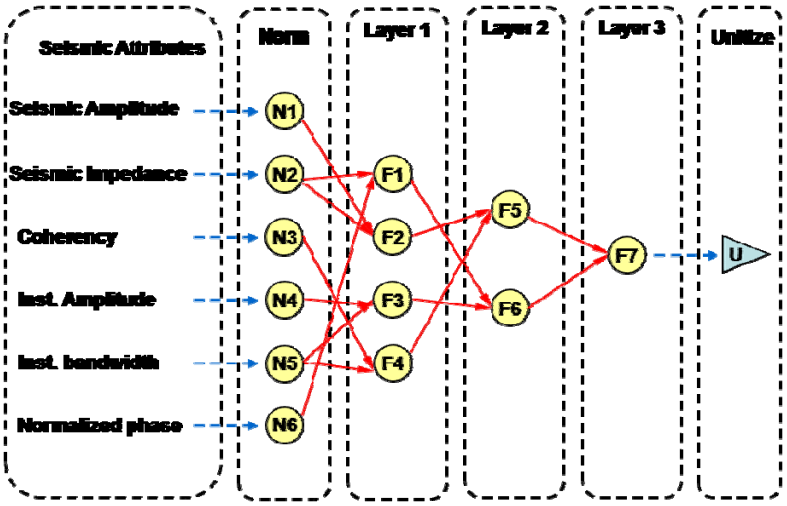


Figure 5.16 The final structure of the GMDH network for predicting porosity. Explanations of N, F, and U are in table 5.4.

Table 5.4 Equations used to predict porosity.

Equations used in GMDH application

Normalization

N1 = 0.008 + 2.4 Seismic Amplitude
N2 = -6.2 + 1.5 Impedance
N3 = -1.47 + 10.45 Coherency
N4 = -1.9 + 3.65 Inst. Amplitude
N5 = -0.9 + 0.099 Inst. Bandwidth
N6 = 0.01 + 1.39 Normalized phase

Quadratic Polynomial Equations

$F1 = 0.365964 - 0.265476N2 + 0.016925\,N6 + -0.030072N2^2 - 0.150317N6^2 - 0.135597N2N6$
$F2 = 0.251942 - 0.042016N1 - 0.280273N2 - 0.021352N1^2 - 0.024378N2^2 - 0.065065N1N2$
$F3 = -0.044748 + 0.153837N4 - 0.030317N5 - 0.155562N4^2 - 0.146390N5^2 - 0.565016N4N5$
$F4 = -0.050523 - 0.176191N3 + 0.032014N5 - 0.072734N3^2 - 0.023578N5^2 + 0.276819N3N5$
$F5 = -0.026812 + 1.096241F2 + 1.279934F4 + 0.152179F2^2 + 0.303276F4^2 + 0.346029F2F4$
$F6 = -0.091599 + 0.960369F1 + 1.043577F3 + 0.215909F1^2 + 0.988259F3^2 + 0.405363F1F3$
$F7 = 0.019299 + 0.628166F5 + 0.643520F6 - 0.325356F5^2 - 0.841927F6^2 + 1.534310F6F6$

Unitizer

U = -1.4 + 18 F7

For the purpose of validating the results, I extract some traces from the generated porosity volume. Figure 5.17 shows the 7 wells that GMDH used during the training process. The horizontal axis is the porosity values and the vertical axis is time (in ms). The black curves in Figure 5.17 are the original porosity from the well logs. The red curves are the predicted porosity using the GMDH algorithm. These red curves are extracted from the predicted porosity volume. The overall prediction is quite good, and the red curves follow the trend of the original porosity logs (black curves). Since these 7 wells are used on the GMDH training, it is sometimes difficult to validate the system using these wells because of the generalization problem. The GMDH might memorize these 7 wells and give good prediction only when these 7 wells are used as an input. Hence, the next step was to extract porosity curves at the same locations of wells that had never been introduced to the GMDH training process. I extract 3 porosity curves from 3 different wells in the area, after making sure that these wells have not been used during training. Figure 5.18 shows the 3 extracted curves. I noticed from Figure 5.18 that the predicted porosity (red curves) is very reasonable when compared with the original porosity logs (black curves). Figure 5.19 shows a 2D line extracted from the predicted porosity volume.

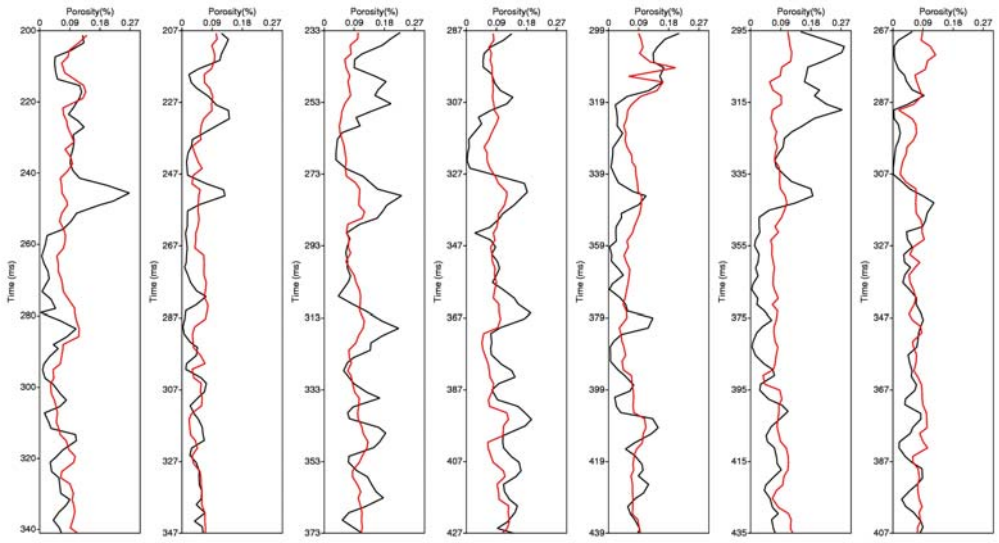


Figure 5.17 Cross section of 7 porosity logs used during training. Black curves are the original porosity logs and red curves are the predicted porosity extracted from the final porosity volume.

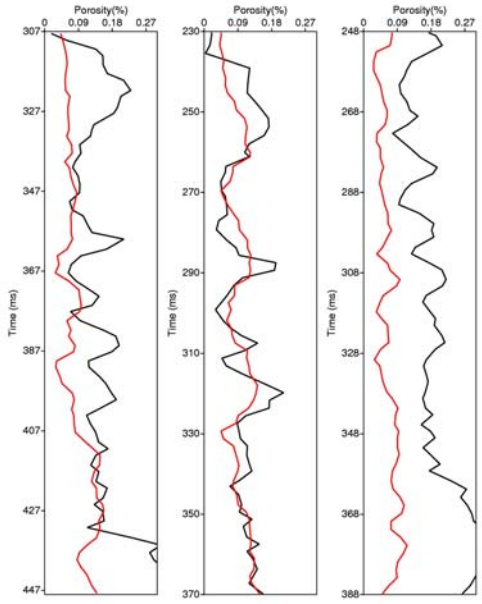


Figure 5.18 Cross section of 3 testing porosity logs that were never used during training. Black curves are the original porosity logs and red curves are the predicted porosity extracted from the final porosity volume.

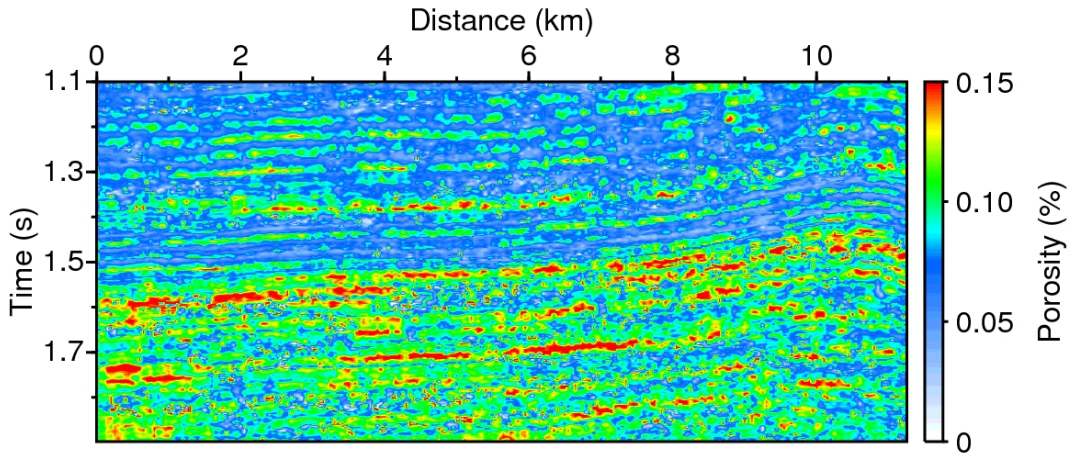


Figure 5.19 2D line extracted from the final porosity volume.

5.8 Summary

In this chapter, I use the GMDH algorithm to predict reservoir properties from clastic and carbonate seismic reservoirs. The GMDH algorithm is used here in order to find a non-linear relation between seismic attributes and pore pressure or porosity. In the geopressure prediction, I use real pressure at the top of the target formation as a dependent variable in my GMDH system. For the carbonate reservoir, the porosity prediction has been done after using the real porosity logs from 7 different wells in the area. The independent variables used are 14 seismic attributes. The GMDH algorithm can do a good job in predicting geopressure and porosity. The final outputs are reasonable and helpful. For a better prediction of geopressure, one needs more information about the geopressure in the area, including mud weight well logs. The porosity prediction is successful when compared with the original porosity well logs. GMDH shows an excellent results when compared to the results of ANN.

Chapter 6

Reservoir characterizations of Jilh carbonate formation

In this thesis, seismic properties were predicted using artificial intelligent methods such as neural networks and group methods for data handling. A carbonate reservoir or formation was used most of the time as an input to the prediction methods. The carbonate formation is called the Jilh formation and is extracted from Uthmaniyah portion of Ghawar field in Saudi Arabia. It is important to study the Jilh formation because many important producing gas and oil formations are underlying Jilh. The prediction methods were applied into Jilh carbonate formation to predict porosity and pore pressure. In this chapter, I am going to give more geological information about the Jilh carbonate formation. Then a discussion of the prediction results will be shown using geological interpretation and reservoir characterization.

6.1 Jilh carbonate formation

The Ghawar field is the world's largest field, producing about 65% of all Saudi oil. Ghawar is about 250 km (155 mile) long and 30 km (18.5 mile) wide. Geologists are saying that Ghawar has more than 300 m (1000 ft) of structural closure. Figure 6.1 is showing the Ghawar field which is divided into smaller parts or fields. These production fields are, from north to south, Ain Dar and Shedgum, Uthmaniyah, Hawiyah, and Haradh. Our study area is called Uthmaniyah which is one of the important oil and gas areas in Ghawar field. The carbonate formation under the study is called Jilh formation.

