

Reservoir Formation Facies Identification using Decision Tree Learning

by

Shaker Ali Ahmed Al-Faraj

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER SCIENCE

June, 1998

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

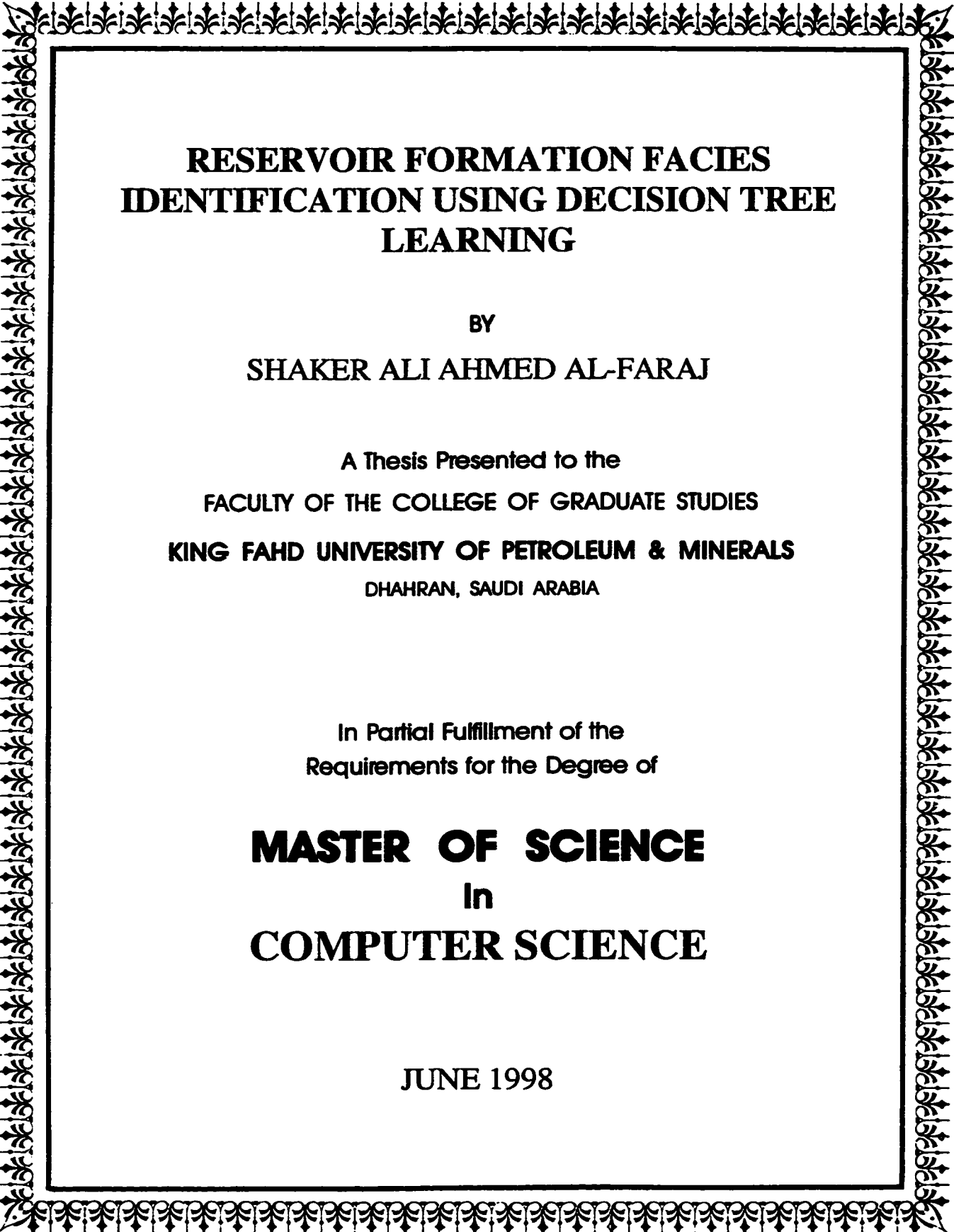
In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI

A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor MI 48106-1346 USA
313/761-4700 800/521-0600



**RESERVOIR FORMATION FACIES
IDENTIFICATION USING DECISION TREE
LEARNING**

BY

SHAKER ALI AHMED AL-FARAJ

A Thesis Presented to the
FACULTY OF THE COLLEGE OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE
In
COMPUTER SCIENCE

JUNE 1998

UMI Number: 1393201

**UMI Microform 1393201
Copyright 1999, by UMI Company. All rights reserved.**

**This microform edition is protected against unauthorized
copying under Title 17, United States Code.**

UMI
300 North Zeeb Road
Ann Arbor, MI 48103

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS
DHAHRAN, SAUDI ARABIA
COLLEGE OF GRADUATE STUDIES

This thesis, written by

SHAKER ALI AHMED AL-FARAJ

under the direction of his Thesis Advisor and approved by his Thesis Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE .

Thesis Committee



Dr. Hussein Al-Muallim (Chairman)



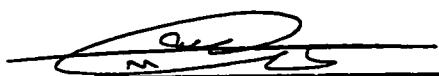
Dr. Sulaiman Al-Bassam (Member)



Dr. Jarallah S. AlGhamdi (Member)



Dr. Jarallah S. AlGhamdi
Department Chairman



Dr. Abdallah M. Al-Shehri
Dean, College of Graduate Studies

19-10-98

Date



Dedication:

I would like to dedicate my thesis to the memory of my nephew Nabil J. Al-Faraj, who left us tragically at an early age. He will forever remain in our hearts. May Allah grant him mercy and forgiveness.

Acknowledgments

Acknowledgement is due to King Fahd University of Petroleum & Minerals for all the support extended during this research.

I would like to express my profound gratitude and appreciation to my thesis chairman, Dr. Hussein AlMuallim, for his guidance and patience throughout this thesis. His continuous support and encouragement can never be forgotten. I feel grateful to my thesis committee members, Dr. Sulaiman Al-Bassam and Dr. Jarallah S. AlGhamdi for their continuous advice, guidance and cooperation.

I would like to thank Mr. MohammedHashem Al-Khatib for his continuous support. Special thanks for Mr. Dick Heil, chief geologist, who provided me with great help and clarification of many geological terms and concepts. Thanks to my colleagues Khalid Al-Zayer, Ahmed Al-Mulla, Sadiq Al-Nasser, Tawfiq Al-Faraj for their suggestions and comments during the preparation of my thesis draft copy, and for all of those who encouraged me. A very sincere appreciation to my wife Amal Al-Awami and my two daughters Reem and Yasmeen for their patience, sacrifices, encouragement and continuous love and support.

Acknowledgment	i
List of Tables	iv
List of Figures	vii
Abstract (English)	viii
Abstract (Arabic)	ix
1. INTRODUCTION.	1
1.2 Problem definition	4
1.3 Objective of this study	7
1.4 Motivation.	8
1.5 Thesis Organization	9
2. Geology Background.	10
2.1 Introduction	10
2.2 Coreing and core analysis	11
2.2.1 Obtaining cores	13
2.2.2 Cores for Special core analysis	14
2.2.3 Core preservation	14
2.2.4 Geological Studies.	15
2.3 Well logs	16
2.4 Facies.	19
2.4.1 Main advantages of facies identification.	21
2.5 Facies identification: Conventional approach.	23
3. Machine Learning: An Overview	27
3.1 Learning from examples: Popular approaches.	27
3.1.1 Classification rules.	30
3.1.2 Neural networks	32
3.2 Decision trees.	35
3.2.1 Basics of decision tree learning.	39
3.2.2 Attribute selection	42
3.2.2.1 The information-gain criterion.	43
3.2.2.2 The gain-ratio criterion.	45
3.3 DT size and pruning	46
3.4 Evaluation of DT.	48
3.5 The C4.5 software package	48
4. Literature Review	55
4.1 Introduction.	55
4.2 Premeability prediction/estimation.	56
4.3 Hydraulic Fracturing	57
4.4 Estimating PVT properties of crude oil.	58
4.5 Zone identification in a complex reservoir.	58
5. Formation Facies Identification Using Decision Tree Learning.	60
5.1 Introduction.	60
5.2 Data capturing phase.	63

5.2.1	Determining relevant well-logs.	65
5.2.1	Key-wells identification.	66
5.3	Data preparation phase.	68
5.4	Data transformation	71
5.5	Computing and adding additional variables	71
5.6	Training phase.	75
5.7	Model verification phase.	77
5.8	Case study.	79
5.8.1	Case study: Initial step.	81
5.8.2	Case study: Step two.	83
5.8.3	Case study: Step three.	87
5.8.4	Case study: Step four	91
5.8.5	Case study: Step five	94
5.8.6	Case study: Step six.	97
6.	Verification of the Final DT Model.	100
6.1	Introduction	100
6.2	Summary of the ten different tests	102
7	Summary and Conclusion	105
7.1	Summary	105
7.2	Conclusions and major contribution.	107
7.3	Future work	112
Keywords		116
References		117
Appendix A		121

List of Tables

Table	Page
1.1 A list of the facies to be predicted	6
2.1 A list of well logs that will be the essential . . . input to DTL algorithm	18
3.1 Eaxmples with 4 attributes and their classes	31
5.1 A list of the selected wells for this study and. . . their associated regions	67
5.2 A sample of the output file that contains the core . analysis results	70
5.3 A list of the substantial new varaibles that had . . been computed or added	74
5.4 A sample of input data used to train C4.5	76
5.5 A snapshot of some data for well X101.	78
5.6 Input to the C4.5 for Initial step	81
5.7 A summary of training & testing results for step 1 .	82
5.8 Confusion matrix for test data evaluation for. . . . Initial step	82
5.9 Input to the C4.5 for step 2	85
5.10 A summary of training & testing results for step 2 .	86
5.11 Comparison between the initial step and step 2 . . .	86
5.12 Confusion matrix for test data evaluation for. . . . step 2	86
5.13 Three new computed variables for step 3.	87
5.14 Input to the C4.5 for step 3	89
5.15 A summary of training & testing results for step 3 .	89
5.16 Comparison between the initial step and step 3 . . .	89
5.17 Confusion matrix for test data evaluation for. . . . step 3	90
5.18 Input to the C4.5 for step 4	93
5.19 A summary of training & testing results for step 4 .	93
5.20 Comparison between the initial step and step 4 . . .	93
5.21 Confusion matrix for test data evaluation for. . . . step 4	94
5.22 Input to the C4.5 for step 5	95
5.23 A summary of training & testing results for step 5 .	96
5.24 Comparison between the initial step and step 5 . . .	96
5.25 Confusion matrix for test data evaluation for. . . . step 5	96
5.26 Input to the C4.5 for step 6	98
5.27 A summary of training & testing results for step 6 .	98
5.28 Comparison between the initial step and step 6 . . .	98
5.29 Confusion matrix for test data evaluation for. . . . step 6	99
6.1 The standard set of input needed to develop DT model	102
6.2 A list of the tested wells of each test.	103

6.3	The evaluation results of DT model with respect. . to unseen data	103
6.4	The average confusion matrix for test data Evaluation	104

List of Figures

Figure	Page
1.1 An abstract view of how DT is constructed.	5
1.2 An abstract view of the input and output of DT	6
2.1 A core sample taken out from the formation	12
2.2 Cylindrical shape core sample.	13
2.3 Well logs that are commonly used for reservoir And will be used for our study	17
2.4 Reservoir can be zoned vertically into facies.	20
2.5 An abstract view to illustrate the importance. of facies identification	23
2.6 An illustration of fluctuation, separation and. closeness of well log that are examined.	25
3.1 Learning from example scenario	28
3.2 Conceptual view of the training/performance. phase.	30
3.3 Basic structure of neural network.	32
3.4 A simple decision tree	37
3.5 An illustration of recursive algorithm of DT	41
3.6 File defining labor-neg classes and. attributes (labor-neg.names)	50
3.7 A snapshot of the labor-neg.data	50
3.8 A snapshot of the labor-neg.test	51
3.9 The output of C4.5 on labor-neg data	52
5.1 An abstract view of the project process.	62
5.2 An abstract view of the data capturing process	64
5.3 Caculating GR_da and GR_db	84
5.4 An illustration of fluctuation, separation and. Closeness of well log that are examined.	88
5.5 An illustration of the fact that OAP and DOP facies might occur at 85% from top of the. formation.	91
5.6 An estimated location of OAP and DOP facies.	92
7.1 A summary of the results of the six-step case.	111
7.2 A summary of the results of the ten different. tests.	112

THESIS ABSTRACT

Name Shaker Ali Ahmed AlFaraj
Title Reservoir Formation Facies Identification
Using Decision Tree Learning
Degree MASTER OF SCIENCE
Major Field INFORMATION & COMPUTER SCIENCE
Date of Degree June 1998

During the past several years there has been a sudden and intense interest in the use of artificial intelligence techniques in petroleum industry. This thesis explores the use of machine learning approach, specifically decision tree learning, as a means to identify geological formation facies from well logs. Identifying geological formation facies is critical for economic successes of reservoir management and development. Formation facies usually influence the hydrocarbon movement and distribution. The identification of various facies, however, is a very complex problem due to the fact that most reservoirs show different degree of heterogeneity.

In this thesis, the existing methods are surveyed, and we propose a new methodology. The current conventional process to solve this problem is tedious and time consuming. Also, it is highly repetitive process and needed to be done for every single well. Notably, however, the other existing method was limited to the use of feed-forward neural networks. It is well known that NNs suffer the important shortcoming that they are not comprehensible by humans and can only be used as a "black box". We propose the use of decision tree learning (DTL) approach as a means to predict facies from well logs. We report a six-step case study on a real oil field. We identify a range of attributes, which could provide a diagnostic tool for facies identification. The C4.5 software package is utilized to build and test the constructed DT model. We achieved a satisfactory (average of 87.0% accuracy) result compared to the core analysis one. We verified the generality of the DT model by conducting ten different tests. The importance of the new approach to geologists and petroleum engineers and the advantages that this computing process has over other conventional method is discussed. The mechanics by which this technique achieves its objectives is also discussed. In addition, we discuss how the new approach is reliable, efficient, and more economic than the conventional method.

King Fahd University of Petroleum & Minerals, Dhahran.
June 1998

خلاصة الرسالة

اسم الطالب : شاكر علي أحمد الفرج
عنوان الدراسة : تمييز وجوه طبقات المكامن باستقراء شجرة القرارات
الدرجة : ماجستير في العلوم
التخصص : معلومات وعلوم الحاسب الآلي
تاريخ الشهادة : يونيه ١٩٩٨ م

خلال السنوات القليلة الماضية كان هناك اهتمام مفاجيء ومكثف لاستخدام الذكاء الإصطناعي في قطاع صناعة البترول. هذه الرسالة تستكشف استخدام تقنية الإستقراء الآلي وبالتحديد شجرة القرارات لتمييز وجوه طبقات المكامن الجيولوجية من واقع سجلات البئر. إن تمييز وجوه الطبقات الجيولوجية يعد أمرا حيويا للنجاح الإقتصادي في إدارة المكامن وتطويرها. وعادة ما تؤثر وجوه الطبقات بدرجة كبيرة في حركة الهيدروكربونات وانتشارها. ولكن تمييز وجوه الطبقات المختلفة يعد مشكلة عويصة لأن معظم المكامن تحوي درجات متفاوتة من عدم التناسق الهيدروكربوني.

هذه الرسالة تشرح الطرق المستخدمة لتمييز وجوه طبقات المكامن وتطرح طريقة جديدة لتمييزها. إن الطريقة التقليدية المتبعة لحل هذه المشكلة طريقة صعبة مملّة وتستغرق وقتا طويلا لإنجازها. كما أنها طريقة تكرارية بحتة ويلزم استخدامها لكل بئر على حدة. كما أن الطريقة الموجودة حاليا محدودة باستخدام تقنية التلقيم الإستباقي لأنموذج الشبكات العصبية. ولكنه من المعلوم أن أنموذج الشبكات العصبية يستغل على الأفهام ويتم استخدامه كصندوق أسود لا تعرف خباياه.