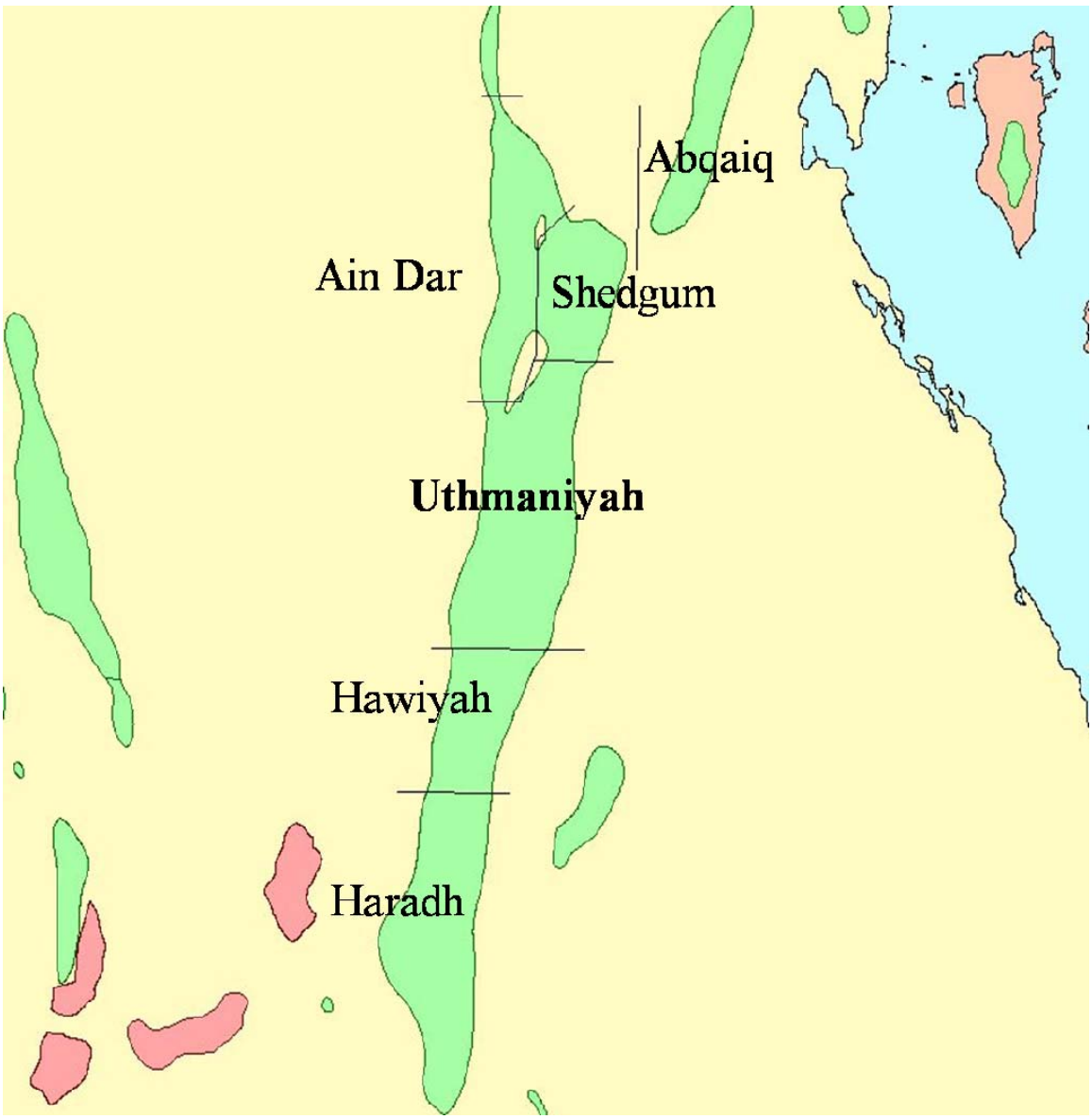
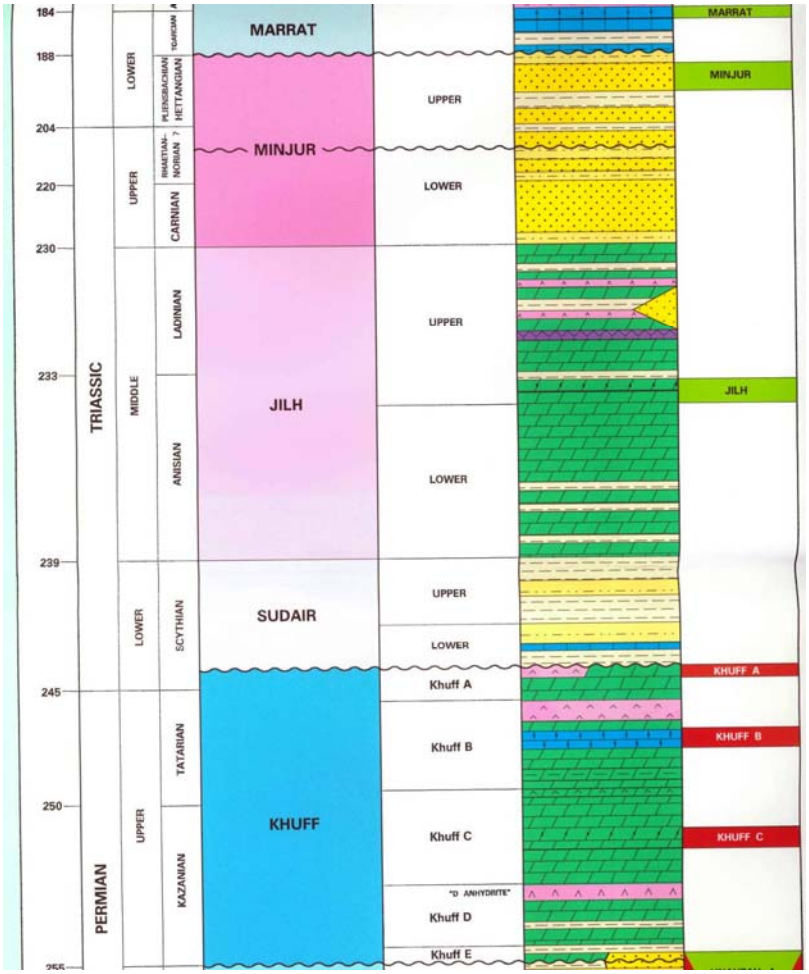


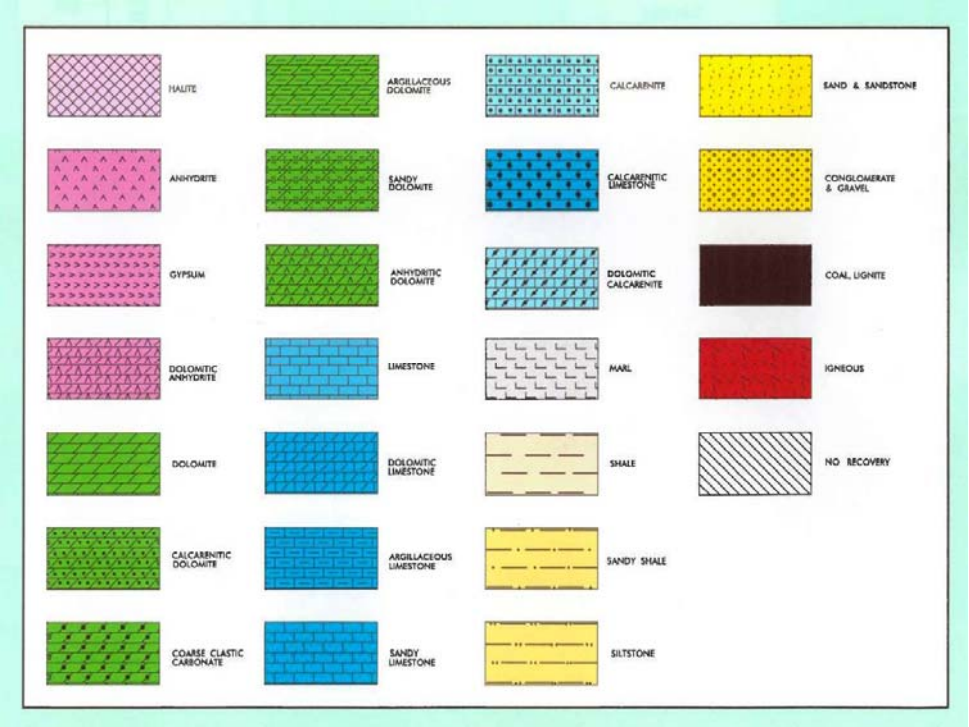
Figure 6.1 Uthmaniyah portion of the Ghawar field. The fields are located east of Saudi Arabia.

Jilh formation or Jilh Limestone member is occurring from 8000 ft (2438.4 meters) of true vertical depth (TVD) to 10000 ft (3048 meters) TVD. Depending on the examined fossils and the correlation with dated subsurface sections, the Jilh formation is aged to Lower and Middle Triassic. Some fossils were found between 45 and 55 meters from the top of the formation which includes *myophoria* and fragments of *ammonites*. Some of the lithology rock types that can be found in Jilh formation are dolomite, shale, anhydrite

stringers, and sandstone. Jilh was deposited with little marine influence. The underlying formation of Jilh is called Sudair formation with a conformable contact. Sudair formation consists of beds of shale interbedded with dolomite and evaporates. The overlying formation of Jilh is called Minjur formation which is thick siliciclastic rocks. The Minjur sandstone was deposited during periods of regression. Figure 6.2 shows the stratigraphic column of the Minjur, Jilh, and Sudair formations. Wells are to be drilled through Jilh formation to reach some important gas producing formations such as Khuff formation.



(a)



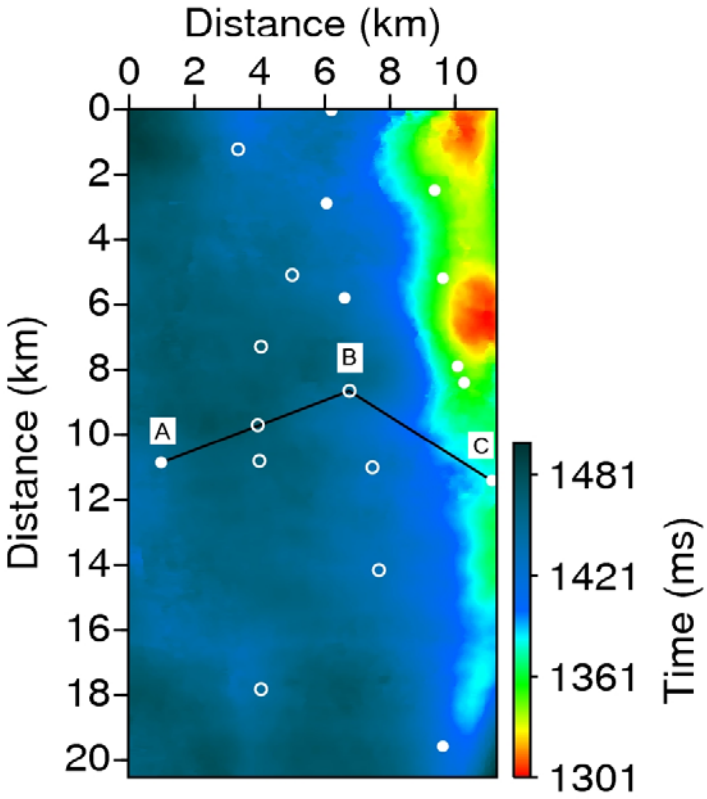
(b)

Figure 6.2. (a) Stratigraphic column of Minjur, Jilh, and Sudair formations. (b) The colour index or legend of (a) which is showing the lithology type at each formation. Khuff formation is an important gas producing zone which is under the Sudair formation.

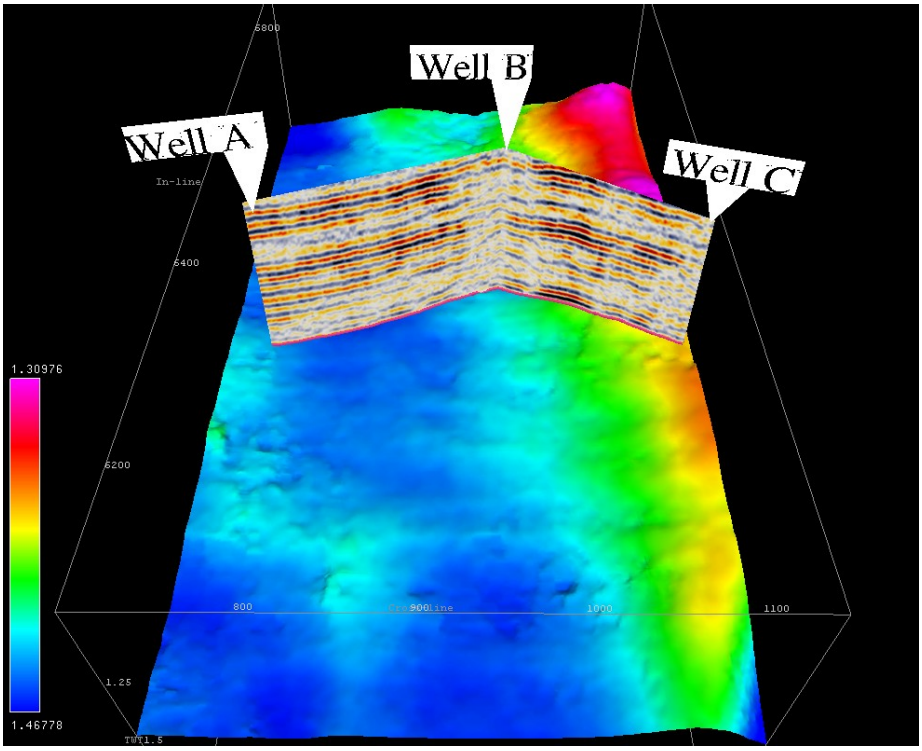
Arkell (1952) split the Jilh formation sequence into two parts which are the upper and lower Jilh. This classification was done on the basis of lithology. The upper Jilh of the subsurface consists of dolomite, coarse clastic carbonate, shale, anhydrite and very little sandstone. The Lower Jilh of the subsurface is mostly dolomite with some strips of shale.

Jilh formation was introduced in chapter 2 when I discussed the 3D seismic data from Saudi Aramco oil company. Figure 6.3 shows the 3D seismic area. Figure 6.3a shows a top view map (basemap) of the study area with 19 wells. These wells represent the available well data that will be used in the reservoir properties prediction process. The background colour of the basemap is the time structure map that was picked on top of the Jilh formation. A traverse line (random line) was picked through 3 wells which are

namely wells A, B, and C. Figure 6.3b shows the 3D image of the study area with a cross section of seismic amplitude data going across wells A, B, and C.



(a)



(b)

Figure 6.3. (a) A top view map of the study area with a horizon picked on top of Jilh formation. The white dots (hollow and solid) are 19 available wells. A random line is connecting wells A, B, and C. (b) A 3D view of (a) with posting a cross section of seismic amplitude data.