و الطريقة الجديدة تعتمد استخدام نوع آخر من طرق الذكاء الإصطناعي وهي طريقة استقراء شجرة القرارات كوسيلة للتنبؤ بماهية وجوه الطبقات من واقع سجلات البئر. لقد تم تقييم هذه الطريقة على سبع دراسات تجريبية لأبار نفطية في المملكة العربية السعودية. ولقد تم التعرف على مجموعة من الخصائص شكلت في مجموعها أداة لتشخيص وجوه الطبقات. هذا وقد تم استخدام الحزمة البرمجية C4.5 لبناء وتجريب أنموذج استقراء شجرة القرارات. لقد حققنا نتائج مرضية (٨٧% كمعدل للدقة) مقارنة بتحليل العينات الصخرية، كما أننا أثبتنا صحة شمولية أنموذج شجرة القرارات بإجراء عشرة اختبارات مختلفة. وتكمن أهمية نتائج هذه الطريقة للجيولوجيين ومهندسي البترول بأنها فاقت في دقتها جميع الطرق التقليدية المعروفة ذات الصلة. كما تم مناقشة الآلية التي حققت بها هذه الطريقة أهدافها. وقد أوضحنا مدى اعتماديتها وكفاءتها وكيف أنها أكثر اقتصادية من الطرق التقليدية.

جامعة الملك فهد للبترول والمعادن

الظهران - المملكة العربية السعودية

يونيه ١٩٩٨ م

CHAPTER I

Introduction

During the past several years there has been a sudden and intense interest in the use of artificial intelligence methods. Artificial Intelligence, sometime called machine intelligence, involves programming computers so that they will respond to situations in apparently the same way as humans [BOW89]. Since most human decisions are qualitative in nature rather than quantitative, it is possible to allow computers to make decisions based on logic and reasoning rather than on numbers alone.

High technique tools are obligatory in production and management of oil reservoirs in today's highly competitive environment. These tools form the foundation for cost reduction of exploration, production, and management of oil resources. Today, geologists and petroleum engineers are using new technologies from different disciplines to solve their problems. Normally, petroleum procedures utilize and employ advance computers in the work place, incorporating sophisticated simulation models in decision making processes, and digital control and monitoring of equipment that were regarded as state of the art only a few years ago [MA95]. These tools are furnishing engineers and scientists with the groundwork upon which intelligent methodologies can be developed.

Economic successes of reservoir management and development method depend very much on reliable reservoir characterizations. One of the major factors that invariably impacts production is that almost all reservoirs show some degree of heterogeneity. Heterogeneity in an oil reservoir is known as the non-uniform, non-linear special distribution of rock properties. However, lack of sufficient data to correctly predict the distribution of the formation make characterizations of a heterogeneous reservoir a complex problem [AMA90].

Therefore, accurate reservoir description plays a critical role in realistically predicting the performance of a complex reservoir. Well log, which is a measurement of the reflection of the electric or radioactive waves, presents measures of wide range of physical properties. Generally, it is very common that most, if not all, the wells in the reservoir are geophysically logged. One major use of well logs is to derive formation properties, e.g., identification of geological facies (the term facies means the general appearance or aspect of a rock) [HNKE94].

Professionals in the petroleum industry make important decisions to handle various tasks based on their previous experience. Quite often, the logic they follow in such situations is not precise. Different experts follow different logic, and even the same expert may not always use the same logic when re-examining a previous problem. Due to the above factor, it is highly desired to develop tools that provide a systematic way to perform the required tasks. An important goal is to automate the process as much as possible in order to rely less on human expertise. Conventional computing methods, however, have been unable to achieve this goal satisfactorily.

Even though geologists use their past experience to perform facies identification from well logs, this task is quite challenging and can be

handled only by skilled experts. Moreover, the required expertise usually varies depending on the physical location. Given these factors, constructing computer programs (expert systems) that perform such tasks automatically or semi-automatically is highly desirable. Nevertheless, such systems are hard to build by conventional means mainly because it is usually difficult for experts to explain their decisions in a precise manner, and consequently, it is hard to turn their expertise into a computer-executable form. Given the fact that a huge amount of historical data of well log interpretation (cases that were previously processed) is routinely kept, machine learning techniques have high potential to assist in overcoming this problem. Because of these characteristics, the formation facies prediction from well log interpretation in geology is ideally suited for the utilization of machine learning techniques. In this thesis, we will study how the decision tree learning (DTL) approach can be used to learn and predict geological facies from well logs data.

1.2 Problem Definition

Reservoirs show different degrees of heterogeneity, which make the identification and recognition of various facies a very complex problem.

However, these facies usually influence the hydrocarbon movement, distribution, and management therefore the identification of these facies seems to be a must [ABKM94]. Generally, it is very common that most if not all wells in a reservoir are geophysically logged. Identifying facies from well logs is manually done; however, this process is tedious and time-consuming. Therefore, utilizing machine learning techniques to identify facies based on these well log data seems to be a significant approach. The operations of this new method shall not require extensive knowledge of geology or the need for an expert geologist.

In this thesis, we are going to construct a decision tree (DT) based on a machine learning technique (MLT) to help in identifying geological formation facies from well logs. The DT is not going to be constructed manually, but will be built from examples. Figure 1.1 shows the abstract view of how this DT is constructed.

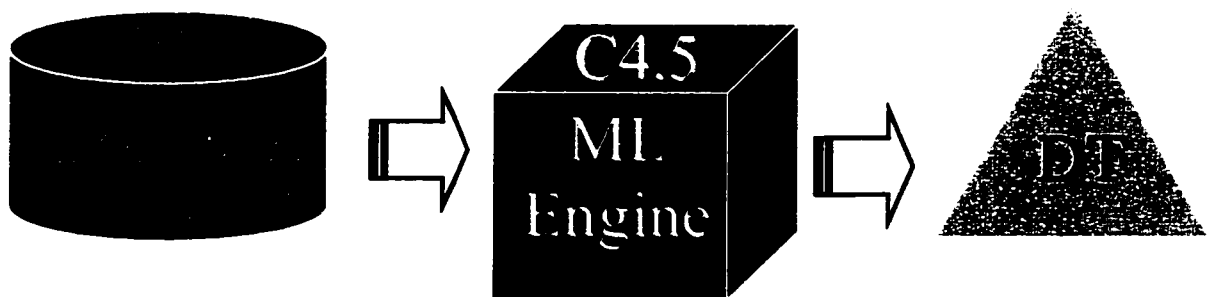


Figure 1.1: An abstract view of how DT is constructed.

The main input to the DT will be gamma rays log, bulk density log, neutron porosity log, and depth indicator and the output will be one of the geological facies shown in Table 1.1.

FACIES NAME	DESCRIPTION
S1	Slope, shallow (upper slope) or Slope, deeper (lower slope). Barrier undifferentiated.
S2	Lagoon or deep lagoon.
S3	Lithocodium-Coral and equivalent complex.
S4	Open algal platform deposits.
S5	Deeper open platform deposits.

Table 1.1: A list of the facies to be predicted.

Figure 1.2 shows the abstract view of the input and the output to the DT.

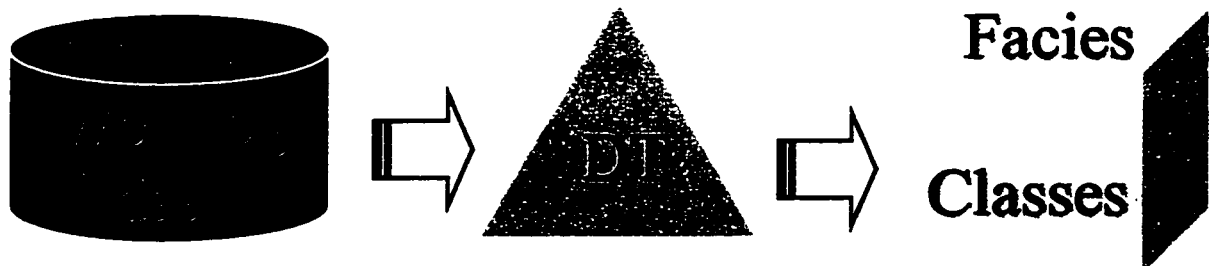


Figure 1.2: An abstract view of the input and the output to the DT.

1.3 Objective of This Study

The main goal of this study is to determine how well the DT approach can recognize facies from well logs. The objectives of this study are first, to identify ranges of well log values that could provide a diagnostic tool for facies identification. Second, to select cored wells having a complete suite of logs to provide a more comprehensive comparison between the result from geologist analysis and the DT technique using well logs as input. In our work, we intend to show how Decision Trees as MLT can be used to solve this quite complicated problem. We will discuss the importance of this new approach to geologists and petroleum engineers and the advantages that this computing process has over other conventional methods. The mechanics by which this technique achieves its objective will also be discussed. A real experimental implementation on real field data with its results will be fully discussed. In addition, we will show how the new approach of facies identification is reliable, efficient, and more economical than the conventional method. Thus this technique would assist geologists to make critical decisions about their geological models without requiring much geological experience.

1.4 Motivation

The motivation of this thesis is centered on the following points:

- Investigating whether or not formation facies predication from well logs data is feasible using the decision tree learning approach. Previous work had considered the use of neural networks to solve similar problems [Ali94,GE97]. No one explored the use of decision tree learning techniques to solve this complex problem.
- Using the current manual process to solve this problem is tedious and time consuming. We believe that this tool would help the geologist considerably in predicting facies, and save a significant amount of time in geological formation analysis. In addition, with relatively less need for core measurement information, this tool will cut cost as well as time in the process of facies identification.
- Following the current conventional process to solve this problem is highly repetitive and needs to be done for every single well. Such redundancy can be avoided by automating this procedure. The constructed DT facies model has the automation characteristic to escape the duplication of efforts done by the geologist.

1.5 Thesis Organization

In Chapter 2, we present basic background on geology and petroleum engineering that is needed for this thesis. Chapter 3 highlights the fundamental concepts of Decision Tree construction methodology. Literature review of machine learning techniques in petroleum industry is highlighted in Chapter 4. The main three chapters of this thesis are Chapters 5, 6, and 7. Chapter 5 contains the core material of this thesis where implementation, building, and testing of DT facies-model are presented. The six steps that had been carried out to build an effective DT model are fully discussed in this Chapter. In Chapter 6, we show that the constructed DT model is general enough by conducting ten different tests. In each of these tests, we use different wells to test the general performance of the DT model. Finally, Chapter 7 contains the discussion and the conclusions of this thesis, our major contributions, and future work.

CHAPTER II

Geology Background

2.1 Introduction

To exploit a reservoir, the geological model must accurately define the depositional environment and the effects of diagenesis on the pore network. Current methods for establishing the geological model of a field usually require subjective, qualitative interpretation of geological and petro-physical data. Hydrocarbon reservoirs are heterogeneous and non-uniform. However, these non-uniform and heterogeneous systems are made of multiple homogeneous groups (facies) [Ali94].

The variation of porosity and permeability corresponds to lithologic (facies) variation and this in itself fundamentally controls reservoir behavior. Hence, reliable reservoir characterization leads into economic success of reservoir management and development. However, one of the major obstacles that impact production is that most reservoirs show some degree of heterogeneity. Heterogeneity in a hydrocarbon reservoir is known as the non-uniform, non-linear special distribution of rock properties. Therefore, accurate reservoir description plays a critical role in effectively estimating the performance of a complex reservoir [CT90]. Generally, highly accurate prediction of facies could be achieved through core analysis. However, this process is very tedious and costly. Prediction can also be based on well logs. This approach may be less accurate, but it is considerably more economical since most if not all the wells in the reservoir are geophysically logged.

2.2 Coring and Core Analysis

Coring and core analysis are integral part of formation evaluation that provide vital information that is not available from either log measurement or productivity tests. Coring simply means that a column sample is taken from the reservoir formation (Figure 2.1). Core information includes detailed lithology, macroscopic definition of the heterogeneity of the reservoir rock, and capillary pressure data defining fluid

distribution in the reservoir rock system [ADD90]. It also includes information on the multiphase fluid flow properties of the reservoir rock. Core analysis is a very important part of an overall reservoir evaluation program. It provides direct evaluation of reservoir properties and also furnishes a basis for calibrating other evaluation tools such as well logs. It can be used as an effective mean for determining reservoir facies. It is more qualitative than logs in describing the reservoir but only a small number of wells are cored whereas all wells are logged. This is because coring and core analyses are very hard and expensive processes. However, early plans for reservoir development should provide for coring a reasonable number of wells [BMA95]. These well locations should be selected to provide representative coverage of the whole reservoir.

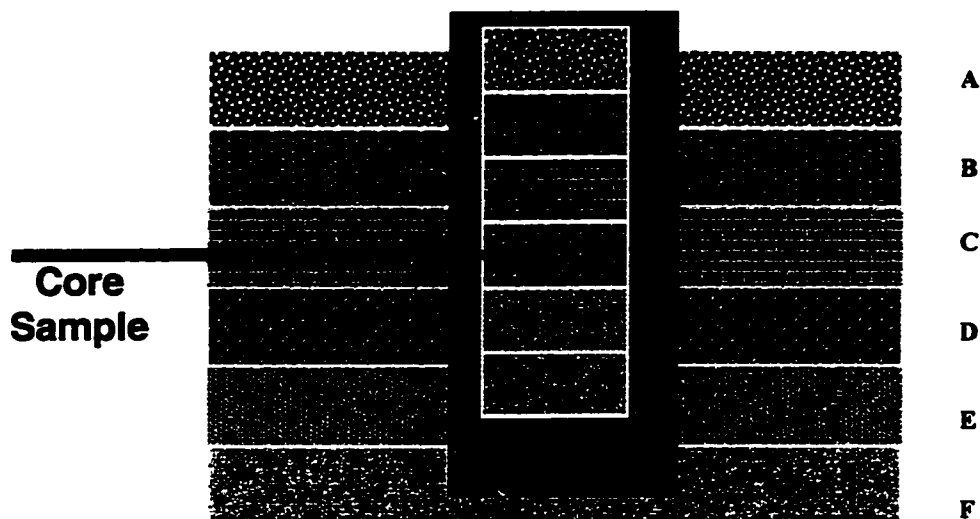


Figure 2.1: A core sample taken out from the formation

2.2.1 Obtaining Cores

A core is a sample of rock from the well section, generally obtained by drilling into the formation with a hollow section drill pipe and drill bit. There is a facility to retain the drilled rock as cylindrical sample (Figure 2.2) with the dimensions of the internal cross-sectional area of the cutting bit and the length of the hollow section. With conventional equipment, this results in cores up to 10 meters in length and 11 cm in diameter.

It is frequently found that variation in drilling conditions and in formation rock character prevent 100% recovery of the core. In addition, the core may also be recovered in a broken condition. In general, two partially conflicting objectives must be met when obtaining core samples [CT85]. First, a careful on-site examination for hydrocarbon traces is desirable (e.g. gas bubbling or oil seeping, and so on), in case an open whole drill stem test is possible and desirable. Second, it is desirable to keep the condition of core unchanged as much as possible prior to laboratory evaluation. We must emphasize that the process of obtaining a core in its best condition is a very delicate and costly job to do.



Figure 2.2: Cylindrical shape core sample

2.2.2 Cores for Special Core Analysis

The selection of a core for special core analysis is frequently a rather loose arrangement resulting from a reservoir-engineering request to the well-site geologist to reserve some representative pieces. While this approach may be inevitable with exploration wells, it should be possible to be more explicit during development drilling when reservoir zonation may be better understood. It is necessary to preserve samples from all significant reservoir flow intervals and these intervals must span permeability ranges [ABKM94].

It is necessary to specify the basis for facies recognition, the amount of samples required and the conditions for preservation, transportation and storage. In case of doubt, it is preferable to preserve too much rather than too little, and the geologist can always inspect the preserved core in the more controlled laboratory environment.

2.2.3 Core Preservation

The objective of core preservation is to retain the wettability condition of a recovered core sample, and to prevent change in petrophysical character. Exposure to air can result in oxidation of hydrocarbons or evaporation of core

fluids with subsequent wettability change. Retention of reservoir fluids (either oil or water) should maintain wetting character, so the core may be stored anaerobically under fluid in sealed containers. The core plug may be wiped clean, wrapped in a plastic seal and foil and stored in dry ice [MA95].

Usually only samples for special core analysis are stored and transported under these special conditions. The core for routine analysis, following visual inspection at the well site, is placed in boxes, marked for identification, without special care for wettability change or drying of core fluids. It is not really known whether this has any effect on the state of core fill/replacement minerals recorded in subsequent geological analysis.

2.2.4 Geological Studies

The purpose of a geological core study is to provide a basis for dividing the reservoir into facies and to recognize the geometry, continuity and characteristics of the various facies. The main areas of study involve recognition of the lithology and sedimentology of the reservoir and its vertical sequence of rock types and grain size. This is achieved by visual observation and the result recorded as a core log. The recognition of depositional and post depositional features is achieved by core description and by microscopic

observation of thin sections from cores. In addition, the fossil assemblages also provide indication of transport energy regimes (palynofacies analysis) which help support sedimentological interpretations.

The environmental and depositional model of a reservoir is largely based on the observations from individual cored wells but requires correlation of data between wells and integration with other sources of information, in order to provide insight into reservoir geometry and continuity.