Figure 6.4 shows the 3 wells with Jilh formation subsurface. The 3 wells represent about 10 km of distance. Jilh formation was quite thick and stable throughout the study area when going from west to east direction. Some well logs data are presented in Figure 6.4 but not very clear because of the quality of the original picture supplied from the oil company. Usually these logs represent sonic, density and porosity logs. Figure 6.5 shows three time slices displayed with the Jilh horizon. The time slices were made at 1.39, 1.430 and 1.447 sec. The figure shows that top of Jilh formation is a high structure on the east part and tends to be flat on the west part.

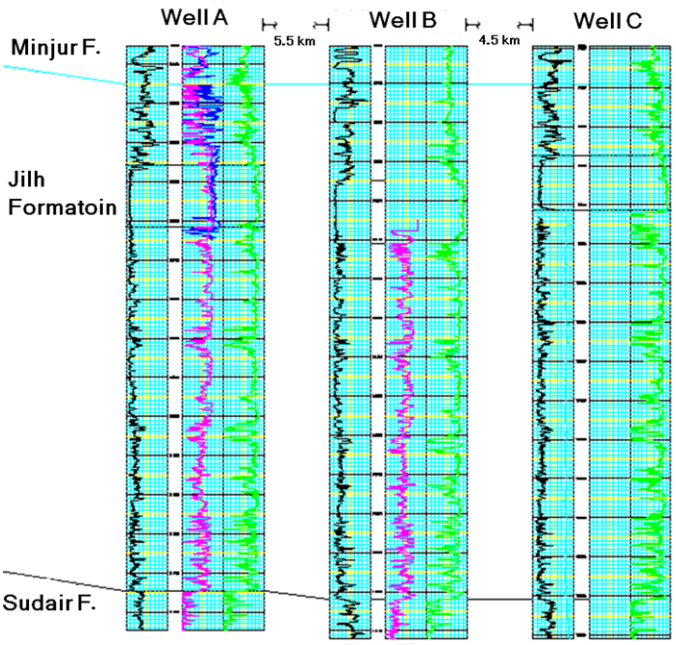
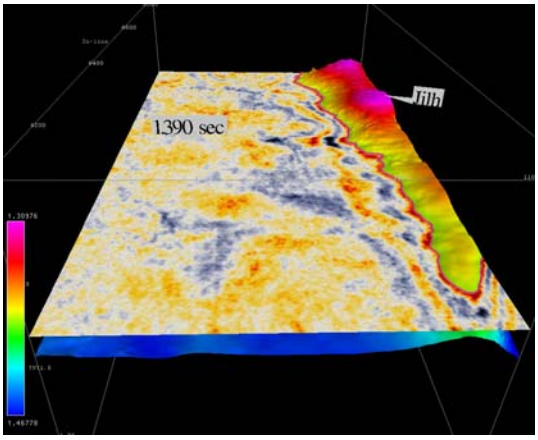
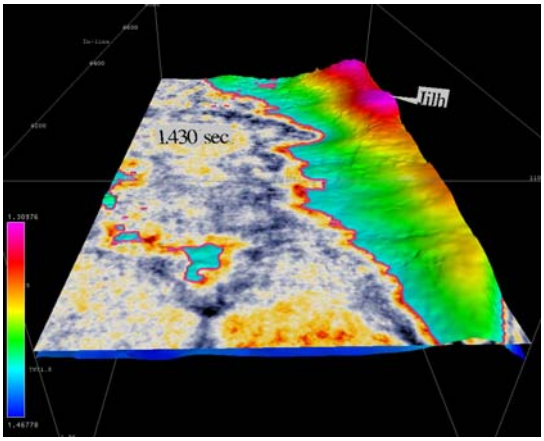


Figure 6.4. Three wells from Figure 6.3 showing a cross section of Jilh formation. Well A from west of the study area is about 10 km away from well C. Some well logs such as porosity, density, and sonic are displayed.

(a)



(b)



(c)

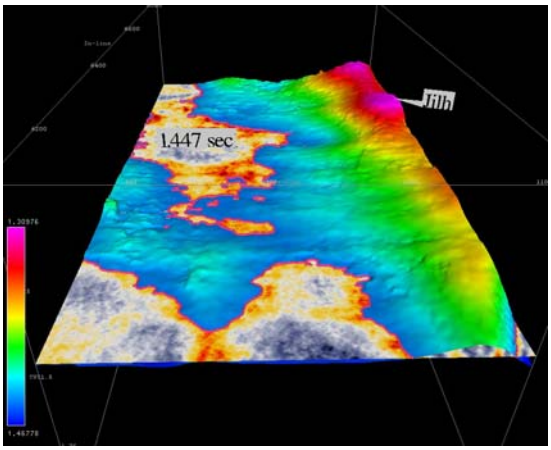


Figure 6.5. Time slices and Jilh horizon displayed on a 3D view. (a) Time slice at 1.39 sec. (b) Time slice at 1.43 sec. (c) Time slice at 1.447 sec. Jilh formation is clearly showing high structures to the east.

The importance of studying the Jilh formation arises when thinking not only about the Jilh but about the producing formations underlying the Jilh. The Jilh formation is not currently considered as one of the important reservoirs in Saudi Arabia but there are many valuable formations underlying Jilh which cannot be utilized unless drilling through Jilh. The gas wells of Jauf, Unayzah, and Khuff formations are an example of producing wells that have to be drilled through the Jilh formation. Unfortunately, some of these wells have encountered high pore pressure while drilling through Jilh which result in non-productive time and sometimes lose of human lives or equipments. So, predicting accurate Jilh formation properties such as pore pressure and porosity (before drilling) is very important.

Drillers have encountered a very high pore pressure of salt-water, oil and/or gas flows in many wells that penetrate through Jilh. Usually, the drillers use a very high mud weight to control the high pressure flow. This high mud weights results in a very low rate of penetration through the Jilh formation. If a previous knowledge of high pressure is well known before drilling, casing strings will be used to isolate the high pressure and get better production. Expertise are saying that 80% of the Khuff gas wells could be drilled with a proper casing design if we know the presence of high pressure before drilling. This will result in huge reductions in overall well costs and increasing productivity (Zhou et al., 2010). In the following I am going to show the geological interpretation of Jilh formation and discuss the different prediction results.

6.2 Geological interpretation and reservoir characterization

The 3D seismic data set acquired over the Jilh carbonate reservoir area and covers about 200 km^2. The 3D survey consists of 411 sub-lines and 226 cross-lines with an interval of 25 m in both directions. The depth of the Jilh formation in this area varies between 2956.6 and 3200.4 m (9700 and 10500 feet). Data were processed at 2 msec. The Jilh zone of interest is equal to 100 ms on the seismic time section. The seismic inversion data was done using the industry standard STRATA software. Geopressure data, in terms of occurrence or non-occurrence of abnormally high formation pressures across Jilh carbonate formation, is available from 19 wells: 10 of these wells have over-pressure and 9 wells have normal pressure. The 19 wells have some well logs like sonic, gamma ray, density, and porosity. Porosity logs have been used to train the artificial intelligent

methods to predict seismic porosity cubes. The pressure information was supplied as mud weight values at the top of the Jilh formation.

Figure 6.6 shows the study area which consists of 2D and 3D seismic data, well data, and a horizon or time picking. Jilh horizon (1.3 - 1.48 sec) was picked and displayed as a time structure map in figure 6.6. Jilh horizon is showing an anticline or higher elevation on the east part. The two higher elevations (lower depth or time) on the east and north east of the study area represent the closures which are usually better areas to keep gas, oil, water or any mixture of them. There are 19 wells posted in figure 6.6. The wells are represented by solid and hollow white circles. The solid white circles are wells that encountered higher pressure zones while drilling. The hollow white circles are wells that had normal pressure while drilling. Two 2D lines were extracted from the 3D seismic volume. One of the 2D lines (11.25 km long) is going east to west (W-E) and crossing a normal pressure well (well D). The second 2D line (20.5 km long) is going south to north (S-N) and crossing a high pressure well (well F). Figure 6.7 shows a 3D image of the study area. The 3D image shows the 2D lines (W-E and S-N), two wells (well D and F) and two horizons.

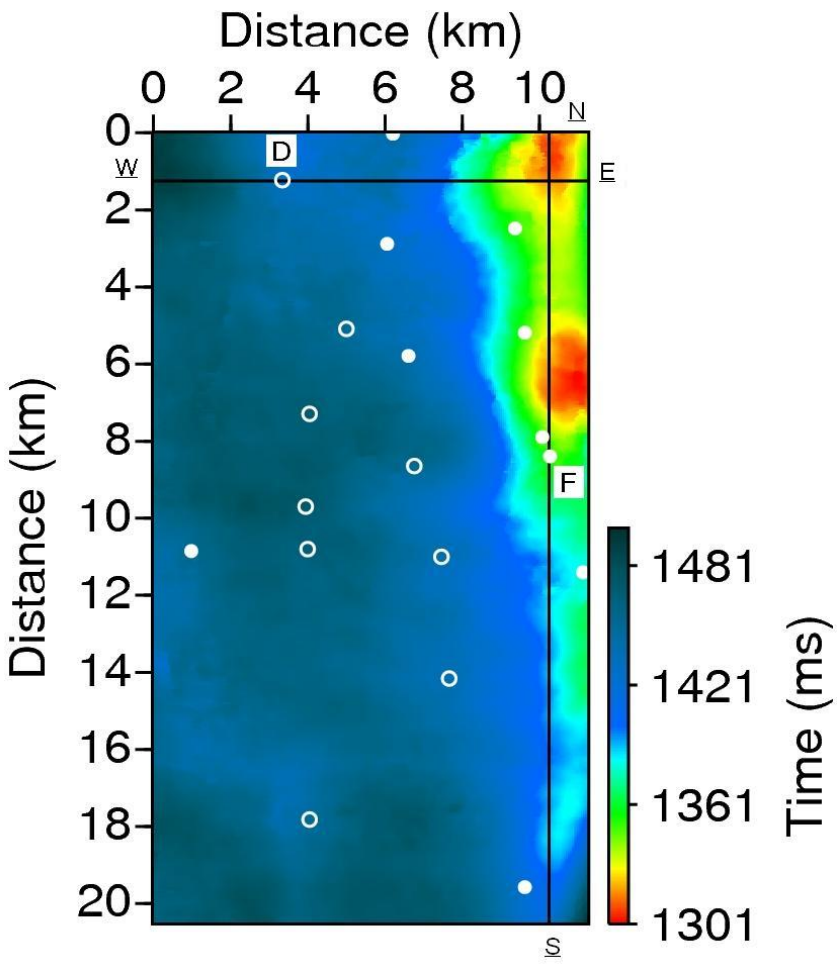


Figure 6.6. The 200 km^2 study area with time structure map of top of Jilh formation. The white circles represent the available wells with solid circles indicating wells with high pressure and hollow circles indicating wells with normal pressure. W-E 2D line passes through the normal pressure well D and S-N 2D line passes through the high pressure well F.

Khuff horizon was mentioned before and it is one of the important gas reservoirs in Saudi Arabia. Top of Khuff formation was picked (1.58-1.73 sec) to get a better picture of the closers and anticlines underlying Jilh formation. Khuff time structure map shows similar events to Jilh time structure map. Figure 6.7 shows Jilh and Khuff formations cross posted with the seismic amplitude sections. I noticed from Figure 6.7 that both Jilh and Khuff formations have high structures on the east side of the study area.

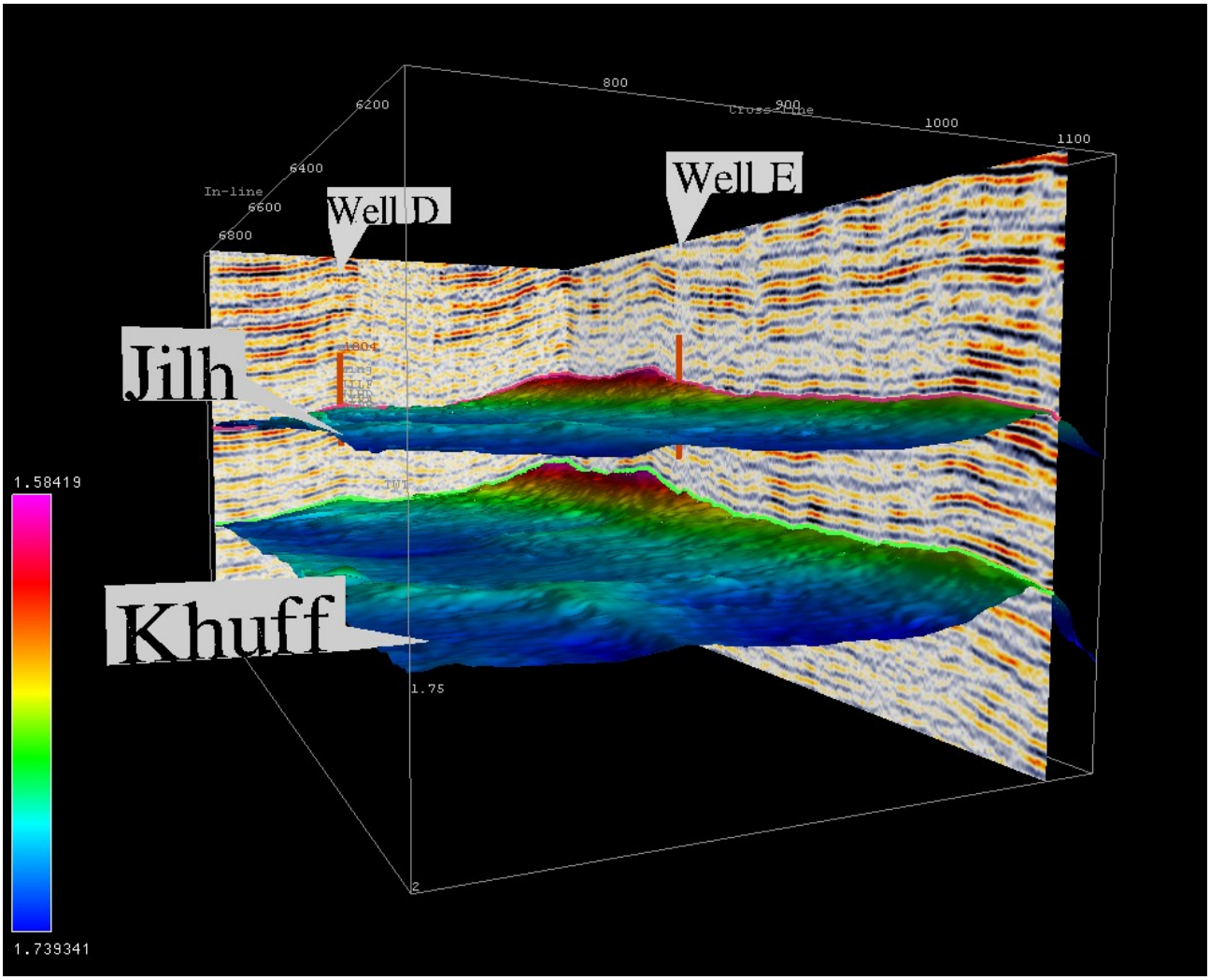


Figure 6.7. 3D image of the study area with two time structure maps picked on top of Jilh and Khuff formations. Well D is posted on the inline going west to east and Well F is posted on the cross line going south to north. The 2D lines are showing seismic amplitude cross sections.

To give a better understanding of the wells, Figure 6.8 shows the synthetic seismogram of well D. In this figure, some real seismic traces are shown near to the location of the well D. The synthetic seismogram is shown in blue traces. Top names or markers are shown near to the area of interest. Minjur, Jilh, Sudair, and Khuff formations are shown across the picture. P-wave and density logs are shown with a red colour to give

a better understanding of the lithology or rock type of each formation. Usually these two log types relate more to the porosity of the subsurface. One will see higher values of velocity and density when going through low porosity zones. From figure 6.8 Jilh formation is having both high and low porosity zones. More porosity can be found in Minjur and Sudair formations.

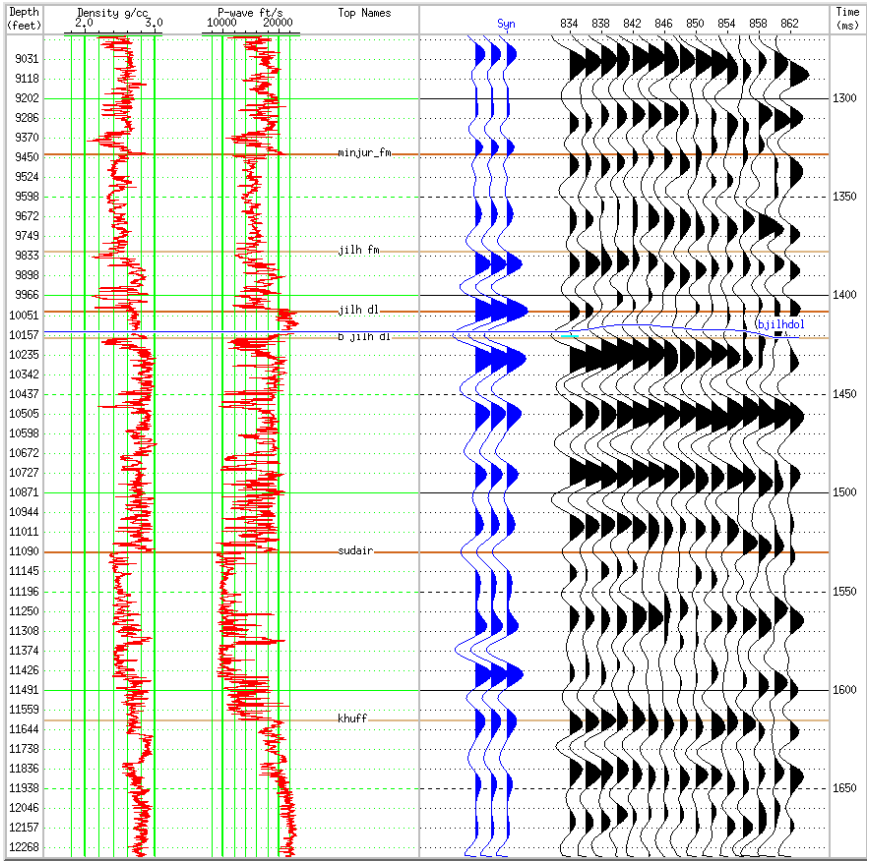
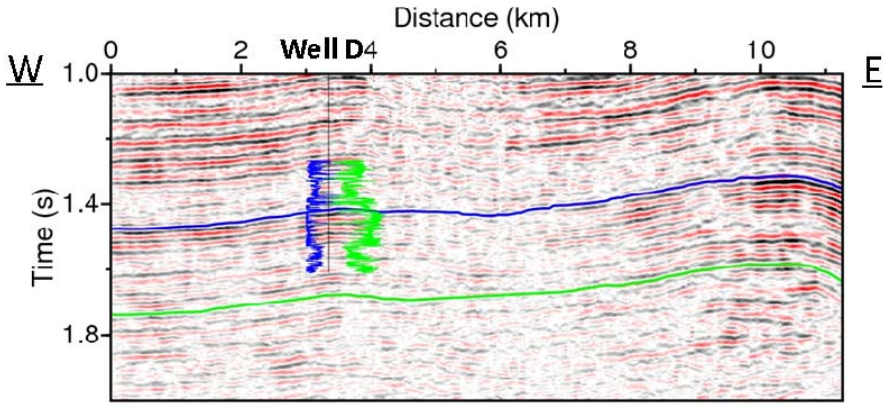


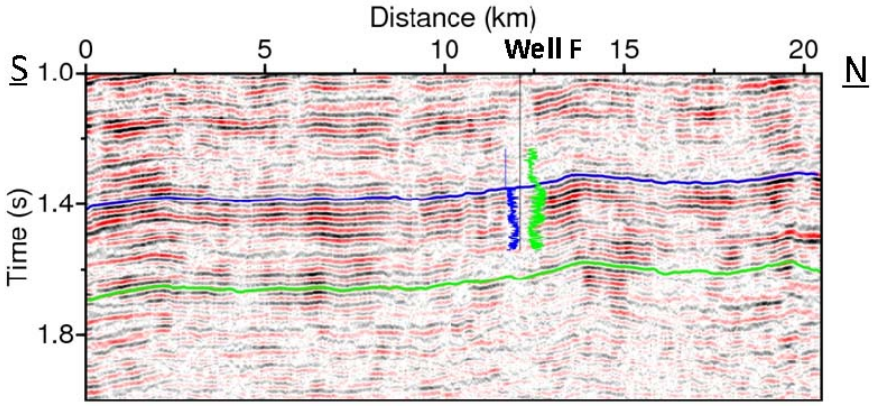
Figure 6.8. Well D synthetic seismogram and nearby seismic traces. Minjur, Jilh, Sudair, and Khuff tops or markers are shown. The blue traces represent the synthetic seismogram. The red curves represent the P-wave and density well logs.

To give a better focus on the selected 2D lines, horizons, and wells, I used Seismic Unix (SU) system to redraw the 2D lines. Figure 6.9a shows the 2D line W-E after posting the well logs and horizons. The blue horizon (upper) is top of Jilh and the green horizon (lower) is top of Khuff. The green well log is a density log and the blue well log is a porosity log posted in well D location. Figure 6.9b shows the 2D line S-N with posting of horizons Jilh and Khuff. The well logs here belong to well F. Figure 6.9 shows

the seismic amplitude data on the background. I noticed that both Jilh and Khuff follow the same event trend.



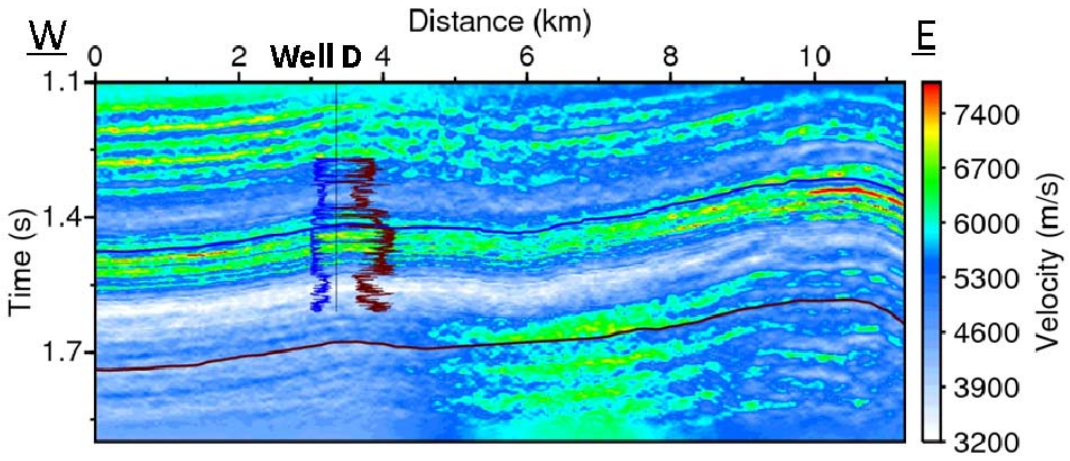
(a)



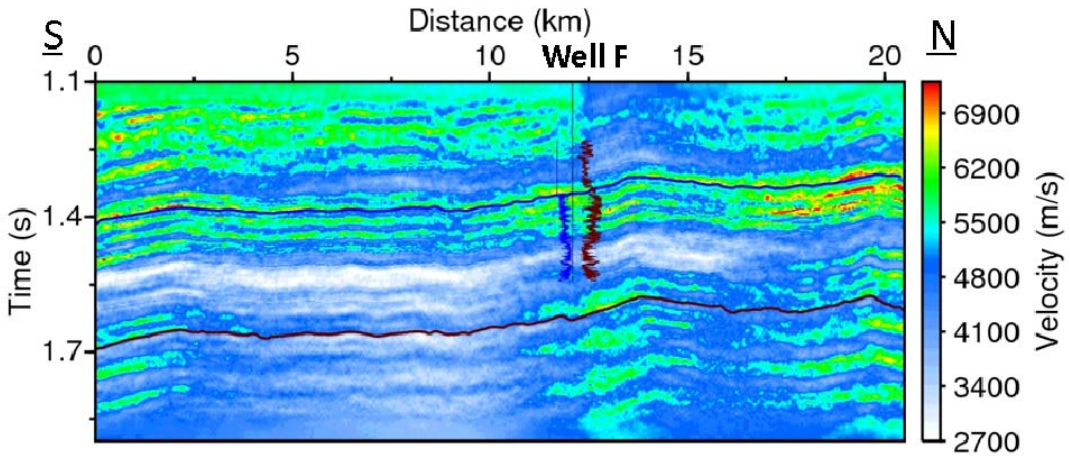
(b)

Figure 6.9. Seismic amplitude 2D lines with posting the wells and horizons (a) W-E 2D line with well D posted and Jilh (blue) and Khuff (green) horizons. (b) S-N 2D line with posting Well F and Jilh and Khuff horizons.

Figure 6.10 shows the seismic impedance (velocity) sections of the 2D lines shown in Figure 6.9. Usually, the contrast in impedance will show better interface between events or layers.



(a)



(b)

Figure 6.10. Seismic impedance (velocity) 2D lines with posting the wells and horizons (a) W-E 2D line with well D posted. (b) S-N 2D line with posting Well F. Jilh (blue) and Khuff (brown) horizons are also posted. The blue well log is porosity and the brown well log is density.

6.2.1 Pore pressure prediction

Pore pressure prediction of Jilh formation is very important to prevent hazardous accidents and get better production from formations underlying Jilh. In this thesis, I tried to predict pore pressure using several methods. Some of these methods gave acceptable qualitative results but some others gave poor results. The main constraint or obstacle in this prediction was the lack of data. Pore pressure information was available only at one sample or location of the well. More information like the mud weight logs should be used to give better prediction.