2.3 Well logs

Well logs are measurements of the reflection of electric or radioactive waves that have been generated by log devices. They can provide valuable information on formations penetrated by the drill bit. Well logs and other types of logs can be important tools for determining reservoir zonation [DBB95]. Electric and radioactive logs can differentiate between sand and shale or between porous and nonporous limestone (Figure 2.3). Well logs also show the net sand thickness, which possesses permeability and contains recoverable hydrocarbon at each well.

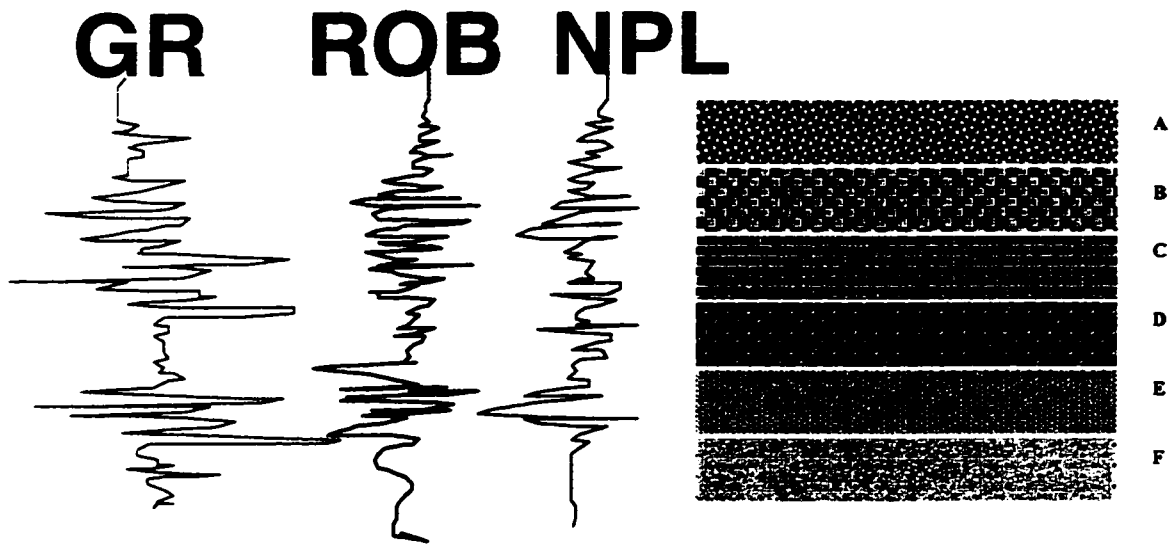


Figure 2.3: Well logs that are commonly used for reservoir and will be used in our study.

Generally, logs are available on all wells drilled in a reservoir and therefore, they represent the most complete set of reservoir descriptive data. An expert geologist should be consulted when analyzing logs for reservoir facies identification [MBA95]. There are various types of well logs that are commonly used for reservoir evaluation but the ones that initially interest us and will be used in our study are the following (Table 2.1):

<i>WELL LOG NAME</i>	<i>DESCRIPTION</i>
<i>GR</i>	Gamma ray log to measure natural radioactivity.
<i>ROBE</i>	Bulk density log to measure the electron density.
<i>NPLE</i>	Neutron porosity limestone log, respond to hydrogen in fluids.

Table 2.1: A list of the well logs that will be the essential input to DTL algorithm.

- **Gamma ray (GR) log:** Gamma ray log measures natural radioactivities in cased holes in oil-based mud wells. It can differentiate between shale and non-shale zone. Its natural radioactivity is greatest in shale and least in carbonates and sandstone is slightly more radioactive than carbonate one. (Figure 2.3).
- **Bulk density (ROBE) log:** This log is generally considered to be the best porosity tool under most conditions. The density log is a contact device containing a constant intensity gamma ray source and two gamma ray detectors. The density log measures the electron density of the formation and its contained fluids. Furthermore, the density of any material is roughly proportional to its electron density. (Figure 2.3).

- **Neutron porosity limestone (NPLE) log:** The neutron porosity tools contain a constant intensity neutron source and a detector. Mostly hydrogen atoms stop neutrons, so the intensity at the detector becomes less, as hydrogen density becomes greater. Most reservoir rock material contains little or no hydrogen, so the log responds to hydrogen in fluids contained in the pores. In shale free sandstone or carbonate, the neutron log can be a good porosity tool. (Figure 2.3).

2.4 Facies

Originally the term facies meant the general appearance or aspect of a rock. Today facies is used in many ways. These include lithologic character, metamorphism, biofacies, stratigraphic relations, structural form, and environmental influence. The facies of interest to geologists, petroleum engineers, and to us in this study are lithofacies and environmental facies. Lithofacies include the physical properties of a rock such as color, mineral composition, bedding, etc. Recognition and mapping environmental facies are important to reservoir engineers because the facies may be thick, widespread and act as a single unit. For example, the term pro-delta facies tells the engineer that although these rocks are probably of non-reservoir quality, they can serve as cap rocks and source rocks but would be of no value as receivers of water for reservoir maintenance [HNKE94]. In any depositional

environment there are facies changes or variations reflecting non-uniform conditions in either or all the source area.

Most reservoirs are deposited from water and are layered because of variations that existed in the depositional environment. Slow moving water deposits mostly small grain particles at a specific location, then when the water is moving much faster relatively large particles will be deposited at the same place. This results in a vertical series of dissimilar units [MAAN96]. Conditions will also vary from one location to another at the same time. Many people think of reservoir formation only in terms of net sand layers and impermeable streaks of sand or shale. This is generally correct, but a smaller scale of different facies also exists within the net sand layers. And although all of the net sand contains hydrocarbons and possesses permeability, the degree of porosity and permeability can vary greatly. A reservoir can be zoned based on well logs and core analysis data and it can be divided vertically into zones or facies (Figure 2.4) [MAA94].

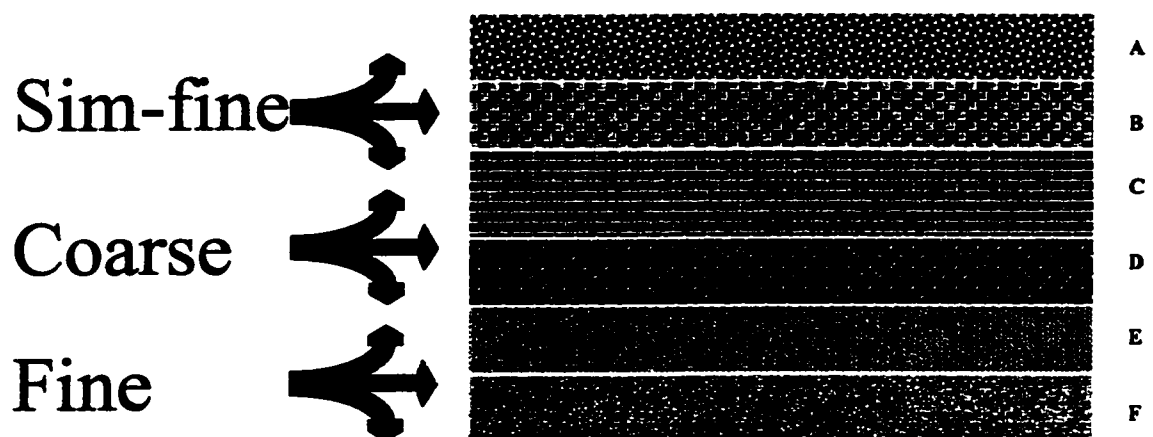


Figure 2.4: Reservoir can be zoned vertically into facies.

2.4.1 Main advantages of facies identification

One of the key issues in the description and characterization of heterogeneous formations is the distribution of various facies and their properties. Therefore, to exploit a reservoir, the geological model must accurately define the depositional environment and the effects of diagenesis on the pore network [MMAA95]. Reservoir behavior is fundamentally controlled by the variation of porosity and permeability. However, the variation of these two most important proprieties of reservoir formation rock corresponds to lithologic (facies) variation [GE97]. The following are some of the main reasons that make geologists and engineers interested in facies identification:

- Oil saturation and productivity in any field highly depend on facies and diagenetic modifiers that control connectivity, heterogeneity geometries and dimensions of flow units.
- By identifying facies, you are indirectly estimating the porosity and permeability of these zones, and this is a must for any reservoir development, production, and management.

- Understanding the field-wide relationships between depositional facies, structural evolution and diagenetic overprint will be vital to flow-unit correlation and permeability prediction in the reservoir.
- The key to 3D reservoir characterization and modeling lies in fully understanding the depositional framework and its diagenetic evolution.

Figure 2.5 informally illustrates the importance of facies identification for economic and successful reservoir description.



Figure 2.5: An abstract view to illustrate the importance of facies

2.5 Facies identification: Conventional approach

The task of estimating geological formation facies for a given reservoir has been a very challenging and time-consuming problem for geologists. However, the identification of these facies is a task that the geologist must do. There are various types of well logs that can be used to help in identifying the general

reservoir characteristics. However, a highly experienced geologist who is familiar with well logs must be consulted to identify the interdependence between well logs and core on one hand and the facies on the other hand.

The three most commonly used well logs by geologists to identify the facies are the gamma ray (GR), density (ROHE), and neutron porosity logs (NPLE) as described in Section 2.3 [MA95]. These are the procedures followed by geologists when describing facies from visual inspection of log-data curves utilizing their past experience in doing so:

1. Identify which well logs from the drilled-well that will be used in the identification of the facies.
2. Identify some wells that span the whole reservoir and have both the selected-well-logs identified in step one, and also these wells which have been cored.
3. Geologists rely on experience and memory of what facies exist, what depositional environment exists; they examine the appearance of the log curves in various depositional environments using graphical tools.
4. Geologists read the degree of fluctuation on each well log with correspond to depth. They also inspect the degree of separation and closeness of the log curves to one another, when drawn to the same scale.

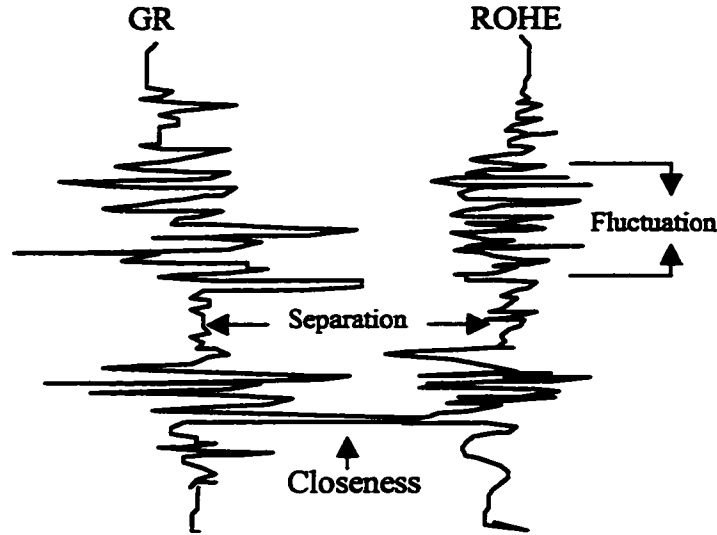


Figure 2.6: An illustration of Fluctuation, Separation, and Closeness that are examined by geologists.

5. If the results read from the log curves and the core data relatively match, then geologist is able to integrate what is seen on the logs to produce a complete facies interruption.
6. This information goes into the geologist's memory so that on other wells where logs alone are available, the geologist can form as competent an interpretation as was performed in wells where cores were also available.
7. In addition, geologists reason to a conclusion about facies description from noisy, incomplete, and uncertain log data. Therefore, sometime geologists apply log-to-facies correlation scenarios to select the most likely emerging scenario.

Although, the conventional approach that geologists follow to identify facies is doable and gives a satisfactory result, this approach is tedious and time consuming. This task is quite challenging and requires an expert geologist to handle it. Moreover, the required expertise usually varies depending on the physical location. Given the fact that abundance of historical data of well log interpretation is usually kept, machine learning methods have high potential to help in overcoming this problem. In Chapter 5 of this thesis, we will study how the decision tree learning (DTL) approach can be utilized to learn and predict geological facies from well-logs data.

CHAPTER III

Machine Learning: An Overview

3.1 Learning From Examples: Popular Approaches

As mentioned in Chapter 2, our goal is to develop a classifier that determines the type of facies at a given depth given well logs. One may think of constructing such a classifier by manually writing a program for it. For example, a classifier may be represented as nested if or case statements so that new cases are matched against the conditions of these statements in

order to make decisions. However, as the number of properties and the number of previous cases get larger, the program becomes more expensive and more difficult to construct. A promising approach to ease the construction of such a classifier is to employ some learning mechanism (Figure 3.1 & Figure 3.2) to automatically induce classifiers from actual cases or examples that have been previously handled by the domain experts.

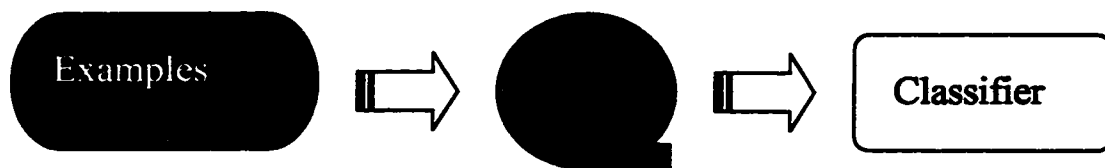


Figure 3.1: Learning from example scenario

This scenario of extrapolating from the given training examples is useful in real-world applications where it is difficult to manually turn human expertise into a programmable form. For example, it may be hard for a doctor to explicitly tell us what exactly the rules he/she follows to diagnose a certain disease. On the other hand, it is relatively easy to collect training data by gathering the doctor's final decisions after examining his/her patients. The conventional approach in expert systems research is to ask the doctor about how he makes his decisions. His answer has then to be encoded into a computer program, the expert system. In contrast, in the

machine learning approach, we start by collecting a sufficient set of examples, where each example consists of the symptoms of a patient and a label that shows the doctor's response. We then employ some learning algorithm to automatically extract from these examples an appropriate diagnosis rule (classifier) which simulates the rule used by the doctor.

This machine learning approach enjoys several advantages: (i) In problems where knowledge is expert-dependent, one can simply learn from examples handled by different experts, with the hope that this will average the differences among different experts. (ii) Being able to construct knowledge automatically makes the upgrading task easier since one can rerun the learning system as more examples accumulate. Some learning methods are indeed "incremental" in nature. Figure 3.2 shows the conceptual view of the training phase and performance phase.

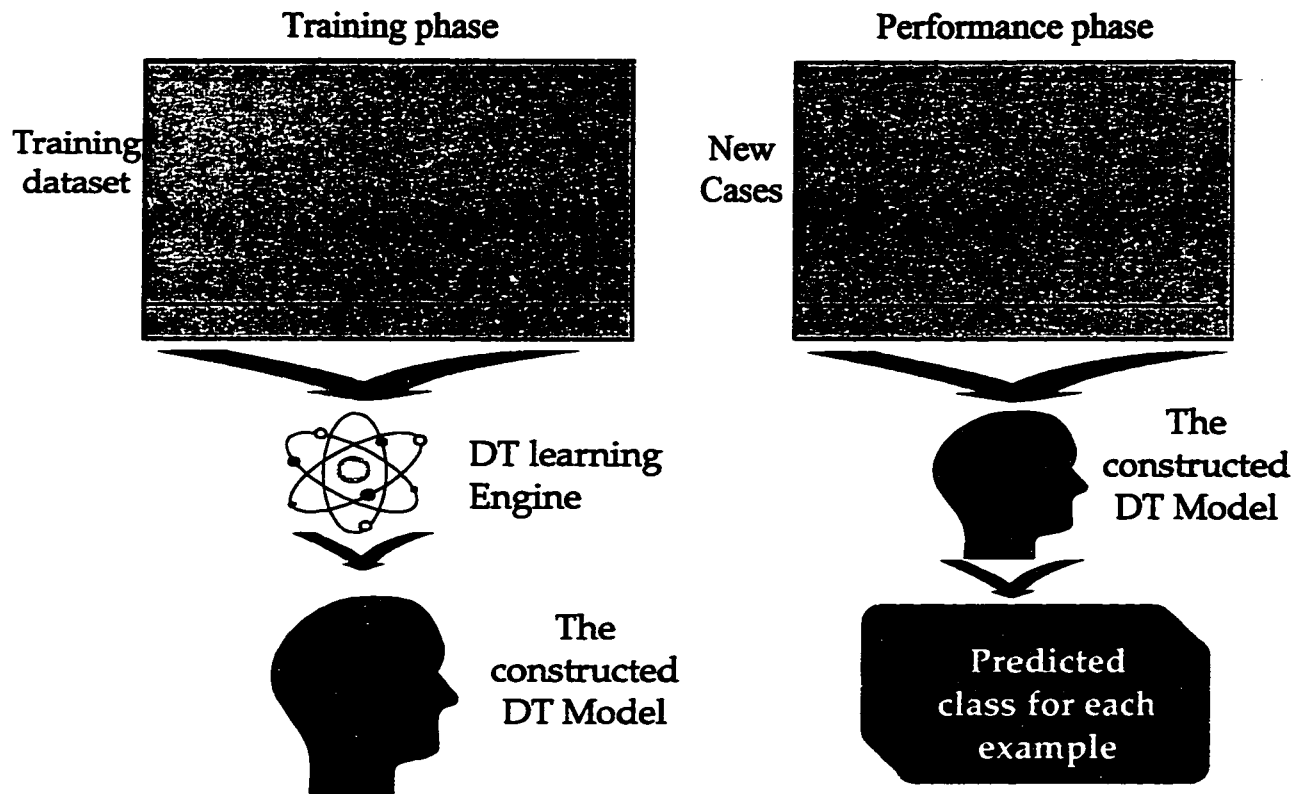


Figure 3.2: The conceptual view of the training phase and performance phase.

Classifiers induced on a training set (i.e. learn from examples) can be represented in many forms. The common ones involve sets of classification rules, neural networks, and decision trees. In this thesis, our focus will be limited to the latter form of classifiers, i.e. decision trees. However, we will briefly touch on the former two forms of classifiers here.

3.1.1 Classification Rules

In this thesis, we will restrict our attention to the task of learning classification rules from previous cases or examples. Consider the simple

task of deciding on a certain day whether to play soccer or not. Let us look at how human would construct a decision rule for this task. First, he would recall previous experiences regarding this task. Then, he would study those experiences trying to detect the factors that led to the decision at each experience. These factors or properties could involve the day outlook (sunny, overcast, or rainy), the temperature, the rate of humidity, and possibly whether it was windy or not. Table 3.1 shows two previous experience with four properties of the weather along with the decision made in each case. Based on the decisions made at such experiences, he would then construct the desired rule that can be applied to make a decision for future cases.

Outlook	Temp (F)	Humidity	Windy?	Class
Sunny	75	Normal	True	Play
Sunny	80	High	True	Don't Play

Table 3.1: Two previous experiences each with 4 attributes and their class

A classification rule for making a decision consists of a specification of the values of one or more properties on the left-hand side and that decision on the right hand side [Quin90]. One rule that can be derived from the examples of Table 3.1 is

IF (Outlook = sunny and Humidity = normal) THEN the decision is to Play.

Learning classifiers as a set of rules is not easy because obtaining a small and consistent set of rules to be used in classifying cases such that one rule will match any single case is a difficult task [Quin90].

3.1.2 Neural Networks

The powerful operational capability of thinking, remembering, and problem solving of the human brain encouraged scientists to attempt simulation of its operation using computer models. Hence, the birth of artificial neural networks (ANN) technology and the name “neural networks.” Figure 3.3 shows the basic structure of an ANN.

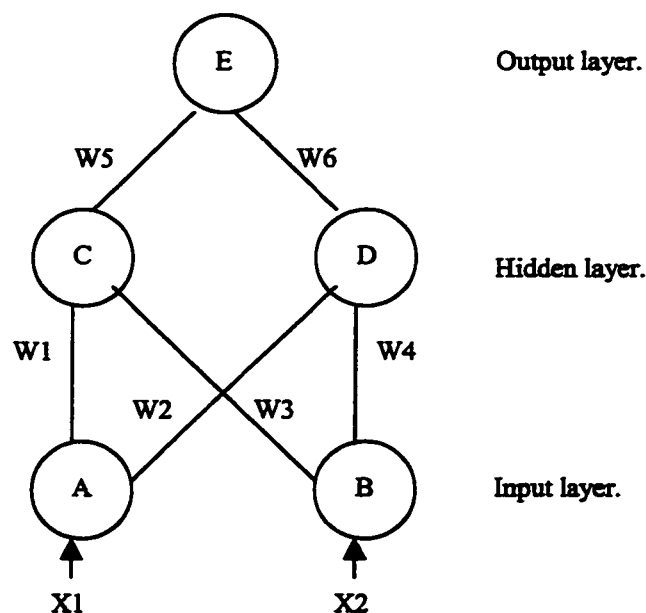


Figure 3.3: The basic structure of an ANN

An ANN is constructed from several artificial neurons, arranged in layers and connected to each other with weights. Usually there are three layers in a typical neural network: input units such as \mathcal{A} and \mathcal{B} that introduce information from the environment to the network, hidden units such as \mathcal{C} and \mathcal{D} , and output units such as \mathcal{E} that carries the result. Each link has an associated weight and some units have a bias. To process a case, the input units are first assigned numbers between 0 and 1 representing the attribute values. The numbers of artificial neurons in the input and output layers are predetermined by the application; however, the numbers of artificial neurons in the hidden layer are determined by the NN training [MS97].

Referring to Figure 3.3, the processing element (PE) has several input paths and usually combines their values by simple summation. This results in an internal activation level of the PE. The combined inputs are transferred to the output using a transfer function. The transfer function can be a threshold function, which passes the information only if the combined activity levels reach a certain value. Alternatively, the function could be a continuous transfer function that regularly transfers the combined inputs. The transferred value is generally passed directly to the output path of the PE [MS97]. Since each connection has a corresponding weight, the signal

in the input line to a PE is modified by this weight prior to being summed. Thus, the summation value is called a weighted sum.

Several nonlinear functions are used to transfer the weighted sum of a PE. The most commonly employed functions are sigmoid ($f(x) = 1/(1+e^{-x})$) and hyperbolic tangent function ($\tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$).

The values of the network weights and the biases are learned through repeated examination of the cases. The deviation of each output unit's output from its correct value for the case is propagated back through the network, where all relevant connection weights and unit biases are adjusted, using gradient descent, to make the actual output closer to the target [Murp97]. Training continues until the weights and biases stabilize.

NN may be considered strictly as "black box" classifiers. Decision trees and symbolic rule classifier approaches produce classifiers in a symbolic logical format that is intended to be meaningful to humans. Moreover, NN requires a large amount of time to be trained. Cases are typically iterated through the network thousands of times before the NN system converges on a local minimum.

Furthermore, there is an important network design problem that remains open: How many internal layers/units should a network have for a given learning task? If this number is too large, the network will simply rote learn the training set and no induction will take place. On the other hand, if it is too small, the network may never be able to converge on a solution that is consistent with the set of cases. In addition there are many parameters that a user needs to specify such as learning rate and the initial connection weights. The reliance on many user-specified parameters as well as the initial settings problem are factors that make decision tree approach more practical than NN approach for industrial applications of machine learning [Murp97].

3.2 Decision Trees

Let \mathcal{A} be a set of attributes and C be a set of classes. A DT is a tree structure with the following criteria:

- Each non-terminal node, called a decision node, is labeled with a test involving one of the attributes in \mathcal{A} . The test has a finite number of disjoint outcomes.

- Each outgoing branch from a non-terminal node corresponds to one of the outcomes of the test at the node.
- Each terminal (leaf) node is labeled with one of the classes from set C .

A DT is used as a classifier by passing each example X that has A attributes down through the tree beginning at the root (topmost) node. A test node may contain a simple test of the form $a_i > \kappa_i$ where a_i is an attribute and κ_i is threshold test. If the test is true, the example is sent down the “yes” branch of the tree; otherwise it goes down the “no” branch. This process continues until the example reaches a leaf node. Each leaf in the tree represents a class (classification rule) [Quin96]. The conjunction of tests on the branches from the root to that leaf constitutes the preconditions of that rule. Figure 3.4 shows a simple DT, which is a possible classifier for the soccer problem described earlier in Table 3.1.

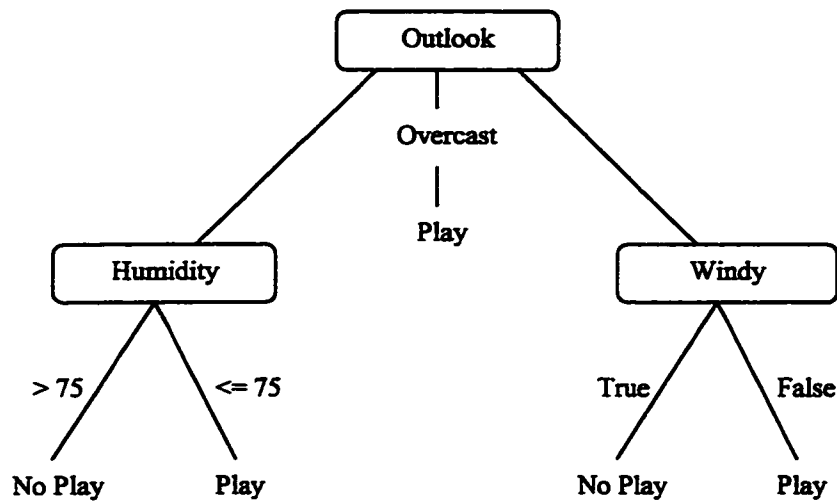


Figure 3.4: A simple DT.

Many DT learning methods have been used for classification, most notably C4.5 [Quin93] and CART. In this work, decision trees will be constructed using the C4.5 package.

A DT represents only one approach to the concept learning problem. What make DT preferable over other approaches are the following factors [Quin96]:

1. A DT offers an efficient means for processing large data sets. This is because the learning problem is being partitioned among branches into sub-problems with smaller sets of training data.

2. DT can be transferred into symbolic classification rules, as was done in C4.5. These rules are symbolic since they are expressed as simple high level conditions, and therefore, they are not difficult for domain experts to understand.
3. Unlike NN, other than what is provided in the set of cases, as parameters to the DT generation approach, no extra information is required.

Looking at these factors, the DT learning approach has the potential for being a powerful and flexible classification tool. However, one must be aware of the following five criteria [Quin93].

1. The C4.5 algorithm should be exposed to the extremes of the data to allow for optimum modeling.
2. The C4.5 should be disclosed to all significant variations in the data to obtain a comprehensive view of the data. Therefore, the training dataset that will be utilized for training C4.5 must symbolize the major characteristics of the collected data.

3. No duplication of training and testing data is allowed. This means that none of the wells used in the training phase may appear in the testing stage.
4. The number of the training and testing wells depends on the complexity of the problem. This is to say that the amount of data required is affected by factors such as the number of properties, the number of classes, and the complexity of the classification model. As these factors increase, more data will be required to build a reliable model.
5. Selections of the attributes used to represent the cases have great impact in the development of the DT model.

3.2.1 Basics of Decision Tree Learning

This section gives a quick overview of the process of DT construction. A DT is induced on training set S , which consists of cases. Each case is completely described by a set of attributes $A = \{a_1, a_2, \dots, a_n\}$ and a class belong to the set of classes $C = \{c_1, c_2, \dots, c_k\}$. The concept underlying a data set is the true mapping

between the attribute set and the class label. A noise-free training set is one in which all the cases are generated using the underlying concept.

The task of constructing a tree from the training set is called tree induction. Most existing tree induction systems proceed in a greedy top-down fashion. Starting with an empty tree and the entire training set, the following algorithm is applied until no more splits are possible [Quin96].

- If all the examples at the current node t belong to class c_i , create a leaf node with class c_i and return.
- Otherwise, score each one of the sets of possible splits of S , using a goodness measure.
- Choose the best split s^* as the test at the current node, and create as many child nodes as there are distinct outcomes of s^* .
- Label edges between the parent and child nodes with outcomes of s^* , and partition the training data using s^* into the child nodes.

The algorithm is recursively applied on each subset S_i of cases to generate a subtree T_i as illustrated in Figure 3.5.

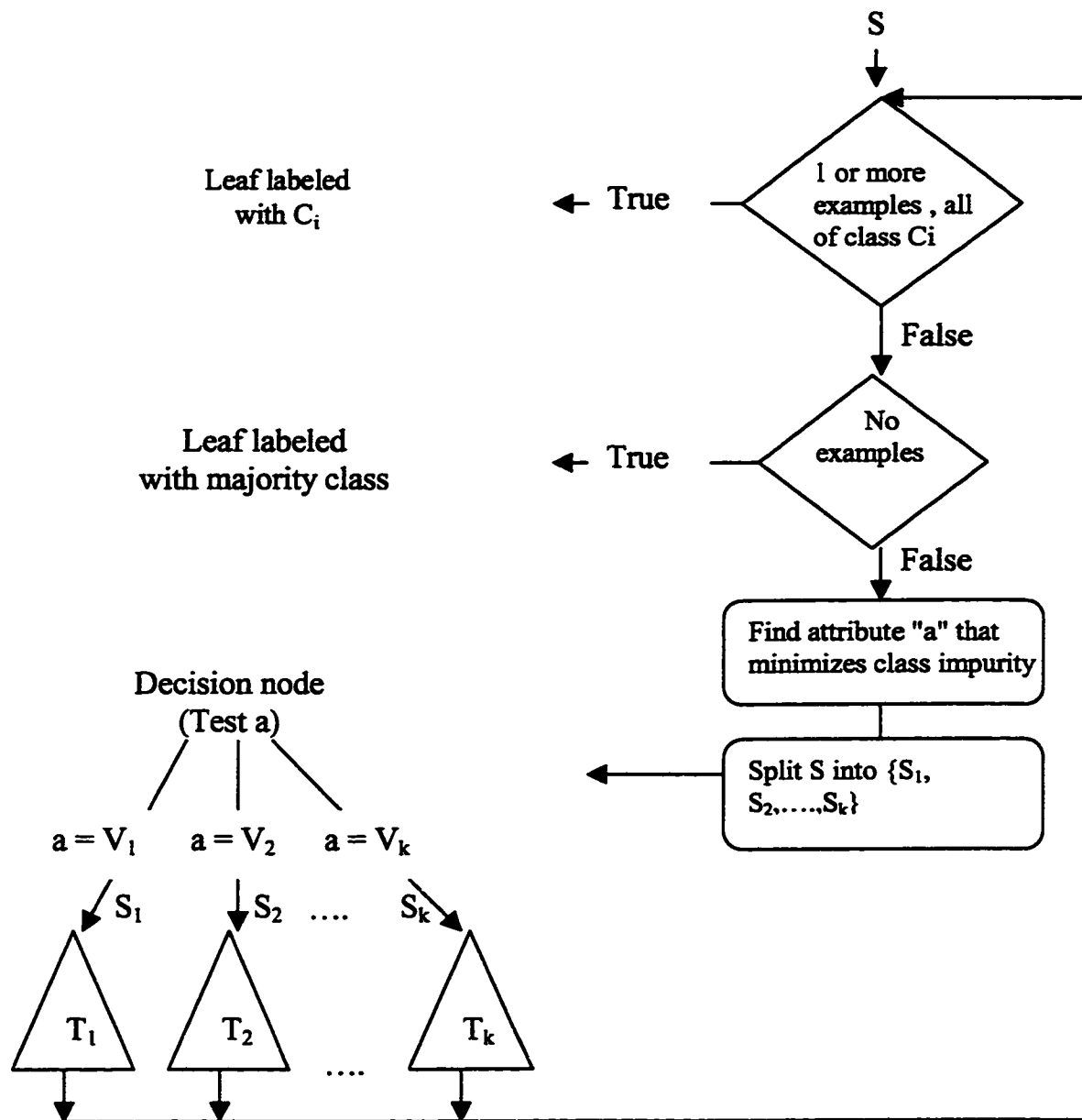


Figure 3.5: An illustration of recursive algorithm of DT.

3.2.2 Attribute Selection

The tree generated by the top-down approach depends on the choice of tests at each recursive call. From the same training data, many consistent DT (a tree that matches the training set) can be constructed using a different choice of tests. Even though these trees are indeed consistent with the training cases, some of them might perform poorly when applied to unseen examples. In a tree building process, any test that splits the set of cases S in a nontrivial way, such that at least two of the subsets $\{S_i\}$ are not empty, will eventually result in a partition into single-class subsets. However, the tree-building process is intended to build a tree that reveals the structure of the domain, and therefore has predictive power [Quin96].

For that, we need a significant number of cases at each leaf. In other words, we want a partition that has as few divisions as possible. Therefore, more compact trees will be favored over larger trees. In fact, with compact trees, as more cases are used to choose a test at each recursive call, the confidence in this choice increases and so does the predictive power.