Supervised and unsupervised neural network methods were used here to find the non-linear relationship between the pore pressure and seismic data. Using neural networks methods I tested three approaches. The first approach uses self-organizing map (SOM), an unsupervised neural network method, to classify seismic attributes based on the data characteristics and no prior knowledge of well information. The second approach uses multilayer perceptrons method, a supervised neural network, to predict porosity and in turn uses SOM to predict geopressure. The third one also uses SOM method but use both seismic impedance and estimated porosity together to predict pore pressure.

One new method that has never been used to predict reservoir properties was also used to solve this pore pressure prediction problem. The group method of data handling (GMDH) is a nonlinear regression method for establishing relations between seismic attributes and reservoir properties. It is a multi-layered structure in which the link between two layers are quadratic polynomial functions, and the final network is a high-order polynomial function linking seismic attributes and the reservoir parameters. I apply the GMDH algorithm to establish the relationship between seismic attributes and pore pressure, and then extrapolate the pressure throughout the entire area.

The available pore pressure data was only indication of the presence of pressure on top of Jilh formation. This constrained the analysis to use only the horizon slice on top of Jilh formation. Figure 6.11a shows the input horizon slice that was used during the training. The unsupervised neural network method (SOM) was used to group and cluster the input seismic impedance data. Figure 6.11b shows the final result of the SOM prediction. SOM grouped the impedance data into 4 clusters. Most of the high pressure wells fall under the red cluster and most of the normal pressure wells fall under the blue

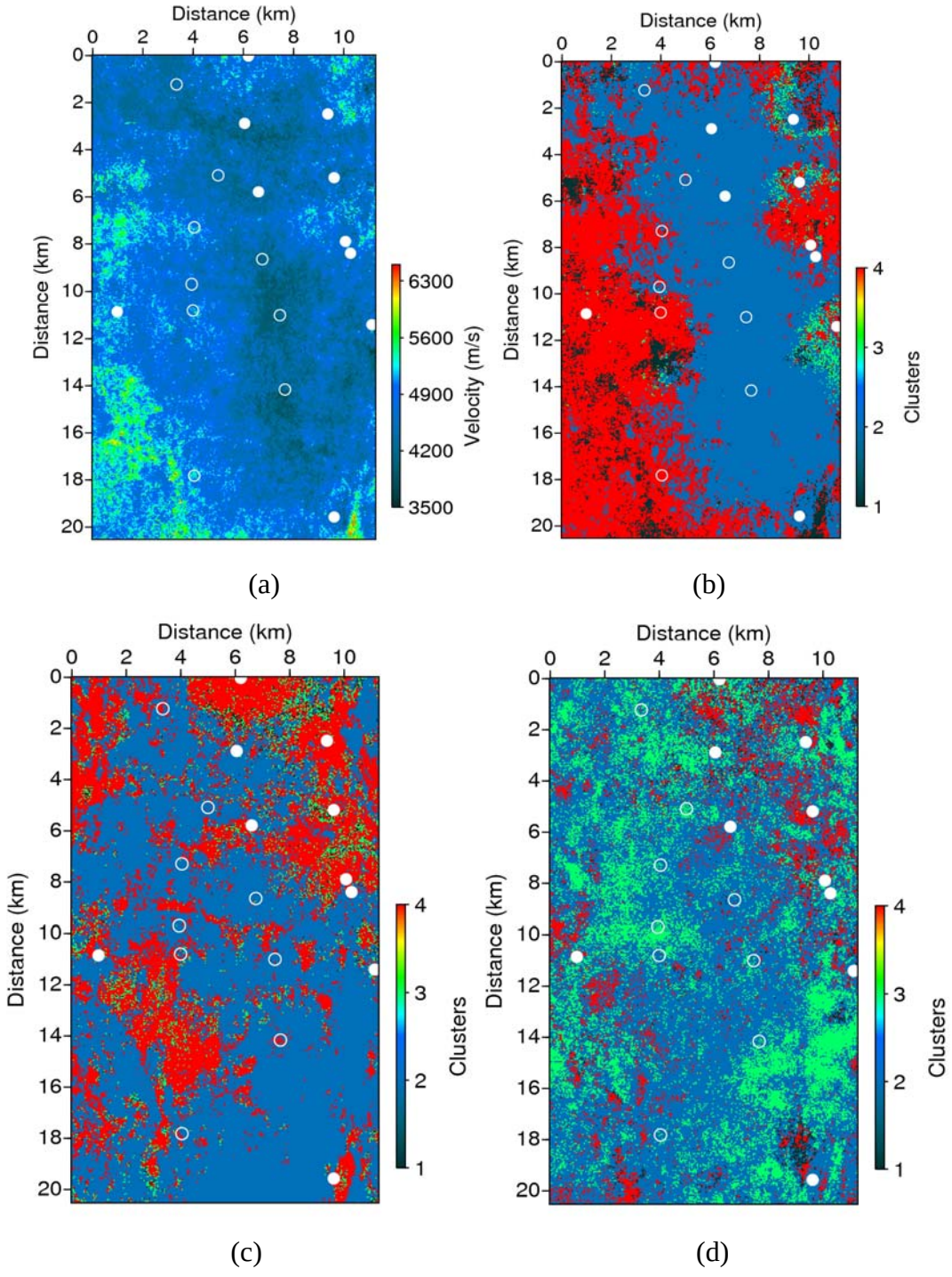
cluster. This was a direct segmentation method that used no prior information about the presence of pressure but only the velocity information from the impedance volume. SOM was also applied using the porosity seismic volume as an input and the result is shown in Figure 6.11c. This time the red area also has more higher pressure wells than the blue area which suppose to have more normal pressure wells. I also clustered both the impedance and porosity volumes and obtained the result in Figure 6.11d. The results are more precise now and the red cluster is getting more involved with the high pressure wells.

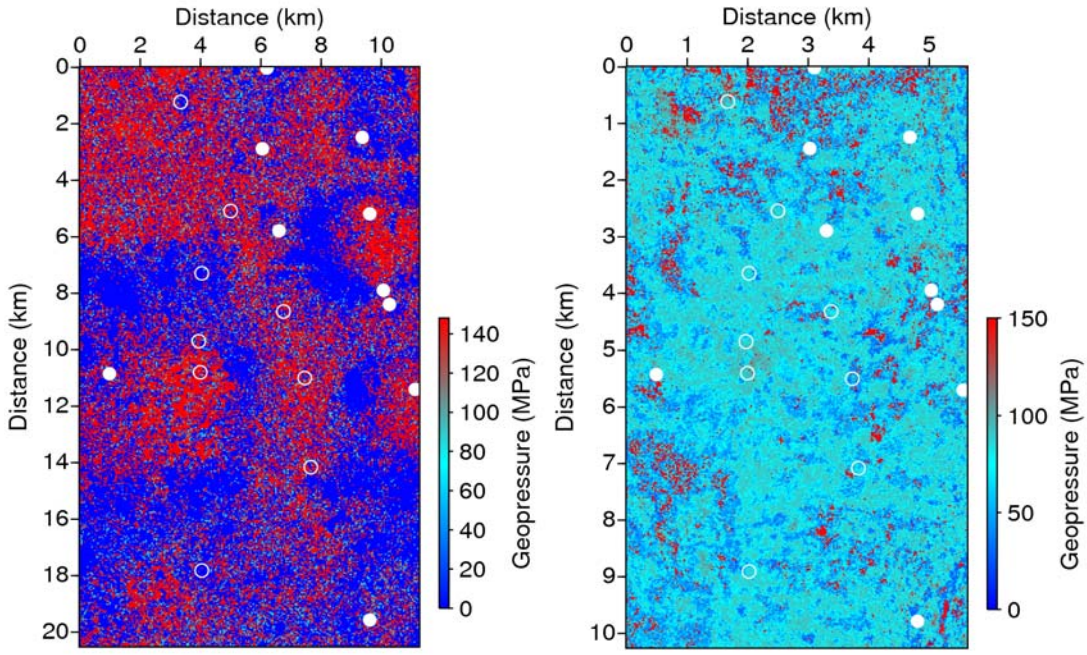
On the other hand, the supervised neural networks multi-layer perceptron (MLP) method and the GMDH method use real data to do the training and learning. The real values of high or low pressure were used to give more accurate pore pressure prediction. MLP used the pressure values and the impedance seismic data to do the training. After applying the MLP method to the impedance data, I obtained the result in Figure 6.11e. The result show that high pressure and normal pressure wells have the same pressure values which is not acceptable. GMDH method was trained using 10 seismic attributes. This was done to find any good relationship between the seismic attributes and the pressure data. Figure 6.11f shows the final GMDH pore pressure prediction result.

From all the results shown in Figure 6.11, I came to some conclusions.

- SOM segmentation of the impedance (Figure 6.11b) and porosity (Figure 6.11c) gave a general clustering of the study area but the clusters were huge and wide which is not a good result scientifically.
- SOM segmentation of both impedance and porosity (Figure 6.11d) gave a better high pressure indications.
- Supervised MLP method did not give a good result in Figure 6.11e. MLP could not learn from very few available pressure data.
- GMDH trained on the pressure data and 10 attributes gave more precise pore pressure predictions (Figure 6.11f).
- GMDH result in Figure 6.11f is superior to the result in Figure 6.11d because the network was trained using some real pressure values.

- The final pore pressure prediction results are still qualitative because of the lack of pore pressure well log data.





(e) (f)

Figure 6.11. (a) A horizon slice of top of Jilh formation extracted from the impedance seismic data. (b) Cluster of the impedance data using the SOM method. (c) Clustering of the porosity data using the SOM method. (d) Multiple clustering of impedance and porosity using the SOM method. (e) Prediction of pore pressure using the supervised MLP method. (f) The prediction of pore pressure using the GMDH method. The GMDH method gave better results than other methods but it is still a very qualitative result due to lack of well log data.

6.2.2 Porosity prediction

Porosity prediction of the Jilh carbonate formation was obtained using supervised artificial intelligent methods. MLP neural network and GMDH methods were used to predict the porosity. In the supervised methods (MLP or GMDH) the input is a part of the trace covering the formation of interest. Porosity logs are used for desired outputs. The training start by letting the input seismic attributes data learn from the real porosity log. This process should teach the supervised method to find the non-linear relationship between the porosity logs and the input seismic data. The process will continue the learning steps without overtraining or overfitting the learning system. The goal here is not to memorize the input porosity logs but to generalize learning. After completing the training step, the supervised system is ready to be tested with other areas on the field. This testing step is called the application step or stage. After applying the trained supervised method onto the seismic impedance cube we obtained a porosity volume in the area.

At the beginning, I used a simple MLP method to get a quick porosity horizon slice. The MLP here was trained using porosity values at the horizon slice location only. Figure 6.12 shows the final horizon slice of the porosity result. This method was quick but gave non acceptable results because I was performing the training with fewer porosity values.

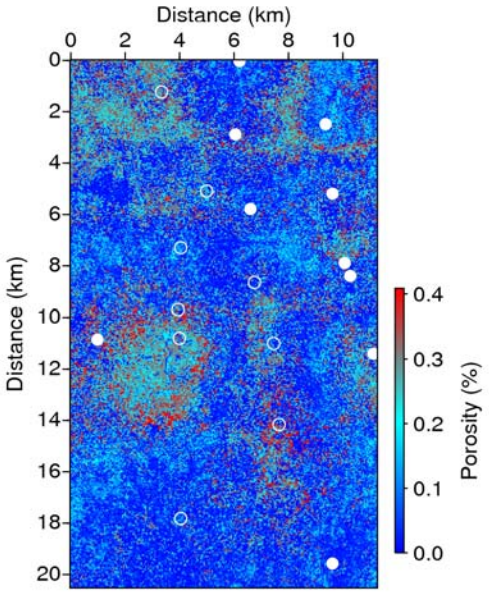


Figure 6.12. Porosity prediction using MLP method. The input data was seismic impedance and porosity values at the top of Jilh formation.

The proper way of doing the supervised artificial intelligent method is to use the full porosity log (or part of it) to predict porosity. Seven wells were used to train both the MLP and GMDH methods. These seven well pass through Jilh formation which is our carbonate area of interest. Figure 6.13 shows the seven porosity logs that were used during training. In this figure, the final porosity prediction is shown. The black curves in Figure 6.13 indicate the original porosity well logs and the red curves show the porosity predictions. The neural network MLP results are shown in dotted red curves and GMDH predictions are shown in solid red curves. Both the MLP and GMDH gave a good porosity prediction.

Since these seven wells were used during the training, I extracted the prediction of three other wells that were never used during training. Figure 6.14 shows the other three wells with original well logs (black curves) and the final porosity prediction (red curves). Still the prediction is quite good and the red curves follow the trend of the original porosity logs.