A DT building method is usually implemented in a non-backtracking, greedy fashion. Once a test has been selected to partition the current set of cases, usually on the basis of maximizing some local measure of progress, the choice is set and

the consequences of alternative choices are not explored [Quin93]. With most data sets, if attributes are selected randomly, classification performance of the generated DT tends to be poorer than that of trees constructed with careful attribute selection criteria. Obviously, we need a measure that indicates how good choosing a certain test over another is. In the next section, two common measurements are explained, namely, the information-gain and the gain-ratio.

3.2.2.1 The Information-Gain Criterion

One of the popular measures used to evaluate tests is the information-gain. The information theory underlying this criterion states the following: the information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm base 2 of that probability [Quin93]. For example, if there are eight equally probable messages, the information conveyed by any one of them is $(-\log_2 1/8)$ or 3 bits.

Let S be a set of cases and let $\text{freq}(C_i, S)$ stand for the number of cases in S that belong to class C_i . And let the notation $|S|$ to denote the number of cases in the set S . For simplicity, we will consider a domain with two classes, P and N .

If we pick a case from the set S at random and say that it belongs to some class C_j . This message has probability $\text{freq}(C_j, S) / |S|$ and so the information it conveys is

$$-\log_2[\text{freq}(C_j, S) / |S|] \quad \text{bits.} \quad (3.1)$$

The expected information from this case is calculated by summing up the classes in proportion to their frequencies yielding:

$$\text{Info}(S) = -\sum_{i=1}^k \text{freq}(C_i, S) / |S| \log_2(\text{freq}(C_i, S) / |S|) \quad \text{bits.} \quad (3.2)$$

This measure gives the average amount of information needed to classify a case in S and is also known as the entropy of the set S .

Now let us assume that the set S has been partitioned based on a test X with n outcomes. The expected information requirement can be found as the weighted sum over the subsets as follows.

$$\text{Info}(S)_x = -\sum_{i=1}^n |S_i| / |S| \times \text{Info}(S_i). \quad (3.3)$$

Therefore, the information-gain is

$$\text{Info-gain}(X) = \text{Info}(S) - \text{Info}(S_i)_x. \quad (3.4)$$

Information-gain measure indicates the information gained by partitioning S on the test X . Therefore, the test that maximizes the information gain will be chosen.

3.2.2.2 The Gain-Ratio Criterion

One serious deficiency in the information gain measure is its bias favoring tests with many outcomes. For example, consider a training set of cases from a hypothetical medical-diagnosis domain. Let one of the attributes be the patient identification, which is usually unique. If this attribute is to be used to partition, we will end up with a large number of partitioned subsets, each of which contains only one patient example. Since all of these one-case subsets necessarily contain cases of a single class, $\text{Info}(S)_x = 0$, and the information gain from using this attribute to partition the set of training cases becomes maximal. However, this partitioning is useless for predicting new cases, especially if we have a large number of cases with unique ID's where it would be very costly to store the tree in term of space [Quin97].

To correct this bias, the information-gain measure has to be adjusted by a kind of normalization in which the apparent gain of attributes with many outcomes is reduced. This is done by considering the outcome of the test to classify a case instead of the class to which the case belongs. By analogy with the definition of $\text{Info}(S)$, we have

$$\text{SplitInfo}(X) = - \sum_{i=1}^n (|S_i|/|S| \log_2 |S_i|/|S|). \quad (3.5)$$

While the information-gain measures the information relevant to classification that arises from partitioning the set S into n subsets, the `SplitInfo` represents the potential information generated by the same division [Quin93]. The gain-ratio is therefore

$$\text{Gain-Ratio}(X) = \text{gain}(X) / \text{SplitInfo}(X) \quad (3.6)$$

The gain ratio measures the proportion of information generated by a useful split, i.e., a split that seems to help classification. The test that maximizes the ratio above is to be selected.

3.3 DT Size and Pruning

One of the main difficulties of inducing a recursive partitioning structure is knowing when to stop splitting the training cases and growing the tree. Obtaining the “right”-sized trees may be important for several reasons, which depend on the size of the classification problem [Quin96]. The critical issue is the relationship between the size of the tree and its accuracy with respect to both the training and the testing phase. A large and complex tree that had been constructed from training examples might fit or sometime over-fit all cases in the training set leading to good accuracy with

respect to the training data. However, that tree might have poor or non-acceptable performance with respect to the unseen test data. Therefore, a simpler tree might have less accuracy with respect to training data but it will be more general and it will have better performance with respect to the unseen data [Quin93].

Pruning is one method most widely used for obtaining right size trees. There are basically two ways in which the recursive partitioning method can be modified to produce simpler trees: deciding not to divide a set of training cases any further, or removing retrospectively some of the structure built by recursive partitioning [Quin93].

The former approach, usually called stopping or pre-pruning, has the attraction that time is not wasted assembling structure that is not used in the final simplified tree. The typical approach is to look at the best way of splitting a subset and to assess the split from the point of view of statistical significance, information gain, error reduction, or any other criterion. Breiman et al suggested the following procedure: build a complete tree, a tree in which splitting no leaf node further will improve the accuracy on the training data, and then remove sub-trees that are not contributing significantly towards generalization accuracy. It is argued that this

method is better than stop-splitting rules, because it can compensate, to some extent, for the sub-optimality of greedy tree induction.

3.4 Evaluation of DT

Generalization performance means how well a DT is able to classify new unseen cases. Naturally, one can always build a tree that achieves 100% accuracy on the training set. Usually, however, we would like to know how accurate a tree will be at classifying other, unseen examples drawn from the same distribution as the training set. To estimate this accuracy, a standard practice is to reserve a portion of the training data as a separate test set, which is not used in building the tree. The accuracy of the tree on this test set is then used as an estimate of the accuracy for unseen examples [Quin96, MS97].

3.5 The C4.5 software package

In this study, we are going to use C4.5 as the software package to build the DT classifier. C4.5 is a commercially available software package released by Quinlan. C4.5 requires a "filename.names" which will include the domain's classes and attributes. Also, C4.5 needs a "filename.data" which will include all the examples from the training data set. This

`filename.data` will be the input for C4.5 at the training phase. The examples in `filename.data` will provide knowledge about the domain in general. However, to provide information on an individual example will require spelling out the variables for each example separated by a comma, followed by the case-class, and then by a dot to indicate to C4.5 the end of each example. A `filename.test` which will include unseen examples from the testing data set is also needed by C4.5.

We will now consider a small example, intended to illustrate the input to and the output of C4.5. The “Labor-neg” is an example that records the outcome of Canadian contract negotiations in 1987-1988. The first step is to define the classes and attributes by preparing a file `labor-neg.names`, shown in Figure 3.6. The file specifies the classes (in this example, good and bad), then the name and description of each attribute. Some attributes, such as “duration” and “wage increase first year”, have numeric values and are described just as continuous; others, such as “pension” and “vacation”, have a small set of possible values that are listed explicitly in any order [Quni93].


```

|Classes
|-----

good, bad.

|  Attributes
|  -----

duration:                continuous
wage increase first year: continuous
wage increase second year: continuous
wage increase third year: continuous
cost of living adjustment: none, tcf, tc
working hours:          continuous
pension:                none, ret_allw, empl_contr
standby pay:            continuous
shift differential:     continuous
education allowance:   yes, no
statutory holidays:    continuous
vacation:               below average, average, generous
longterm disability assistance: yes, no
contribution to dental plan: none, half, full
bereavement assistance: yes, no
contribution to health plan: none, half, full

```

Figure 3.6: File defining labor-neg classes and attributes (labor-neg.names)

While Figure 3.7 is a snapshot of the labor-neg.data that will be used as input to the C4.5 at the training phase to construct a DT model.

```

-----
3,3.7,4.0,5.0,tc,?,?,?,?,yes,?,?,?,?,yes,?,good
3,4.5,4.5,5.0,?,40,?,?,?,?,12,average,?,half,yes,half,good
2,2.0,2.5,?,?,35,?,?,6,yes,12,average,?,?,?,?,good
3,4.0,5.0,5.0,tc,?,empl_contr,?,?,?,12,?,yes,none,yes,half,good
3,6.9,4.8,2.3,?,40,?,?,3,?,12,below average,?,?,?,?,good
2,3.0,7.0,?,?,38,?,12,25,yes,11,below average,yes,half,yes,?,good
1,3.0,?,?,none,36,?,?,10,no,11,generous,?,?,?,?,good
2,4.5,4.0,?,none,37,empl_contr,?,?,?,11,average,?,full,yes,?,good
1,2.8,?,?,?,?,35,?,?,2,?,12,below average,?,?,?,?,good
1,2.1,?,?,tc,40,ret_allw,2,3,no,9,below average,yes,half,?,none,bad
1,2.0,?,?,none,38,none,?,?,yes,11,average,no,none,no,none,bad
2,4.0,5.0,?,tcf,35,?,13,5,?,15,generous,?,?,?,?,good
2,4.3,4.4,?,?,38,?,?,4,?,12,generous,?,full,?,full,good
2,2.5,3.0,?,?,40,none,?,?,?,?,11,below average,?,?,?,?,bad
3,3.5,4.0,4.6,tcf,27,?,?,?,?,?,?,?,?,good
2,4.5,4.0,?,?,40,?,?,4,?,10,generous,?,half,?,full,good
1,6.0,?,?,38,?,8,3,?,9,generous,?,?,?,?,good
-----

```

Figure 3.7: A snapshot of the labor-neg.data.

Also C4.5 needs a "filename.test" which will contain the examples from the test data set. In this example, Figure 3.8 shows a sample of the labor-neg.test that will be used as input to the C4.5 at the testing phase to test the general performance of the DT model [Quni93].

```

-----
1,4.0,?,?,none,?,none,?,?,yes,11,average,no,none,no,none,bad
2,2.0,3.0,?,none,38,empl_contr,?,?,yes,12,?,yes,none,yes,full,bad
2,2.5,2.5,?,tc,39,empl_contr,?,?,?,12,average,?,?,yes,?,bad
2,2.5,3.0,?,tcf,40,none,?,?,?,11,below average,?,?,yes,?,bad
2,4.0,4.0,?,none,40,none,?,3,?,10,below average,no,none,?,none,bad
2,4.5,4.0,?,?,40,?,?,2,no,10,below average,no,half,?,half,bad
2,4.5,4.0,?,none,40,?,?,5,?,11,average,?,full,yes,full,good
2,4.6,4.6,?,tcf,38,?,?,?,?,?,yes,half,?,half,good
2,5.0,4.5,?,none,38,?,14,5,?,11,below average,yes,?,?,full,good
2,5.7,4.5,?,none,40,ret_allw,?,?,?,11,average,yes,full,yes,full,good
2,7.0,5.3,?,?,?,?,?,?,?,11,?,yes,full,?,?,good
3,2.0,3.0,?,tcf,?,empl_contr,?,?,yes,?,?,yes,half,yes,?,good
3,3.5,4.0,4.5,tcf,35,?,?,?,?,13,generous,?,?,yes,full,good
3,4.0,3.5,?,none,40,empl_contr,?,6,?,11,average,yes,full,?,full,good
3,5.0,4.4,?,none,38,empl_contr,10,6,?,11,generous,yes,?,?,full,good
3,5.0,5.0,5.0,?,40,?,?,?,?,12,average,?,half,yes,half,good
3,6.0,6.0,4.0,?,35,?,?,?,14,?,9,generous,yes,full,yes,full,good
-----

```

Figure 3.8: A snapshot of the labor-neg.test.

The output of the decision tree generator in this example appears in Figure 3.9. After a preamble recording the options used, it shows the constructed DT as well as the simplified DT after pruning. It also shows evaluation results on the training and the testing data before and after pruning.

Options:

File stem <labor-neg>
Trees evaluated on unseen cases

Read 40 cases (16 attributes) from labor-neg.data

Decision Tree:

```
wage increase first year <= 2.5 :
|   working hours <= 36 : good (2.0/1.0)
|   working hours > 36 :
|   |   contribution to health plan = none: bad (5.1)
|   |   contribution to health plan = half: good (0.4/0.1)
|   |   contribution to health plan = full: bad (3.8)
wage increase first year > 2.5 :
|   statutory holidays > 10 : good (21.2)
|   statutory holidays <= 10 :
|   |   wage increase first year <= 4 : bad (4.5/0.5)
|   |   wage increase first year > 4 : good (3.0)
```

Simplified Decision Tree:

```
wage increase first year <= 2.5 : bad (11.3/2.8)
wage increase first year > 2.5 :
|   statutory holidays > 10 : good (21.2/1.3)
|   statutory holidays <= 10 :
|   |   wage increase first year <= 4 : bad (4.5/1.7)
|   |   wage increase first year > 4 : good (3.0/1.1)
```

Tree saved

Evaluation on training data (40 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
12	1 (2.5%)	7	1 (2.5%)	(17.4%) <<

Evaluation on test data (17 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
12	3 (17.6%)	7	3 (17.6%)	(17.4%) <<

(a)	(b)	<-classified as
10	1	(a): class good
2	4	(b): class bad

Figure 3.9: The output of C4.5 on labor-neg data.

C4.5 implements and supports all the features that had been discussed in this chapter such as pruning of decision trees, Information-gain, and gain ratio. In addition, the following are some of the other features that are C4.5 supports [Quni93]:

- **Accounting for unavailable values:** It is an unfortunate fact of life that data often has missing attribute values. This might occur because the value is not relevant to particular case, was not recorded when the data was collected, or could not be decipher by the person responsible for putting the data into machine-readable form. Such a situation can be handle by either a significant proportion of available data must be discarded or the algorithms must be amended to cope with missing attribute values. In most situations the former course is unacceptable as it weakens the ability to find patterns. C4.5 handles such case by replacing a question mark for the missing values and treating only that value for that specific example as ignored attribute [Quni93].
- **Handling different data type input:** The input attributes for C4.5 may have either discrete values such as `days_of_the_week` (Sat, Sun, Mon,...) or they may have continues values such as `age` [Quni93].

- **Rule derivation:** C4.5 also has the capability to produce rule classifier that is usually about as accurate as a pruned tree, but more easily understood by human [Quin93].

CHAPTER IV

Literature Review

4.1 Introduction

A handful of articles reporting the use of some machine learning techniques in the petroleum industry has appeared in SPE (Society of Petroleum Engineers) conferences and related proceedings and publications in the last few years. Some of the articles discuss the use of MLT as a means to analyze formation lithology from well logs while others center on the use of MLT as a

methodology to pick a reservoir model to be used in conventional well test interpretation studies [MAA94].

Notably, however, the reported work is limited to the use of feed-forward neural networks. It is well known that NN suffer the important shortcoming that they are not comprehensible by humans and can only be used as a “black box.” The use of other symbolic machine learning techniques (particularly those based on decision tree learning), which are expected to be more suitable for the intended task, has never been investigated. The following sections highlight a sample of problems in the petroleum industry that have been dealt with using artificial neural networks.

4.2 Permeability Prediction/Estimation

Conventionally, core analysis and well test data interpretations are the most reliable way of acquiring permeability values of a formation. Dependency of rock permeability on parameters that can be measured by well logs (which are much less costly compared to core analysis) has remained one of the most fundamental research areas in petroleum engineering. It has not been possible to capture such dependency satisfactorily by means of conventional computing. Mohaghegh et. al. [MAAN96] suggested the use of neural networks to capture such dependency. Training a three-layer feed-forward neural networks using

geophysical well log data, they have been successful in predicting/estimating the permeability of a highly heterogeneous formation in West Virginia [MBA95].

4.3 Hydraulic Fracturing

Successful prediction of well performance after fracturing is very important for any company, since it can reduce the cost by not performing fracturing on wells that will not show any improvement. Two and three-dimensional models are frequently used for fractal design and monitoring [MBAD96]. Use of these models, however, requires detailed information about rock mechanics and reservoir characteristics, which may be difficult to obtain due to heterogeneties and excessive costs.

Basic well information such as reservoir thickness, porosity, depth, tabular design, initial open flow, offset production, flow tests is usually readily available with no extra costs. This information, however, is generally not usable as engineering data for hydraulic fracture design and post-fracture well performance prediction. Mohagheh et. al. [MAAN96, MA95] followed a hybrid, neuro-genetic approach to optimize fracture design and predict well performance from abundant historical information in a close

geographical area with multiple and varied hydraulic fracture. Their method accepts available data on each well, which includes basic well information and production history, and (as reported in [MBAD96, MHA96]) provides engineers with a detailed optimum hydraulic design, along with the expected post-fracture deliverability [MHA96].

4.4 Estimating PVT Properties of Crude Oil

The importance of PVT properties such as bubble point pressure, solution gas-oil ratio, and oil formation volume factor makes their accurate determination necessary for reservoir performance calculations. Gharbi and Elsharkaway [GE97] present neural network based models for the prediction of these PVT properties. They give a comparative study that suggests that the results predicted by their neural network models are superior to those predicted by other correlation methods [GE97].

4.5 Zone Identification in a Complex Reservoir

One of the key issues in the description and characterization of heterogeneous formations is the distribution of various zones and their properties. White et. al. [WMMA95] presented a study in which several

artificial neural networks were designed and developed for facies identification in a heterogeneous formation from geophysical well logs in Granny Creek Field in West Virginia. Well log data, on a substantial number of wells in this reservoir together with core analysis results from few wells, were utilized to train the networks for zone recognition. According to the reported results, the prediction performance indicated that neural networks could be a useful tool for accurately identifying the zones in the complex reservoir [MAA94, WMMA95].

CHAPTER V

Formation Facies Identification Using Decision Tree Learning

5.1 Introduction

Almost all reservoirs show different degrees of heterogeneity, which make the identification and predication of various geological facies a very complex problem. However, the identification of these facies is a task that the geologists must do since these facies usually influence the hydrocarbon movement, distribution, and management. Observing similarities in well log trace thickness, shapes, and vertical position in geological sequences, expert geologists are often able to predict facies with fair accuracy. They

are also able to give at least tentative geologic interpretation to their findings. Other data besides well log traces are sometimes used to improve the quality of prediction, but traces contain a large amount of information and are used extensively for this purpose [Well88].

The manual approach in facies prediction, even though proven to be doable, is tedious and very time consuming. Therefore, it is highly desirable to reproduce the human reasoning that will be used to identify geological facies from well logs using some automated machine learning approach.

In this thesis, we are going to introduce DTL approach to help geologists identify geological formation from well logs. The DTL method attempts to emulate the manual process done by an expert geologist. Operations of this method shall not require extensive knowledge of geology or an experienced geologist. The main input to the DTL algorithm (DTLA) will be well logs, cored data at the training phase only, and other computed or inferred data such as the geologists' knowledge about the reservoir. The output of the developed DT model will be the geological facies as shown in Table1.1.

In general terms, we have divided the process in this project into the following three phases:

1. Data capturing and preparation phase.
2. DT model development phase.
3. Testing and verification phase.

Figure 5.1 illustrates the abstract view of this process.

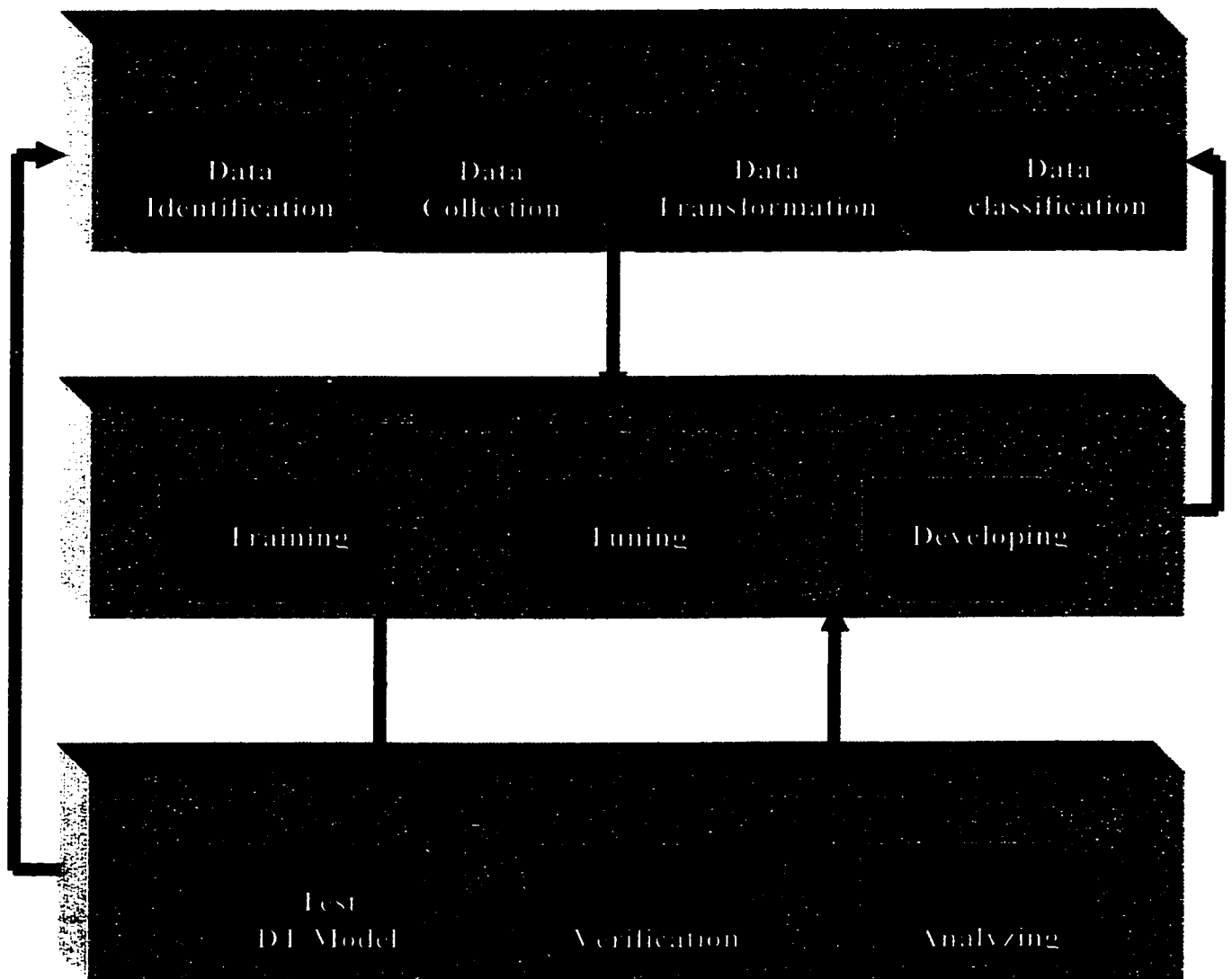


Figure 5.1: An abstract view of the project process

5.2 Data capturing phase

Data collection and interpretation play a very substantial role in the development of DT models to predict formation facies from well log responses. This process requires the integration of several disciplines such as geophysics, structural geology, and log interpretation. The generally low-cored cover leads to an approach, which uses hole logs data to obtain detailed facies information from uncored sections and wells. Therefore, a database of various types of well logs is accessible as a common practice for automatic lithology determination from these well logs. However, the quality of a DT facies-model not only relies on the quantitative data available, but also on the quality of this data [MS97]. Therefore, the performance and the accuracy of the developed DT facies-model adheres directly and very much to the accuracy of the input data presented at the outgrowth stage. The DT facies-model quality thus appears to be complexly controlled by both the "data capturing and interpretation" phase and the "development, training, and tuning" phase. Development of the DT facies-model requires various procedures and coordination with other groups. The following are several series of steps needed to be completed in the first phase of this thesis; Figure 5.2 pictures the conceptual view of this process:

1. Identify which well logs to be used as an input to the DTLA.
2. Identify key wells that will be employed in the development of a DT facies-model.
3. Create a database by gathering well logs, core data, and computing various petrophysical variables that will be used in building the estimation model. Well logs data usually is downloaded from a common well log database.
4. Preprocess and transform the data to place it in the right formats for DTLA.
5. Partition the data into either training data set or testing data set.



Figure 5.2: An abstract view of data capturing process

5.2.1 Determining relevant well-logs

A key activity in the life of a well is the acquisition and good interpretation of well logs data. Different service companies in general supply data obtained at the well site (wireline logs, cores, cuttings, drilling logs, etc.) to the client (geologists and engineers). In other words, the “raw data signals” have to be processed and converted into formation response signals. These steps are performed separately somewhere else, either by the service company supplying the data or by conventional petro-physical analysis systems. Generally, logs are available on all wells drilled into the reservoir. Therefore, they represent the most complete set of reservoir descriptive data [Boom95].

There are tens of various types of well logs that are commonly used for reservoir evaluation. Foretelling which logs would be most substantial for facies description seems to be unrealizable for our case study. It is not known before approaching a problem like this which logs are going to respond to the desired facies to be identified. Thus, not only the quality of data provides a test of how well DTL technology can perform an interpretation of the data, but also for a sound prediction of the facies to be

made between wells, there must be data commonality between all the wells.

In our project, with the assistance of an expert geologist, we distinguishably identify well logs (Table 2.1) to be the most collective and interesting ones for our study. Henceforth, whenever we refer to “essential variables” we mean the three well logs in Table 2.1 plus the depth indicator. As will be demonstrated in our case study, using only these essential variables as input to DTLA does not lead to satisfactory results. Therefore, additional variables will be added to enhance the discrimination between facies. For example, DN_mPOR is an additional variable that represents the difference between the density log (ROHE) and porosity log ($PNLE$) after normalization. These additional variables will be explained in detail later in Section 5.5.

5.2.2 Key-wells identification

Following the naming of the essential well logs that will be used as fundamental input to DTLA; key wells need to be selected to prepare petro-physical database for facies detection. Singling out these wells followed basically four criteria:

1. Representative wells must cover all the existing facies of interest in the field as defined by regional geology.
2. The elected wells must have been cored to allow a comprehensive comparison between the result from geologist analysis and the outcome of the developed DT facies-model.
3. These wells must have a modern logging suite containing at least GR, ROBE, and NPLE.
4. Not only must these wells possess both the well log suite and have been cored, but they must also span the entire reservoir.

A substantial number of wells in the reservoir used in our experimental work were available with complete suite of well log data. Core analysis results were also obtainable for quite a few wells. A set of sixteen wells was picked to train, develop, and test the DT facies-model. Table 5.1 shows the selected wells with their associated region within the field.

REGION	WELLS
Reg1	X0050, X0091, X0101, X0141
Reg2	X0206, X0238, X0246, X0271, X0291, X0301, X0341, X0386, X0441, X0461, X0486, X0526

Table 5.1: A list of the wells and their associated regions that have been used in this project.

Thirteen wells from the sixteen are utilized for training. The input to the C4.5 will be examples that consist of well logs as their attributes and previously determined facies as their classes. The facies are determined at the laboratory using the core samples that had been taken from the wells. Based on the geologist suggestion, we hold on the data from the other three wells to test the DT facies-model constructed at the training phase. These three wells also have the well log data with earlier determined facies from core analysis data. This allows us to compare the results found by the DT facies-model against the core analysis result.

5.3 Data preparation phase

In our study, it has been found that data preparation is a remarkably significant phase to be carefully carried out. The data preparation stage contains two major steps:

1. **Data editing and depth matching:** physo-geologists and possibly well log analysts do these tasks. At this step, log data are examined for spurious reading and edited where possible. Depth matching is known to be fundamental to maximize the correlation between predicted and observed data. This is particularly important for the core data because of its central role in defining the relationships between the data used to

predict facies and the core analysis data. Although we do not perform this step, we must make sure that the quality of the process has been followed correctly. For example, in our data, there was one case where well X0091 had some large GR values which very much indicates to us that there is some noisy data that had not been caught by the data editing process.

2. **DP4FP (data preparation for facies prediction):** This is a C language program that has been written by us to carry out some data groundwork for DTLA. The objectives of this program are as follows:

- **Downloading data:** Download the well log data from the well log database (Model 204 MVS database). For each of the selected key wells, data are downloaded and stored in a stand-alone flat file named with the same name as the well name.
- **Reading and merging data:** Core analysis is done as a separate job at the laboratory. Therefore the core analysis results, which include the facies identified from the core, are stored exclusively from the well log data. Table 5.2 is a sample of the output file that contains the core analysis results. This file consists of the well

name, the depth interval at which the core had been completed, and the associated facies that have been found from the core.

<i>Well name</i>	<i>Core's depth interval</i>	<i>Facies identified</i>
.....
X0091	4320 .. 4390	SS
	4391 .. 4435	SS
	4436 .. 4488	LG

	5412 .. 5440	DOP
.....
X0101	4110 .. 4173	SS
	4174 .. 4255	BB
	4256 .. 4365	SS

	5170 .. 5310	OAP
.....

Table 5.2: A sample of the output file that contains the core analysis results.

It is the responsibility of DP4FP program to read and process both the core analysis output-file, on one hand and the set of all well-logs-data flat files, on the other hand. These files are merged into a single output database that is used later as a source to originate both the training and the testing data sets. The depth interval read from the core analysis file is processed

and outputted at foot-by-foot measurement to match that of well logs output files.

5.4 Data transformation

The main objective of this step is to pull out the data and to transform it and put it in the right format for C4.5 algorithm. Data will be presented to C4.5 as cases and there will be one case for every data point, i.e., for every one-foot depth in the format shown in Section 3.5 . Each case is concerned with the values of the well logs and the core-facies measured at that specific point of depth.

5.5 Computing and adding additional variables:

In some domains, a classifier can not easily capture the relationship between variables and classes. Therefore, for a classifier to recognize a pattern between the variables and the classes, additional variables are passed to it as well. In our case, such auxiliary variables could be inaugurated for example by computing the derivative of some logs, taking the average over certain depth intervals, or the difference between two of the essential parameters.

One of the most significant tasks that DP4FP performs is to compute additional parameters out of the essential ones. We found, as will be seen from the cases studied, that the DTLA (C4.5) can not recognize the various facies satisfactorily if only GR, ROHE, PNLE, and depth indicators are provided as input. Therefore to avoid this problem, for every data point we invented and introduced new variables. These new variables that were added either had been computed from the essential ones or symbolize some kind of known geological facts about the reservoir.

Note that these additional variables can not be haphazardly guessed at or added. A wrong addition of variables might lead into a construction of a more confused tree that will produce less accurate predictions. The course of discovering and deciding on these variables has consumed a large portion of the DT facies-model development time, because for every additional variable used as an input to C4.5; one must do the following tasks:

1. Modify DP4FP to compute this new parameter and then run it to construct both the training and testing data sets to output this computed variable into these data sets.

2. Run C4.5 and check how this variable influences the final size and the accuracy of the assembled DT.
3. Test the produced DT on the test dataset (unseen examples) and compare its results with and without the addition of this variable.
4. Decide whether to encompass that variable as an input parameter or declare it as an "ignored " one. Ignored input variables to C4.5 are not used to build the DT model, which means that the variable will have zero contribution in the final classifier.

Table 5.3 lists the substantial additional variables that were computed or added to help C4.5 captures the main relation between the input and the output.

<i>ADDED VARIABLE NAME</i>	<i>DESCRIPTION</i>
GR_bd	GR[previous depth] - GR[current depth]
GR_ad	GR[next depth] - GR[current depth]
ROHE_bd	ROHE[previous depth] - ROHE[current depth]
ROHE_ad	ROHE[next depth] - ROHE[current depth]
PNLE_bd	PNLE[previous depth] - PNLE[current depth]
PNLE_ad	PNLE[next depth] - PNLE[current depth]
GR_m_DEN	Difference between GR and ROHE after normalization
GR_m_POR	Difference between GR and PNLE after normalization
DEN_m_POR	Difference between ROHE and PNLE after normalization
OAP_or_DOP	Possibility that this is OAP or DOP area
Pos_lico	Possibility that this is LICO area
Region	Sector the whole field into three regions
GRM	Cal. mean of the GR for estimated vertical interval.
DENM	Cal. mean of the ROHE for estimated vertical interval.
PORM	Cal. mean of the PNLE for estimated vertical interval.

Table 5.3: A List of the substantial new variables that had been computed or added

5.6 Training phase

After the collection and the transformation of the data, two subsets should be extracted from it, the learning subset and the validation one. The learning subset is utilized to train C4.5 to construct the DT facie-model. The collection of training data was based on the five criteria that have been mentioned in Section 3.2.

1. Data collected for thirteen wells (X0050, X0091, X0141, X0238, X0246, X0271, X0291, X0301, X0341, X0386, X0441, X0461, and X0486) were utilized for the system training. In this process, the C4.5 was provided with log data and additional computed parameters as the input. It was also provided with the definition of the various facies found from the core.
2. Data is presented to C4.5 as one training record per one-foot of depth. These records are formatted as shown in Table 5.4.

Depth	Essential Variables			Added Variables						Class
	E1	E2	E3	A1	A2	A3	A4	.	A _n	
5000	23.25	3.22	0.254	0.86	0.03	12.2	3	.	N	SS
5000	23.25	3.22	0.254	0.86	0.03	12.2	3	.	N	SS
-	-	-	-	-	-	-	-	.	-	-
-	-	-	-	-	-	-	-	.	-	-
-	-	-	-	-	-	-	-	.	-	-
6119	34.67	2.44	0.31	0.59	0.04	14.5	1	.	Y	DOP
6120	35.01	2.34	0.289	0.62	0.04	13.9	1	.	Y	DOP

Table 5.4: A sample of input data used to train C4.5

- At this point, the C4.5 builds a DT model that is hoped to be capable of predicting the output from a given set of input for unseen examples.