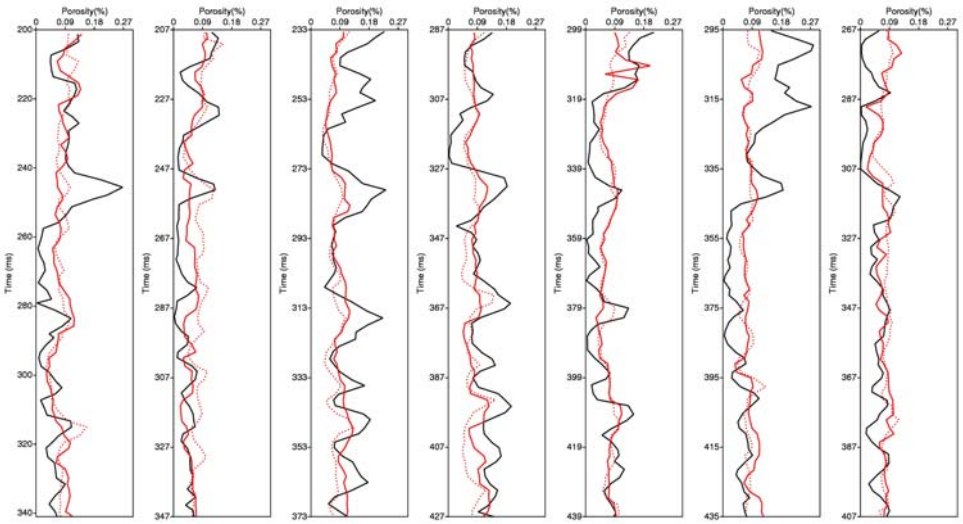


Figure 6.13. Porosity prediction using the supervised methods. Seven porosity logs (black curves) were used during the training. Red solid curves are the porosity prediction of GMDH method. Red dotted curves are the porosity prediction of the neural network MLP method.

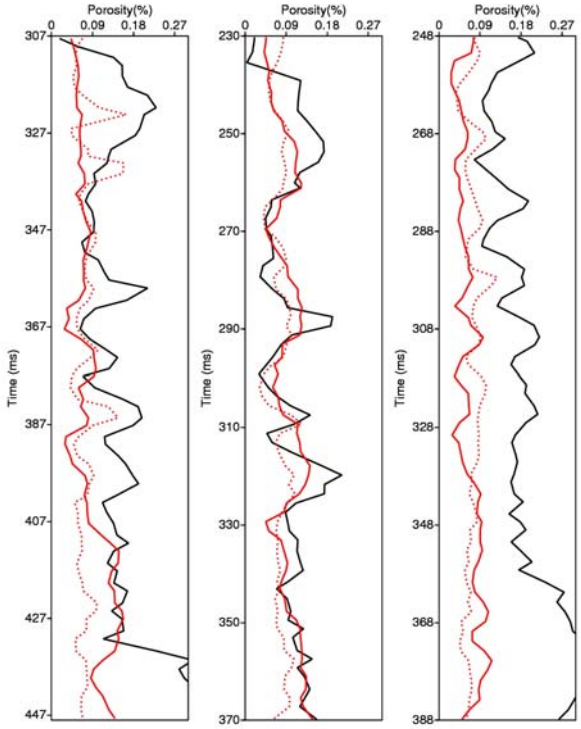


Figure 6.14. Porosity prediction using the supervised methods. Three wells that were used during training. Black curves are the original porosity logs. Red solid curves are the

porosity prediction of GMDH method. Red dotted curves are the porosity prediction of the neural network MLP method.

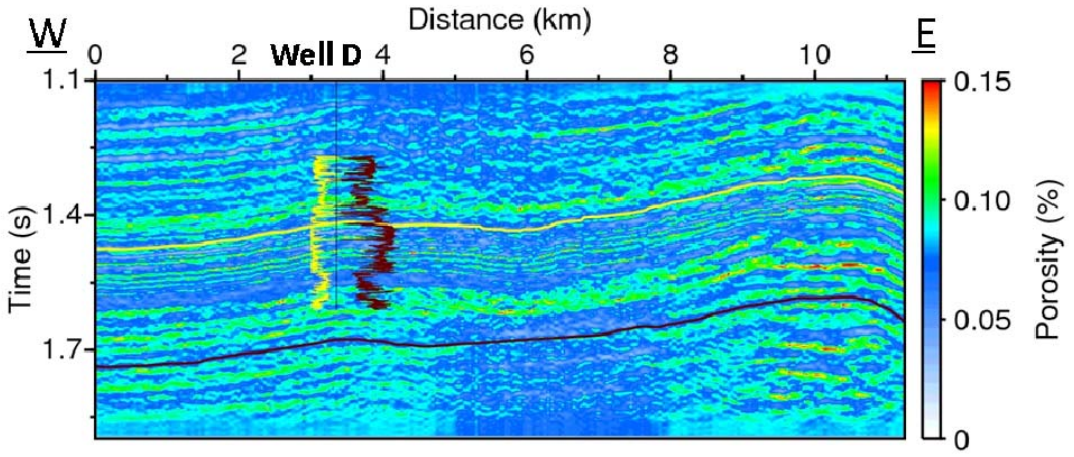
I extracted two 2D lines from the final porosity volumes. The same 2D lines that I used from Figure 6.6. The first 2D line is W-E and displaying well D. The second 2D line is S-N and displaying well F. Top of Jilh formation is the horizon with yellow colour while top of Khuff formation is the horizon with brown colour. The porosity logs are displayed with yellow well log while the density logs are displayed with brown well log. Figure 6.15a shows the porosity results of the MLP method for the W-E line. Figure 6.15b shows the porosity results of the MLP method for the S-N line. The W-E 2D line shows an indication of higher porosity next to the top of Jilh formation but the porosity is getting less when we get deeper in Jilh. Layers just above Khuff formation are have higher porosities in the W-E 2D line. The S-N 2D line that goes across the higher structural areas in the east side of the study area shows higher porosity values in Figure 6.15b. Figures 6.16a and 6.16b show the porosity prediction results of the GMDH method. The W-E 2D line is shown in Figure 6.16a and the S-N 2D line is shown in Figure 6.16b. The same comments that I mentioned regarding the MLP porosity applied for the GMDH porosity results. The Results look the same but GMDH method showed higher porosity values than the MLP porosity.

On the other hand, I extracted the horizon slices passing on top of Jilh formation for both porosity results. The horizon slice should show the entire porosity covering the study area. Figure 6.17 shows the extracted porosity horizon slices. The GMDH result in Figure 6.17b shows higher or brighter porosity values than the MLP result in Figure 6.17a. Both results show a good general indication of the porosity for the study area.

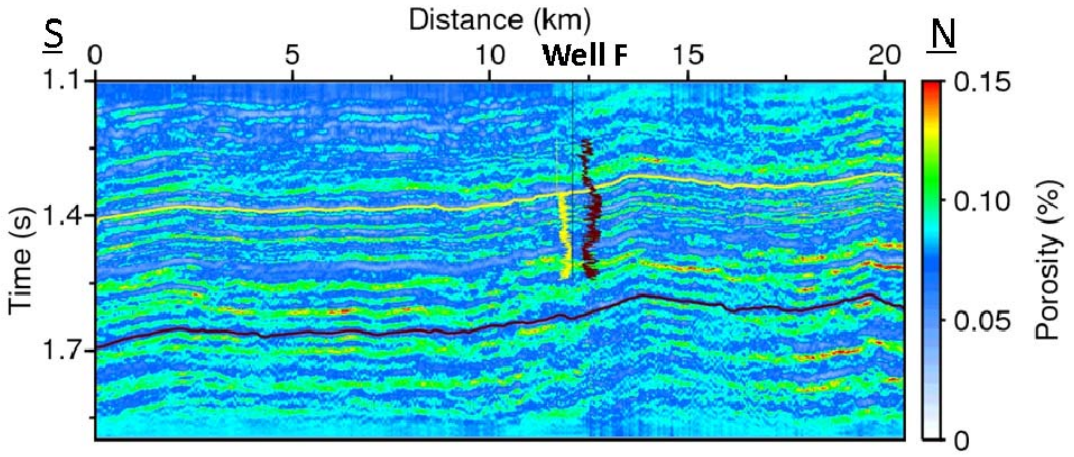
From all the results shown in Figures 6.15, 6.16, and 6.17 I came to some conclusions.

- Both MLP and GMDH porosity results show good prediction when displayed in the 2D lines.
- The porosity predictions were not very far from the porosity well logs displayed on the 2D lines.

- The main closure or structure on the east side of the study area show higher porosity values inside the Jilh formation.
- The well in the west side of the study area (well D) shows more porosity next to the top of Khuff formation.
- GMDH results were consistent with MLP results when displayed on the cross sections.
- The horizon slices extracted on top of Jilh formation showed that GMDH porosity have higher values than the MLP porosity results.

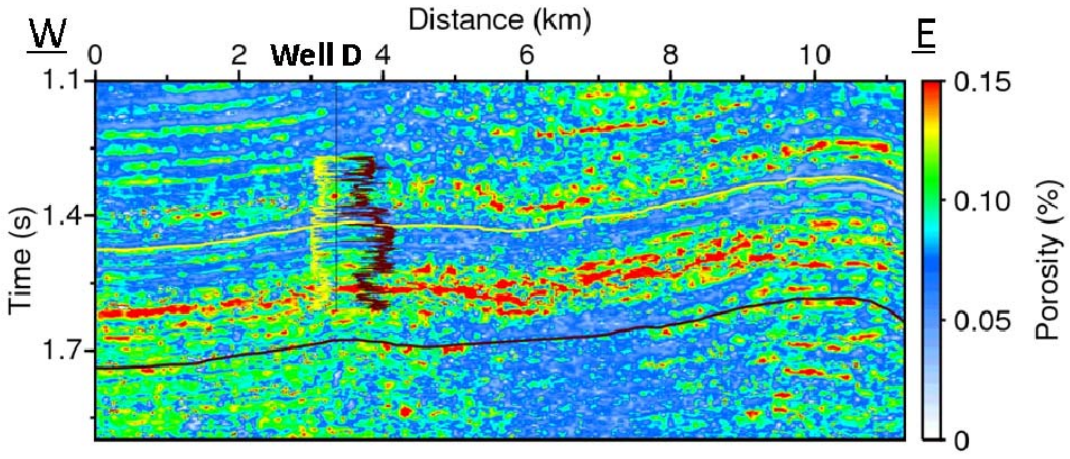


(a)

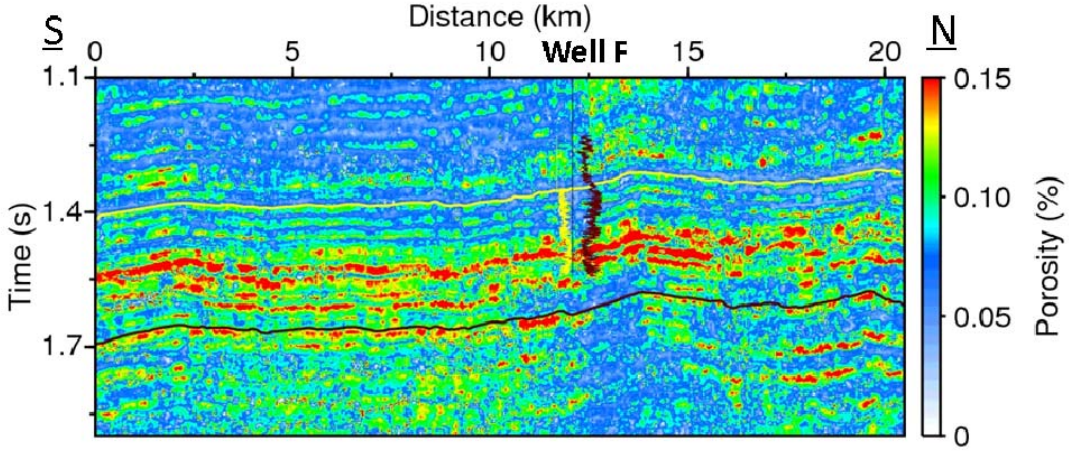


(b)

Figure 6.15. Porosity prediction using the supervised neural network method. (a) The porosity result of the W-E 2D line with well D displayed. (b) The porosity result of the S-N 2D line with well F displayed. Jilh horizon is yellow and Khuff horizon is brown. The yellow logs are porosity logs and the brown logs are density logs.