During this phase, one must bear in mind the fact that the simpler the constructed DT, the better the generalization. Therefore, a revisit and tuning of the input data seems to be a must. As you will see from the cases studied, the tuning and the massaging of the input data can be achieved by deciding on the right set of input variables. The main objective of the variable selection is to obtain a DT model that has a better performance based on the selected input parameters.

5.7 Model Verification Phase

The main objective of this step is to evaluate the performance of the DT model over new unseen data or wells. However, this phase is very important to determine whether further training data needs to be collected to build the DT model or maybe a different way of training needs to be done. The testing phase is conducted as follows:

1. The data collected from three wells (X0101, X0206, and X0526) were utilized for testing the DT model. The geologist suggested these three wells to be retained to test the accuracy and the performance of the developed DT model.
2. In this phase the C4.5 was provided with only log data and additional computed parameters when available.
3. Although core-facies definitions were accessible for all the examples in the test data set, no definition of the various facies were provided to the constructed DT model. At this stage no updating to the constructed DT model is done. The facies predicted by the DT model is compared against the core-facies. Table 5.5 shows a snapshot of some data for well X0101 that has been used in testing phase.

Depth	Essential Variables			Added Variables					
	E1	E2	E3	A1	A2	A3	A4	.	A _n
4612	33.53	5.22	0.314	0.99	0.07	15.1	2	.	N
4613	23.25	3.22	0.331	0.89	0.06	14.9	2	.	N
-	-	-	-	-	-	-	-	.	-
-	-	-	-	-	-	-	-	.	-
-	-	-	-	-	-	-	-	.	-
5877	26.33	4.02	0.23	0.74	0.05	13.5	3	.	Y
5878	26.55	3.98	0.30	0.73	0.06	15.9	3	.	Y

Table 5.5: A snapshot of some data for well X0101 that has been used in testing phase

4. Just as in the training phase, data was presented to C4.5 for every one-foot depth.
5. The DT model tests one testing record at a time. For each record, it will try to predict the facies from the given set of input parameters.

Once all the examples in the test dataset are examined, we start evaluating the final results of the DT model. This is done by comparing the predicted facies from the DT model with the facies known from the core at foot-by-foot basis. The goal of the evaluation is to achieve a DT model's error that is relatively small and acceptable. A close study of facies that have been mismatched and where these mismatches occur could benefit us in doing the following tasks:

- Adding additional input variables such as incorporating some kind of known geological facts about the reservoir and presenting these facts as input parameters to C4.5. This helps in building a DT model that makes better predictions at the testing phase.
- Removing variables with little or no influence on the outcome.

This is done with the main objective in mind, which is to achieve the best facies prediction. Again, this is consummated in a way such that the last measured error in the DT model, over the test data, is relatively low and reasonable. An iterative process by going between the training phase and testing phase is necessary to fulfill this goal.

5.8 Case study

This thesis details experiments done on a real oil reservoir field. The formation in this field is coarsening upward skeletal limestone that ranges from 390 to 450 feet in thickness. The lower one-half to two-thirds of the section is dominated by mudstones grading upwards into mudstones to wackestones and finally into wackestones to packstones. The field is a

massive fossilized carbonate structure, which owes its origin to the activities of a multitude of marine organisms.

A distinct advantage in petroleum geology is the ability to predict sedimentary facies. To reduce the possibility of an undetected facies between any two studied wells, the wells should be sufficiently closely spaced. Sample spacing of the cored section should also be as close as practically possible. The data presented in this study have been gathered from 16 wells, thirteen of which are used for training and the other three wells, for testing. Both core and logs data were obtained for all the sixteen wells.

The wells that had been utilized for training were X0050, X0091, X0141, X0238, X0246, X0271, X0291, X0301, X0341, X0386, X0441, X0461, and X0486, while X0101, X0206, and X0526 were used for model verification as suggested by geologist. The above mentioned information holds true for all the following six-step case study unless stated otherwise. 2667 examples were utilized for training and 681 unseen examples were used for testing.

5.8.1 Case Study: Initial Step

Gamma ray, bulk density, and the neutron porosity logs are the most commonly used well logs by geologists to identify geological facies. A highly experienced geologist who is familiar with well logs is usually consulted to identify the association between these three well logs and the core result, on one hand, and the facies, on the other hand. The goal of the first step of this case study was to explore whether using only these essential logs (Table 2.1) as input to C4.5 can lead to good predication of geological facies. The input to the C4.5 was only the essential well log variables as shown in Table 5.6.

Essential input parameters			
GR	ROHE	NPLE	DEPIND

Table 5.6: Input to the C4.5 for step 1.

Table 5.7 summarizes the evaluation of the training and the testing results of the initial step.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	663	2538	129	4.8	493	2487	180	6.7
Testing	663	336	345	50.7	493	353	328	48.2

Table 5.7: A summary of training and testing results for step1.

Table 5.8 shows for each facies how many of a specific facies occurs and how closely it has been predicted. For example, there are 123 cases of type LG facies and 90 of them were classified correctly. Whereas, 30 cases were misclassified as SS and 3 were misclassified as BB. The larger the values on the diagonal of that matrix the better the predication and we prefer the distribution of the wrongly classified facies to be as close to the diagonal as possible.

	SS	BB	LG	LICO	OAP	DOP
SS	54	37	83	14	0	0
BB	9	86	18	0	0	0
LG	30	3	90	0	0	0
LICO	38	12	6	82	1	0
OAP	15	1	0	20	35	3
DOP	1	2	0	26	9	6

Table 5.8: Confusion matrix for the test data evaluation in step 1.

The results of this experiment indicated to us that using the DT model that had been assembled using only the essential input variables at the training

phase is insufficient to recognize the facies for the test wells. To alleviate this problem, we thought of introducing additional input variables to be passed to the DTLA. This led us to the next step in our case study.

5.8.2 Case study: Step Two

In the conventional approach, the geologist visually and manually inspects well log curves and depends on past experiences to describe facies. In this process, the geologist does not inspect these curves based on foot-by-foot approach. However, they bear in mind and inspect these curves interval-by-interval. To somehow mimic the geologist in doing so, we had to think of a way that can tell the DTLA the values of GR, ROHE, and PNLE for a certain interval above and below the current point. For example, let us assume that the current example that had been passed to DTLA is at a depth of 405 feet and that the essential input to C4.5 for that example is the record 405, 26, 3.0, and 0.25 that correspond to DEPIND, GR, ROHE, and PNLE, respectively. Then if we assume further that the interval we would like to assemble consists of five points above and five points below the current point, we add two new variables that are GR_da and GR_db that are calculated as shown in Figure 5.3.

```

GR_da:
Total_value = 0.0;
FOR I = current depth + 1 TO current depth + interval size DO
    Val = abs( GR[curent_depth] - GR[I] );
    Total_value = Total_value + val;
END_FOR
GR_da = GR[current_dpeth] - Total_value / interval size;

GR_db:
Total_value = 0.0;
FOR I = current depth - interval size TO current depth - 1 DO
    Val = abs( GR[curent_depth] - GR[I] );
    Total_value = Total_value + val;
END_FOR
GR_db = GR[current_dpeth] - Total_value / interval size;

```

Figure 5.3: Calculating GR_da and GR_db.

The same procedure is used to calculate ROHE_da, ROHE_db, PNLE_da, and PNLE_db. Training and testing in the same data used in the initial step, we started examining the introduction of these new six computed (GR_da, GR_db, ROHE_da, ROHE_db, NPLE_da, NPLE_db) variables. Our findings were as follows:

- We noted that the degree of change in the Neutron Porosity Limestone (NPLE) log was very minimal. Therefore, there was no real contribution to the outcome from using NPLE_db and NPLE_da.
- Also, the experiment showed that when including ROHE_ad as input to C4.5 the size of the constructed DT was larger with very or no contribution to final result. Therefore only ROHE_db was considered between the two computed parameters from the ROHE log.
- Both GR_da and GR_db had some benefit and helped in reducing both the size and the error of the built DT.

The inputs to the C4.5 were the essential well log variables and some computed new parameters as shown in Table 5.9.

Essential input variables				Computed added variables		
DEPIND	GR	ROHE	NPLE	GR_da	GR_db	ROHE_db

Table 5.9: Input to the C4.5 for step 2.

Tables 5.10, 5.11, and 5.12 illustrate the findings and the results of this step for both the training and the testing phase.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	423	2591	76	2.8	379	2597	88	3.3
Testing	423	371	310	45.5	379	372	309	45.4

Table 5.10: A summary of training and testing results for step2.

Initial step best result		Current step best result		Gain	
Tree size	Error %	Tree size	Error %	Tree size reduction	Error %
493	48.2	379	45.4	114	2.8

Table 5.11: Comparison between the Initial step and the current step.

	SS	BB	LG	LICO	OAP	DOP
SS	64	23	95	6	0	0
BB	17	78	12	6	0	0
LG	34	1	88	0	0	0
LICO	36	12	11	79	1	0
OAP	3	1	3	18	47	2
DOP	2	0	0	17	9	16

Table 5.12: Confusion matrix for the test data evaluation for step 2.

The evaluations of the constructed DT model over the unseen test examples showed accuracy of around 55 %. This is not a satisfying result and more work needs to be done to achieve better performance. After talking with the geologists, we knew that it is not practical that well logs be

examined and inspected independently of each other. This in itself led us to the next step in the study case.

5.8.3 Case study: Step Three

In this step we introduced the following three input computed variables:

Variable name	Description
GR_m_DEN	GR[current depth] - ROHE[current depth]
GR_m_POR	GR[current depth] - NPLE[current depth]
DEN_m_POR	ROHE [current depth] - NPLE[current depth]

Table 5.13: Three new computed variables for step 3

The aim of introducing these three variables was again to mimic what the geologists do when they inspect GR, ROHE, and NPLE well log curves simultaneously. Geologists not only examine the degree of fluctuation on the well log curves, but also, they overlook the degree of separation and closeness of these well log curves to each other's when drawn to the same scale (Figure 5.4).

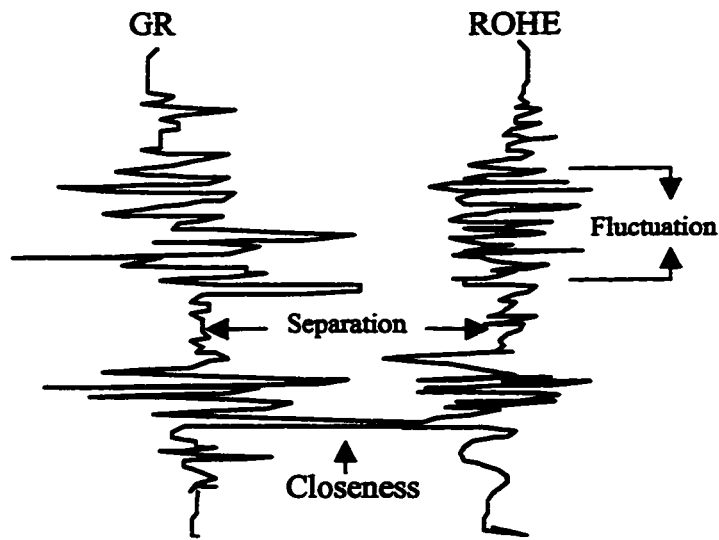


Figure 5.4: An illustration of Fluctuation, Separation, and Closeness that are examined by geologist.

Our experiment with the above stated variables showed the following:

- The GR_m_POR and DEN_m_POR parameters contributed nothing or very little to the final outcome of the DT model.
- However, when GR_m_DEN was used as additional input to DTLA, a little bit more accurate DT model was constructed.
- Therefore we decided to include only GR_m_DEN as additional input to DTLA.

The inputs to the C4.5 were the essential well log variables and some computed new parameters as shown in Table 5.14.

Essential input variables				Computed added variables			
DEPIND	GR	ROHE	NPLE	GR_m_DEN	GR_da	GR_db	ROHE_db

Table 5.14: Input to the C4.5 for step 3.

After deciding on the above-mentioned input parameters for C4.5 we performed training over the same examples from the initial step of case study. Tables 5.15, 5.16, and 5.17 illustrate the findings and the results of this step.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	445	2590	77	2.9	401	2585	82	3.1
Testing	445	380	301	44.2	401	387	394	43.2

Table 5.15: A summary of training and testing results for step3.

Initial step best result		Current step best result		Gain	
Tree size	Error %	Tree size	Error %	Tree size reduction	Error %
493	48.2	401	43.2	92	5.0

Table 5.16: Comparison between the Initial step and the current step.

We notice in this step that the tree size has slightly increased over the previous step (Step2). However, the accuracy had little improvement, around 2.5 % from the previous step and by accumulative of 5.0 % from the Initial step.

	SS	BB	LG	LICO	OAP	DOP
SS	72	32	78	6	0	0
BB	11	81	15	6	0	0
LG	28	1	94	0	0	0
LICO	33	15	8	82	1	0
OAP	3	1	3	21	42	4
DOP	2	0	0	21	5	16

Table 5.17: Confusion matrix for the test data evaluation for step3.

The accuracy of DT model is still around 57 percent, which is not yet satisfactory. However, the gain in the outcome thus far over the initial step's result indicated to us that we are on the right track. Thus far we had not included any geologist's known facts about the reservoir that could be used when doing the evaluation to recognize formation facies. For example, a fact such as a specific facie might not ever occur above certain depth indicators. The understanding and use of these known facts led us to the next step in the study case.

5.8.4 Case study: Step Four

Geologists usually possess a general knowledge about their reservoirs. And they use this knowledge as well-known facts to help them evaluate their reservoirs. For example, geologists would use the fact that if facies OAP or DOP presents in the studied well, then these two facies most likely would occur at 85 percent or more from the top of the formation.

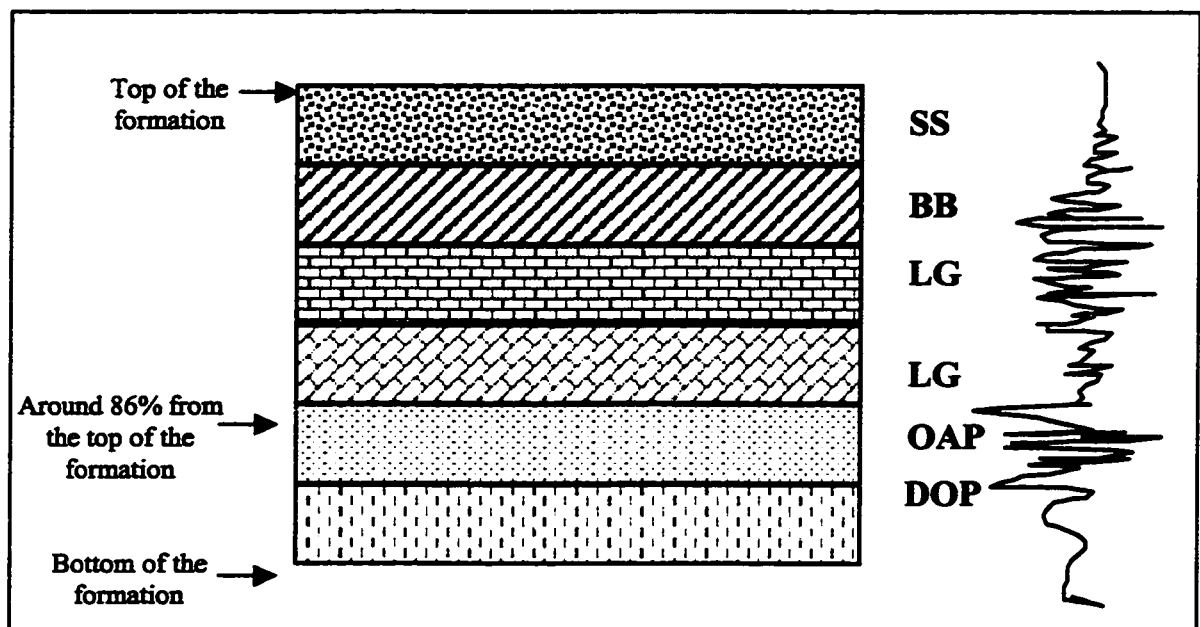


Figure 5.5: An illustration of the fact that OAP and DOP facies might always occur at depth that is a round 85 % or more from top of the formation.