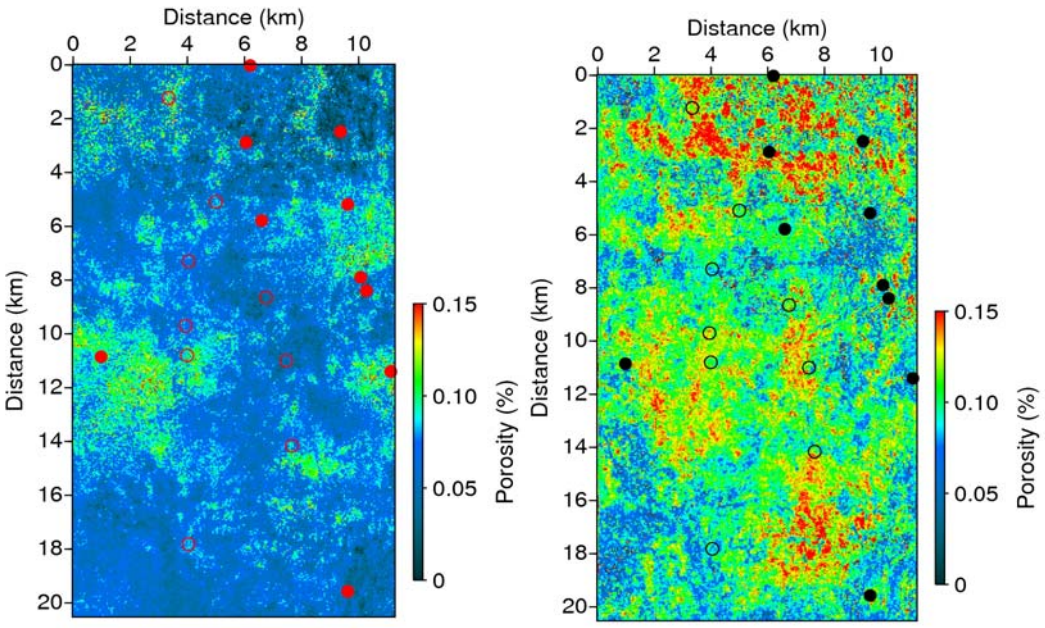


(a)



(b)

Figure 6.16. Porosity prediction using the GMDH method. (a) The porosity result of the W-E 2D line with well D displayed. (b) The porosity result of the S-N 2D line with well F displayed. Jilh horizon is yellow and Khuff horizon is brown. The yellow logs are porosity logs and the brown logs are density logs.



(a) (b)

Figure 6.17. Jilh Horizon slices of the predicted porosity using the supervised methods. (a) Horizon slice extracted from the porosity volume that was predicted using MLP. (b) Horizon slice extracted from the porosity volume that was predicted using GMDH. The solid and hollow circles are the available well data.

6.3 Summary

Throughout this thesis, artificial intelligent methods were used to predict the reservoir properties using seismic data. Supervised and unsupervised neural networks and group methods for data handling were used to do the prediction. The area of interest is called Jilh formation which is a carbonate formation from Uthmaniyah field. Uthmaniyah is one of four parts of Ghawar which is the biggest oil field on the world. Prediction of Jilh porosity and high pore pressure is significant not only to Jilh but also to the producing gas and oil formations underlying Jilh. GMDH did a better pore pressure prediction using the available well data. Pore pressure predictions might have better results if more complete data is available. GMDH and MLP predict a good porosity cubes. Both methods gave similar porosity cross sections but different horizon slices. GMDH gave brighter porosity or higher values than the MLP.

Chapter 7

Overall conclusions and future work

The aim of this project is to predict reservoir parameters (porosity and pore pressure) from carbonate reservoirs, using seismic data if possible. From the literature I have read, I see that no one yet (based on my knowledge) has achieved promising results by using the conventional ways of predicting pore pressure from seismic data. The only methods I found to have attained some successful experiments are those using modelling 4D seismic and the neural network clustering techniques or statistical techniques. The 4D seismic method was not one of my target methods because of the need to reacquire the seismic data at two different times. Artificial intelligent methods, such as neural network techniques and the group method of data handling produced meaningful results and were used as the main tools for this study. The artificial intelligent methods were used here because of the power of these methods to deal with complex data such as the carbonate reservoirs. Also, the artificial intelligent methods are using more input data such as seismic attributes which could show more physical meaning about a specific reservoir property such as pressure or porosity.

Three seismic datasets were used in this research. The first dataset was generated using cross-hole seismic data waveform tomography. The source points were in one vertical borehole and the receivers in the other. The data contain igneous rocks on the bottom and layers of mudstone and sandstone. The second dataset is a demo 3D seismic dataset that was acquired from the Dutch sector of the North Sea. This dataset has large deposits of deltaic package. The overall porosity is between 20% and 30%. The general lithology of the data is sand and shale with some carbonate streaks. Sonic, density, and gamma ray logs are available from 4 wells in the area. The third dataset is a 3D seismic cube from the Saudi Aramco Oil Company. The seismic data were acquired from the

largest oil field on the world, which is the Ghawar field. These data have an overpressured carbonate formation, which is my area of interest. With dolomite and inter-bedded thin shale bodies, the thickness of the formation is about 900 feet (274.3m). 19 wells are available with this dataset and each well has sonic, density, and porosity logs. These wells also have an indication of overpressure on the top of my target carbonate reservoir.

Not only the seismic amplitude but also other seismic attributes were used in this research. The seismic attributes are helpful in interpreting the geological events, and show better visualization of faults and folds. Quantities derived from the complex-trace analysis are examples of seismic attributes. These attributes include instantaneous amplitude, instantaneous phase, instantaneous frequency, normalized phase, thin-bed and instantaneous Q factor. Also, geometric attributes, such as coherence and curvature, were used in this research. These attributes relate more to reservoir properties. The artificial intelligent methods in this research made full use of these attributes. Some seismic attributes were selected by the statistical methods to predict better and faster reservoir properties.

This thesis showed that the carbonate reservoirs are more complex than the clastic reservoirs. The velocity methods are using empirical equations that could be applied on the clastic reservoirs with some assumptions know beforehand. The complex characteristics of the carbonate reservoirs derived me to look for better methods such as the artificial intelligent methods. ANN showed a powerful ability to predict the reservoir properties but GMDH was more automated and gave better results.

7.2 Summary of each chapter

(1) High resolution velocities

Pore pressure was predicted using the seismic velocity methods. Most of the velocity methods predictions done in the industry were applied in the clastic reservoirs. Two of the most famous empirical velocity methods were those of Eaton (1975) and Bowers (1995). These empirical methods are widely used in the industry for predicting pore pressure in clastic reservoirs (shale and sandstone). While Eaton was transforming the velocity directly to pore pressure, Bowers was estimating the effective stress from velocities to predict the pore pressure. This means that one should have a high resolution velocity inversion in order to obtain better pore pressure prediction. A direct comparison between the pore pressure predictions of two seismic velocity inversion methods was done to show the importance of using high resolution velocities. Seismic waveform tomography proved that the resolution of the prediction depends on the quality of the input velocity. The travel-time inversion method showed a good prediction but, in terms of quality, was not as good as waveform tomography. Pore pressure was also predicted using the nearby wells. Sonic logs with more data samples and resolution showed a more detailed prediction. The prediction from the wells was compared with the seismic inversion data prediction. Both predictions were consistent with each other. Real carbonate seismic data were used with the Bowers equation. This experiment meant to show that the Bowers equation is a good prediction when used with clastic data only. The different structure of the carbonate data prevented it from being easily estimated using simple prediction equations. Three lines were extracted from the carbonate reservoirs and used as an input to the Bowers equation. The three lines have a total of three wells. Two overpressured wells were in two lines. The third line has a normal pressure well. As was expected, the results were not very reliable and cannot be used as a direct pore pressure prediction. This taught me that I should look for better carbonate reservoir prediction methods, such as the artificial intelligent methods.

(2) Self organizing map (SOM)

Neural networks methods were investigated here to solve the problem of predicting pore pressure in carbonate reservoirs. Neural networks methods were used to relate quantities that have non-linear relations for the purpose of predicting the reservoir parameters. The artificial neural networks were used here to find a non-linear relationship between geopressure and seismic attributes. The SOM is one of the unsupervised neural networks methods that classify the data, based on competitive learning. The method uses the internal structure of the data and does not need any additional information such as well logs. Qualitative information can be quickly produced using the SOM clustering method. The SOM simplifies and separates the output data into smaller groups. Quantitative descriptions of the clustered data can be done by clustering the results of the SOM. This can be done by using clustering methods like k-means and FCM. The K-means and FCM clustering algorithms can be applied after training the SOM. For the seismic carbonate dataset I have, both clustering methods gave the same final result.

My SOM code was tested and validated by comparing the clustering results with an industrial neural network tool box. After that, real carbonate seismic attributes data were used to predict the pore pressure. The SOM did a good job in clustering all the input seismic attributes. Among these attributes, only a few gave a good response to geopressure. Geopressure was tested by using an indication of geopressure from 19 wells. 10 of these wells have geopressure indications, while the remaining 9 wells have normal pressure indications. The best final prediction model should pass 3 quality tests. The quality tests are visualization, logical, and mathematical tests. The visualization quality test is applied to check if the background colours are consistent with the pressure indication at the well locations. The logical quality check discards any background colour covering the entire area with the same colour. The final statistical or mathematical quality check counts the occurrence of overpressured and normal pressured wells at each cluster location. The final decision was the selection of the two best seismic attributes. These seismic attributes are coherence and seismic impedance.

(3) Multilayer perceptrons (MLP)

Supervised methods are another branch of neural networks methods. While the unsupervised methods were learning directly from the data, the supervised methods learn by introducing examples to the neural networks. The MLP was used here to find a non-linear relation between seismic attributes (impedance) and pore pressure or porosity. The MLP method was coded, first using a 2D slice trained operator to learn quickly the geopressure at the top of the reservoir, and predict new values in the missing areas. The results show some consistency with the results of the unsupervised neural networks method. For the purpose of generalization, the 2D slice trained operator was upgraded to 3D trained operator. The new supervised neural networks code was validated by comparing the results with an industrial neural network tool box. The supervised neural network method was applied to real seismic carbonate data to predict porosity. The predicted porosity was then used with an unsupervised neural network method (SOM) for the purpose of predicting geopressure. The results were acceptable. Finally, a multi-attribute SOM method was developed to predict pore pressure using seismic impedance and porosity as an input. The final results were not as good as predicting pore pressure using seismic impedance or porosity alone.

(4) Group method of data handling (GMDH)

Quadratic polynomial functions were used by GMDH to simplify the complex systems into smaller subsystems. The artificial intelligent systems that I had used previously entailed either self organizing the input data or learning from examples. The GMDH has the ability to self organize and learn from examples during the training. This saves computation time and programming efforts. The GMDH uses multilayers and nodes, which is the same method as the supervised neural networks. The data move forward during training from one layer to the next. GMDH can automatically determine the best network structure, number of nodes, important inputs, and coefficients without overtraining the data. The learning process of the GMDH network will stop when the system no longer predicts good quality results. The original classical GMDH was enhanced to have better stopping criteria, such as the predicted squared error. Also, the

enhanced GMDH uses only one dataset for training and testing, while the classical GMDH needs two separate datasets.

The GMDH algorithm had not been used before to predict reservoir properties. Among the many uses of GMDH, I mention here the prediction of tool life in drilling, temperature distributions, and many other predictions. The GMDH code was validated by comparing the results with the synthetic data used by the original classical GMDH algorithm. The new enhanced GMDH code produced better results. GMDH was also used to predict pore pressure from a carbonate seismic data using 19 wells with geopressure indications. The results were quick but not very accurate because of missing necessary data, such as mud weight logs. Among the 14 seismic attributes that were used during the training, GMDH selected only 5 attributes to represent pore pressure prediction. The GMDH algorithm was used here in order to find a non-linear relation between seismic attributes and pore pressure or porosity. GMDH was trained using 7 wells that contain porosity logs. The GMDH was trained using 13 seismic attributes and only 6 attributes were chosen to represent porosity. The final porosity prediction was successful when compared with original porosity well logs.

7.3 Future Research

From my point of view, I see that work could be done in the future in the following areas:

1. The pore pressure prediction that was done on chapters 3, 4, and 5 using the artificial intelligent methods needs to be done again carefully, with more data from the wells. Mud weight logs need to be used and not only indications of overpressure on top of the target formation.
2. More seismic attributes could be used and tested to predict the carbonate reservoir properties.
3. More experiments should be done using the GMDH algorithm to predict better pore pressure.
4. GMDH should be utilized to predict not only pore pressure and porosity but other reservoir properties, such as permeability.

5. The coding and application of the ANN methods was not very easy. New ways to select the networks parameters are needed. The parameters such as the learning rate, radius, random values, and minimum and maximum errors were taking long programming time to come up with the best combinations. Better selection criterion is needed.
6. GMDH method requires more attributes to give better predictions. New attributes such as the seismic data acquisition parameters and survey parameters might help the GMDH predict better reservoir properties.
7. More well logs can be utilized such as the resistivity log. The resistivity log can be used to predict resistivity volumes or as an extra attribute to predict other attributes.
8. The artificial intelligent methods could be applied on part of the data first to help in the parameters selections on the training process.
9. The input seismic volume could be divided into parts if the data is not good enough or if noise presents on some parts of the data.

Having said this, I feel that this dissertation advances research in the area of carbonate reservoir parameter prediction, and hope that the ideas presented here will be of use to my colleagues and to future geophysical students.

REFERENCES

Abdel-Aal, R. E., 2003, Direct estimation of noisy sinusoids using abductive networks: Engineering Applications of Artificial Intelligence, **16,** 615-627.

AlMustafa, H. and AlZahrani, S., 2001, Geopressure detection using neural classification of seismic attributes in Base Jilh Dolomite in Central Uthmaniyah, Saudi Arabia: SEG Technical Program Expanded Abstracts, **20,** 1540-1543.

Aminzadeh, F. and Brouwer, F., 2006, Integrating neural networks and fuzzy logic for improved reservoir property prediction and prospect ranking: SEG Technical Program Expanded Abstracts, **25,** 1752-1756.

Aminzadeh, F. and Groot, P., 2006, Neural Networks and Other Soft Computing Techniques with applications in the oil industry: EAGE Publications bv.

Anselmetti, F. S. and Eberli, G. P., 1993, Controls on sonic velocity in carbonates: Pure and Applied Geophysics, **141,** 287-323.

Arkell, W. J., Bramkamp, R. A., and Steineke, M., 1952, Jurassic Ammonites from Jebel Tuwaiq, Central Arabia: Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences, **236,** 241-314.

Banik, N. C., Wool, G., Schultz, G., den Boer, L., Mao, W., Huggard, J. D., Nibbelink, K., and Taylor, N., 2003, Regional and high-resolution 3D pore-pressure prediction in deep-water offshore West Africa: SEG Technical Program Expanded Abstracts, **22,** 1386-1389.

Barron, R. L., Mucciardi, A. N., Cook, F. J., Craig, J. N., and Barron, A. R., 1984, Adaptive learning networks: Development and application in the United States of algorithms related to GMDH *in Self-Organizing Methods in Modeling: GMDH Type Algorithms*: Farlow, S. J., Ed., Marcel Dekker, New York.

Bowers, G. L., 1995, Pore Pressure Estimation From Velocity Data: Accounting for Overpressure Mechanisms Besides Undercompaction: SPE Drilling & Completion, **10,** 89-95.

Bruce, B., 2002, An introduction to this special section: Pore pressure: The Leading Edge, **21,** 169-177.

Bruce, B., Borel, R., and Bowers, G., 2002, Well planning for SWF and overpressures at the Kestrel well: The Leading Edge, **21,** 669-671.

Bruce, B. and Bowers, G., 2002, Pore pressure terminology: The Leading Edge, **21,** 170-173.

Campbell, J. L. E. and Johnson, K. E., 1993, Abductive networks: generalization, pattern recognition, and prediction of chemical behavior: Can. J. Chem., **71,** 1800-1804.

Chilingar, G., Serebryankov, V., and Robertson, J., 2002, Origin and Prediction of Abnormal Formation Pressures: Elsevier.

Chopra, S. and Huffman, A. R., 2006, Velocity determination for pore-pressure prediction: CSEG Recorder, 28-46.

Draper, N. R. and Smith, H., 1966, Applied Regression Analysis: Wiley, New York.

Dunn, J. C., 1973, A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters: Journal of Cybernetics, **3,** 32-57.

Dutta, N. C., 2002a, Deepwater geohazard prediction using prestack inversion of large offset P-wave data and rock model: The Leading Edge, **21,** 193-198.

-----2002b, Geopressure prediction using seismic data: Current status and the road ahead: Geophysics, **67,** 2012-2041.

Dutta, N. C. and Khazanehdari, J., 2006, Estimation of formation fluid pressure using high-resolution velocity from inversion of seismic data and a rock physics model based on compaction and burial diagenesis of shales: The Leading Edge, **25,** 1528-1539.

Eaton, B. A., 1972, Graphical method predicts geopressure worldwide: World oil, **182,** 51-56.

Eberli, G. P., Baechle, G. T., Anselmetti, F. S., and Incze, M. L., 2003, Factors controlling elastic properties in carbonate sediments and rocks: The Leading Edge, **22,** 654-660.

Farlow, S. J., 1984b, The GMDH Algorithm *in Self-Organizing Methods in Modeling: GMDH Type Algorithms*: Farlow, S. J., Ed., Marcel Dekker, New York.

-----1984a, *Self-Organizing Methods in Modeling: GMDH Type Algorithms*: Marcel Dekker, New York.

Fertl, W. H., 1976, Abnormal formation pressure: Elsevier.

Fulcher, G. E. and Brown, D. E., 1994, A polynomial network for predicting temperature distributions: Neural Networks, IEEE Transactions on, **5,** 372-379.

Hagan, M., Demuth, H., and Beale, M., 1996, Neural Network Design: PWS Publishing Company, Boston MA.

Hampson, D. P., Schuelke, J. S., and Quirein, J. A., 2001, Use of multiattribute transforms to predict log properties from seismic data: Geophysics, **66,** 220-236.

Haykin, S., 1999, Neural networks: A comprehensive foundation: Prentice Hall.

Hornik, k., Stinchcombe, M., and White, H., 1988, Multilayer feedforward networks are universal approximators: Manuscript Dept. of Economics. University of California, San Diego, June 1988.

Hottman, C. E. and Johnson, R. K., 1965, Estimation of formation pressures from log derived shale properties: J. Pet. Tech., **6,** 717-722.

Huffman, A. R., 2002, The future of pore-pressure prediction using geophysical methods: The Leading Edge, **21,** 199-205.

Iglewicz, B., 1983, Robust scale estimators and confidence intervals for location *in* Hoaglin, D.C., Mosteller, M. and Tukey, J.W., Understanding Robust and Exploratory Data Analysis: Wiley, N.Y.

Ivakhnenko, A. G., 1971, Polynomial Theory of Complex Systems: Systems, Man and Cybernetics, IEEE Transactions on, **1,** 364-378.

Kohonen, T., 1989, Self-Organizing and associative memory: Springer-Verlag.

-----1997, Self-Organizing Maps: Springer-Verlag.

LeCun, Y., Denker, J. S., and Solla, S. A., 1990, *Optimal brain damage in* Advances in Neural Information Processing Systems 2: D.S.Touretzky (ed.), Ed., Morgan Kaufman, Denver, CO.

Lee, B. Y., Liu, H. S., and Tarng, Y. S., 1999, An abductive network for predicting tool life in drilling: Industry Applications, IEEE Transactions on, **35,** 190-195.

Liaw, A. L., 2002, Pore pressure prediction from 3-D seismic data: SEG Technical Program Expanded Abstracts, **21,** 454-456.

Lin, F. C. and Kwan, C. T., 2004, Application of abductive network and FEM to predict an acceptable product on T-shape tube hydroforming process: Computers & Structures, **82,** 1189-1200.

Liu, E., Vlastos, S., Li, X. Y., Main, I. G., and Schoenberg, M., 2004, Modeling seismic wave propagation during fluid injection in a fractured network: Effects of pore fluid pressure on time-lapse seismic signatures: The Leading Edge, **23,** 778-783.

MacQueen J. B., 1967, Some methods for classification and analysis of multivariate observations: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, **1**, 281–297

Mao, W., Fletcher, R., and Deregowski, S., 2000, Automated interval velocity inversion: SEG Technical Program Expanded Abstracts, **19,** 1000-1003.

Montgomery, G. J. and Drake, K. C., 1990, Abductive networks: SPIE, **1294,** 56-64.

-----1991, Abductive reasoning networks: Neurocomputing, **2,** 97-104.

Musgrave, A. W. and Hicks, W. G., 1966, OUTLINING OF SHALE MASSES BY GEOPHYSICAL METHODS: Geophysics, **31,** 711-725.

Pennebaker, E. S., 1968, Seismic data indicate depth and magnitude of abnormal pressure: World oil, **166,** 73-82.

Rao, Y. and Wang, Y., 2005, Crosshole seismic tomography: working solutions to issues in real data travel time inversion: Journal of Geophysics and Engineering, **2,** 139-146.

Rao, Y., Wang, Y., and Morgan, J. V., 2006, Crosshole seismic waveform tomography - II. Resolution analysis: Geophysical Journal International, **166,** 1237-1248.

Saggaf, M. M., Toksoz, M. N., and Marhoon, M. I., 2003, Seismic facies classification and identification by competitive neural networks: Geophysics, **68,** 1984-1999.

Sayers, C. M., Johnson, G. M., and Denyer, G., 2002a, Predrill pore-pressure prediction using seismic data: Geophysics, **67,** 1286-1292.

-----2002b, Seismic pore-pressure prediction using reflection tomography and 4-C seismic data: The Leading Edge, **21,** 188-192.

Schoenberg, M., 2002, Time dependent anisotropy induced by pore pressure variation in fractured rock: J. Seismic Exploration, **11,** 83-105.

Shastri, V., Rabelo, L., Onjeyekwe, E., and Vila, J., 1998, Device-independent Color Correction for Multimedia Applications Using Neural Networks and Abductive Modeling Approaches: Expert Systems, **15,** 110-119.

Smith, S. W., 1997, The Scientist and Engineer's Guide to Digital Signal Processing: California Technical Publishing.

Stork, C., 1992, Reflection tomography in the postmigrated domain: Geophysics, **57,** 680-692.

Swan, H. W., 2001, Velocities from amplitude variations with offset: Geophysics, **66,** 1735-1743.

Taner, M. T., Koehler, F., and Sheriff, R. E., 1979, Complex seismic trace analysis: Geophysics, **44,** 1041-1063.

Terzaghi, K., 1943, Theoretical Soil Mechanics: John Wiley and Sons, Inc., New York.

Tsimel'zon, I. O., 1959, The relationship between local gravity anomalies and the tectonics of oil- and gas-bearing regions: Izv. Vyss. Uchebn. Zaved. , Neft Gas, **8,** 13-16.

Ultsch, A., 1993, Knowledge extraction from self-organizing neural networks *in* Opitz et al., eds., Information and classification: Springer-Verlag.

Vlastos, S., Narteau, C., Liu, E., and Main, I. G., 2003a, Numerical study of scattering attenuation in fractured media: The effects of scalelength on multiple scattering attenuation: SEG Technical Program Expanded Abstracts, **22,** 1612-1615.

Vlastos, S., Schoenberg, M., Maillot, B., Main, I. G., Liu, E., and Li, X. Y., 2003b, Dual simulations of fluid flow and seismic wave propagation in a fractured network: Effects of changes in pore pressure on signature of seismic waves: SEG Technical Program Expanded Abstracts, **22,** 1382-1385.

Wang, B., Pann, K., and Meek, R. A., 1995, Macro velocity model estimation through model-based globally-optimized residual-curvature analysis: SEG Technical Program Expanded Abstracts, **14,** 1084-1087.

Wang, Y. and Rao, Y., 2006, Crosshole seismic waveform tomography - I. Strategy for real data application: Geophysical Journal International, **166,** 1224-1236.

Wieland, A. and Leighton, R., 1987, *Geometric analysis of neural network capabilities*: first IEEE International Conference on Neural Networks, **III,** 385-392.

Woodward, M., Farmer, P., Nichols, D., and Charles, S., 1998, Automated 3D tomographic velocity analysis of residual moveout in prestack depth migrated common image point gathers: SEG Technical Program Expanded Abstracts, **17,** 1218-1221.

Yilmaz, O., 2001, *Seismic data processing*: Soc. Expl. Geophys. Publication.

Zhou, S., Al-Hajhog, J. A., Simpson, M. A., Maolin, L., Mohiuddin, M. A., and Tan, C. P., 2010, Study of the Jilh Formation Overpressure and its Prediction: Saudi Aramco Journal of Technology, 52-59.

LIST OF PUBLICATIONS

AlBinHassan N. M. and Wang Y., 2010, Reservoir Porosity Estimation Using the Group Method of Data Handling (GMDH). 72nd EAGE Meeting, Expanded Abstracts, P111.

AlBinHassan N. M. and Wang Y., 2010, Carbonate Seismic Geopressure Prediction Using Neural Networks. 72nd EAGE Meeting, Expanded Abstracts, F046.

AlBinHassan N. M. and Wang Y., 2009,Geopressure Prediction in Carbonate Reservoirs Using Self-organized Mapping Method. 71st EAGE Meeting, Expanded Abstracts, P082.

AlBinHassan N. M. and Wang Y., 2007, Pore Pressure Prediction Using High-Resolution Velocities of Traveltime and Waveform Tomography. 69th EAGE Meeting, Expanded Abstracts, F036.

AlBinHassan N. M., Luo Y., and Al-Faraj M. N., 2006, 3D edge preserving smoothing and applications. Geophysics, 71, Issue 4, P5-P11.

AlBinHassan N. M. and Luo Y., 2005, 3D edge preserving smoothing and applications. 67th EAGE Meeting, Expanded Abstracts, F036.

Wang, Y., Luo Y., AlBinHassan N. M., and Alfaraj M., 2006, Computation of dips and azimuths with weighted structure-tenser approach: Geophysics,Issue 5, pp. V119-V121.

Wang, Y., Luo Y., and AlBinHassan N. M., 2005, Computation of dips and azimuths with weighted structure-tenser approach. 67th EAGE Meeting, Expanded Abstracts, P316.

AlBinHassan N., and Marfurt K., 2003, Fault detection using Hough transforms. 73rd SEG Meeting, Expanded Abstracts, 1719-1721.

Al-Dossary S., Marfurt K. J., and AlBinHassan N., 2003, Comparative study of edge detection algorithms. 73rd SEG Meeting, Expanded Abstracts.