To capture this kind of knowledge, we presented these facts as input variables to the DTLA. For example, a new variable had been added which is OAP_or_DOP. This variable will stand for the possibility that the

specific point-of-depth for this case is of type OAP or DOP facies. A value of "Y" or "N" is set for OAP_or_DOP based on the following criteria:

$$\text{OAP_or_DOP} = \begin{cases} \text{Y} & \text{For all the examples occurring at 85 \% or more from top of the formation} \\ \text{N} & \text{Otherwise} \end{cases}$$

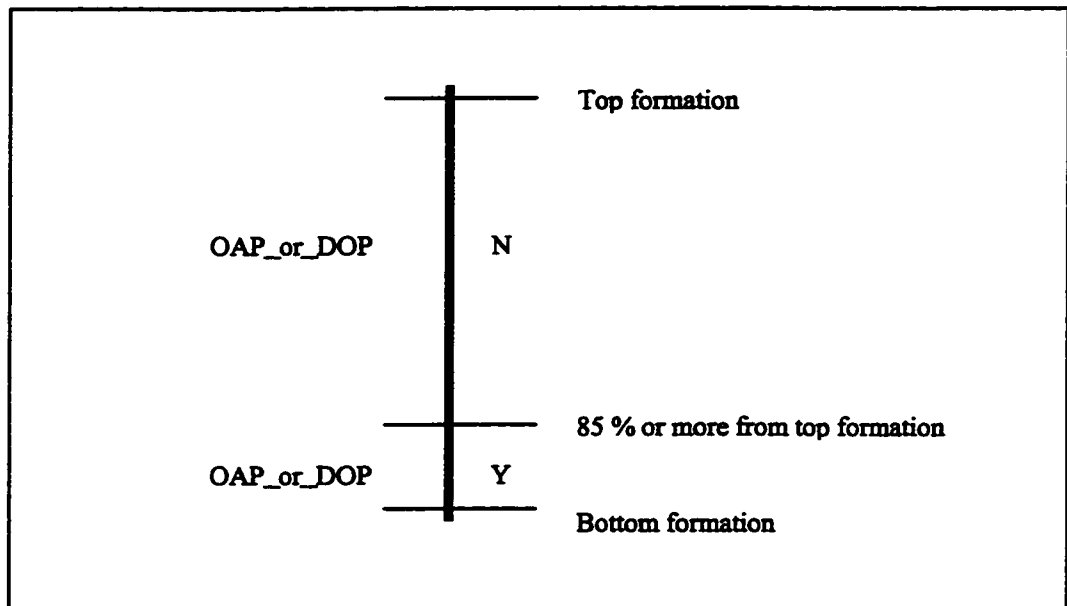


Figure 5.6: An estimated location of OAP and DOP facies from the top of the formation.

The aim of this step was to cross-examine the ability of capturing the geologist's wisdom about the reservoir and to present it as an input

variable to the DTLA. Tables 5.18, 5.19, 5.20, and 5.21 illustrate the results of adding two new parameters of this type.

Essential input variables					
DEPIND	GR		ROHE	NPLE	
Computed added variables					
OAP_or_DOP	Pos_lico	GR_m_DEN	GR_da	GR_db	ROHE_db

Table 5.18: Input to C4.5 for step 4.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	299	2611	56	2.1	261	2604	63	2.4
Testing	299	429	252	37.0	261	427	254	37.3

Table 5.19: A summary of training and testing results for step 4.

Initial step best result		Current step best result		Gain	
Tree size	Error %	Tree size	Error %	Tree size reduction node	Error % reduction
493	48.2	299	37.0	194	11.2

Table 5.20: Comparison between the Initial step and the current step.

The tree size has reduced dramatically from the previous step (Step3). And the accuracy had improved by around 6.2 percent from the previous step and by an accumulative 11.2 percent from the Initial step.

	SS	BB	LG	LICO	OAP	DOP
SS	49	41	93	5	0	0
BB	10	90	10	3	0	0
LG	31	1	91	0	0	0
LICO	3	27	3	105	1	0
OAP	0	0	1	5	65	3
DOP	1	0	0	0	16	27

Table 5.21: Confusion matrix for the test data evaluation for step 4.

At this stage, the over all accuracy of the DT model is around 63 percent, which is not yet satisfactory. However, we knew from the beginning that even the geologists, when studying their field, prefer to sector the reservoir into multiple regions based on the characteristics of each section of the field. Thus far we had not used this fact, aiming that maybe DTLA can recognize the entire field as just one object. To make use of this knowledge, we started our next step.

5.8.5 Case study: Step Five

Most of the time a reservoir can not be studied in its entirety as one region. This is true if the reservoir under consideration is huge and highly heterogeneous. The field that we are working with is a very large and heterogeneous oil field. Therefore, we had sectored our field into three regions and associated each well with one of the three regions. We have

introduced to the DTLA a new input variable which is "region" with the following values:

region = $\left\{ \begin{array}{l} 1 \text{ For all examples from wells associated with region one.} \\ 2 \text{ Otherwise} \end{array} \right.$

Table 5.22 shows the set of input variables for DTLA at this step.

Essential input variables						
DEPIND	GR		ROHE		NPLE	
Computed added variables						
region	OAP_or_DOP	Pos_lico	GR_m_DEN	GR_da	GR_db	ROHE_db

Table 5.22: Input to C4.5 for step 5.

Tables 5.23, 5.24, and 5.25 illustrate the findings and the results of this step after deciding on the above-mentioned input variables for C4.5.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	233	2631	36	1.3	177	2618	49	1.8
Testing	299	585	96	14.1	177	590	91	13.4

Table 5.23: A summary of training and testing results for step 5.

Initial step best result		Current step best result		Gain	
Tree size	Error %	Tree size	Error %	Tree size reduction node	Error % reduction
493	48.2	177	13.4	316	34.8

Table 5.24: Comparison between the Initial step and the current step.

We notice in this step that the tree size was reduced even further from the previous step (Step4). The accuracy had improved dramatically, approximately 23.6 % from the previous step and by the accumulative 34.8 % from the initial step.

	SS	BB	LG	LICO	OAP	DOP
SS	188	0	0	0	0	0
BB	0	99	11	3	0	0
LG	0	19	104	0	0	0
LICO	0	32	1	105	1	0
OAP	0	0	0	4	67	3
DOP	1	0	0	0	16	27

Table 5.25: Confusion matrix for the test data evaluation for step 5.

The over all accuracy of the DT model at this stage is around 87 percent, which is indeed a satisfactory one. Most of the large values fall on the

diagonal of the matrix. The values in bold-face in the matrix indicate that this is acceptable error, i.e. since confusing these two facies do not have real consequence on the geologists' and the engineers' final decisions on maintaining their reservoir. Therefore, we can think that the accuracy of this DT model is around 90 % and this is considered an excellent result.

5.8.6 Case Study: Step Six

In this step we utilized an existing locally written program that reads through the data once and estimates breaking points between facies. Utilizing this predication of braking points, we took the estimated interval between every two consecutive breaking points and did the following:

1. We introduced three new variables which are GRM, ROHEM, NPLEM.
2. These variables are calcuated for every estimated interval "i" as follow :

GRM[I] = the mean of GR for all the examples of interval "i"

ROHEM[I] = the mean of ROHE for all the examples of interval "i"

NPLEM[I] = the mean of NPLE for all the examples of interval "i"

The final inputs to the C4.5 for this step is as shown in Table 5.26.

Essential input variables							
DEPIND		GR		ROHE		NPLE	
Computed added variables							
OAP_or_DOP		Pos_lico	GR_m_DEN	GR_da	GR_db	ROHE_db	
GRM	ROHEM	NPLEM			region		

Table 5.26: Final input to C4.5 for the final step.

Tables 5.27, 5.28, and 5.29 illustrate the findings and the results of this step.

Results	Before pruning				After pruning			
	Tree size	Correct cases	Missed cases	Error %	Tree size	Correct cases	Missed cases	Error %
Training	95	2659	8	0.3	91	2657	10	0.4
Testing	95	621	60	8.8	91	621	60	8.8

Table 5.27: A summary of training and testing results for step 6.

Initial step best result		Current step best result		Gain	
Tree size	Error %	Tree size	Error %	Tree size reduction node	Error % reduction
493	48.2	95	8.8	398	39.4

Table 5.28: Comparison between the Initial step and the current step.

The tree size was reduced even further from the previous step (Step5). And the accuracy had improved little by around 4.6 % over the previous step and by accumulative of 39.4 percent from the initial step.

	SS	BB	LG	LICO	OAP	DOP
SS	188	0	0	0	0	0
BB	0	95	18	0	0	0
LG	0	2	121	0	0	0
LICO	0	0	24	115	0	0
OAP	0	0	0	3	66	5
DOP	1	0	0	0	7	36

Table 5.29: Confusion matrix for the test data evaluation for step 6

The overall accuracy of the DT model at this stage is around 91.2 percent. Again the values in bold-face in the matrix indicate that these are acceptable error and this will bring the final accuracy of this DT model to around 93 percent accurate and this is considered an excellent result. For final and full detail of the output of the C4.5 for the case study, see Appendix A.

CHAPTER VI

Verification of the Final DT Model

6.1 Introduction

The data presented in the previous six steps of the case study that we discussed and carried out in Chapter 5 have been gathered from 16 wells. For all the six steps, the thirteen wells that had been utilized for training were X0050, X0091, X0141, X0238, X0246, X0271, X0291, X0301, X0341, X0386, X0441, X0461, and X0486, while X0101, X0206, and X0526 were used

for model testing as suggested by a geologist. However, to further test and verify the generalization and the quality of the DT model, we decided to do a cross-validation-like procedure. This procedure involves doing the following tasks:

- In every distinct verification test, we randomly picked thirteen wells to be utilized for training the DTLA.
- We use the remaining three wells for testing the DT model that had been built at the training phase.
- In all the tests that were conducted, the DT model was supposed to predict the same facies that were predicted in the previous case study (Table 1.1).
- In all the tests, we fixed the set of variables that would be used as an input to the C4.5. The final set of variables that evolved from the previous steps in the case study is used as standard inputs to the C4.5 (Table 6.1).

Essential input variables					
DEPIND	GR		ROHE	NPLE	
Computed added variables					
OAP_or_DOP		Pos_lico	GR_m_DEN	GR_da	ROHE_db
GRM	ROHEM	NPLEM		region	

Table 6.1: The standard set of input needed to develop DT model.

The main objective of these tests was to prove that the process used to develop the DT model was general enough. Another aim of this experiment was to show that the final set of input variables that we came up with was crucial to train the DTLA to mature the effectiveness of the DT model in predicting the facies.

6.2 Summary of the ten different tests

Tables 6.2, 6.3, and 6.4 summarize the findings of the ten different tests conducted to prove the generalization performance of our DT model. Table 6.2 lists the testing wells that had been used for every test, while Table 6.3 presents the evaluation results of DT model with respect to the unseen data before and after pruning for each test. Table 6.4 is an average calculation of how many of a specific facies occurred and how closely it has been predicted.

TEST NUMBER	TESTED WELLS
1	X0091, X0386, and X0461.
2	X0141, X0291, and X0486.
3	X0206, X0246, and X0486.
4	X0050, X0461, and X0486.
5	X0141, X0441, and X0526.
6	X0101, X0291, and X0461.
7	X0206, X0271, and X0341.
8	X0091, X0238, and X0441.
9	X0050, X0246, and X0271.
10	X0101, X0246, and X0461.

Table 6.2: A list of the tested wells for each test.

Test Number	Before pruning			After pruning		
	Tree size	Misclassified examples	Error %	Tree size	Misclassified examples	Error %
1	87	87	12.8	77	91	13.4
2	105	64	9.7	97	64	9.7
3	121	85	13.0	113	84	12.9
4	91	74	10.6	83	74	10.6
5	95	124	17.1	89	140	19.3
6	103	54	8.1	103	54	8.1
7	87	115	18.6	77	115	18.6
8	111	123	17.5	95	121	17.2
9	109	113	17.9	87	114	18.0
10	105	88	13.4	89	87	13.2
Average	101.4	92.7	13.87	91.0	94.4	14.1

Table 6.3: The evaluation results of DT model with respect to the unseen data before and after pruning for each test.

The average accuracy of the DT model for the ten tests is around 86.35 %, which is a satisfactory result. And as can be seen from Table 6.4, most of the large values fall on the diagonal of the matrix. The values in bold-face in the Table indicate that these are acceptable error and this will bring the final average accuracy of the DT model to around 89.0 % accurate and this is an excellent result.

	SS	BB	LG	LICO	OAP	DOP
SS	149.6	0	0	0	0.6	0.3
BB	0	123.4	9.6	6.9	0	0
LG	0	24.3	115.7	5.8	0	0
LICO	0	7.7	3.5	112.1	13.7	0
OAP	0	0	0.1	3.5	63.4	10.2
DOP	0.2	0	0	0	7.3	28.9

Table 6.4: The average confusion matrix for the test data evaluation.

CHAPTER VII

Summary and Conclusion

7.1 Summary

The main goal of this study was to examine how effective the DTL technique could be used as a tool for geological facies recognition from well logs. Although, the conventional manual approach that the geologist follows to identify facies is doable and gives a satisfactory accuracy, this approach is tedious and time consuming. This task is quite challenging and requires a highly experienced geologist to handle it. And given the fact that abundance of historical data of well log interpretation is usually

kept, we had considered machine learning methods as an alternative approach to solve such a problem. Therefore, we had developed a DT model using the C4.5 software package to explore the effectiveness of predicting formation facies from well log responses in a real field. The facies of the reservoir indicated a high degree of heterogeneity in the formation. Both core and logs data were obtained for sixteen wells. Thirteen wells were utilized to train and construct the DT model, while three wells were segregated at the beginning of this study as indicated by the geologist for the purpose of testing the constructed DT model. The essential input variables for C4.5 were gamma rays, bulk density, and neutron porosity logs together with the depth indicator.

We conducted a six-step case study in this thesis. The initial step was to explore whether using only the above three essential logs (Table 2.1) as input to C4.5 can lead to good prediction of formation facies. Our evaluation of the test-data results indicated that using only the essential input variables could not satisfactorily predict the facies. To alleviate this

problem, we introduced additional input variables to the DTLA. These new input variables are inferred from the essential ones. In the other five steps we examined and imprinted how the gradual introduction of these variables helped the DTLA to capture more patterns about the input and made it more accurate in making the right decision. This success, however, is subjected to the following conditions:

- Enough key-wells with good core cover must be chosen to define all the facies to be encountered in the field. The input to the C4.5 must contain the full range of facies types in reasonable quantity.
- Relatively complete sets of high quality logs must be used.
- Careful data preparation is vital to the success of this technique.

7.2 Conclusions and major contribution

The following are some of the conclusions that were reached in this study and few humble contributions:

- This study established the fact that a formation facies prediction from well-logs data is feasible using the decision tree learning approach. All

previous work had investigated the use of neural networks in this kind of problem. No one had explored the use of decision tree learning technique to solve such a complicated problem.

- We proved that the use of the essential well log variables alone (Table 2.1) did not lead into good construction of a DT model that is effectively capable to predict formation facies from the essential well logs.
- We progressively identified and examined new input variables that are inferred from the essential ones. With the introduction of these new additional variables to C4.5, we gradually alleviate the problem stated above.
- We finally identified and came to the conclusion that the set of input variables shown in Table 6.1 is the final standard input for C4.5. Such an input set led into production of an effective DT facies model.
- We built a DT model that has an average accuracy of 86.35 % with tree size of approximately one hundred nodes. Taking into consideration the “acceptable error”, which is an error that does not have any

consequence in the geologist decision, then the final average accuracy of the DT model was around 89 %.

- We conducted 10 different tests to prove the generality of the DT model. In each of these tests, we randomly picked three wells to test the general performance of the DT model, whereas the rest of the wells were used in the training phase.
- We captured the geologist's knowledge and known facts about the reservoir and presented them as input parameters to C4.5.
- Our experience with the design and the development of the DT model showed that it is fundamental to have enough data to train the DTLA properly in order to achieve acceptable results. We believe that we should have at least three times the amount of data we had in order to achieve comfortable and free training and testing for this specific problem. We also believe that the high non-uniformity of the wells and the limited amount of data made the training and testing quite complex.

- We believe that not only would this tool help the geologist in predicting facies, but also it would save a considerable amount of time of geological formation analysis. And with relatively less need for core measurement information, this tool will cut on cost as well as in time in the process of facies evaluation.
- We have done all of our works using real-world data from a real oil field.
- Figure 7.1 summarized the results of the six steps for the case study. It also illustrates how the progressive addition of the new variables helped in maturing the DT model to achieve better results. In addition, for this specific case study, it shows the reverse relationship between the size of the constructed tree and the accuracy of the DT model. As the accuracy of the DT model improved the size of the DT is reduced.

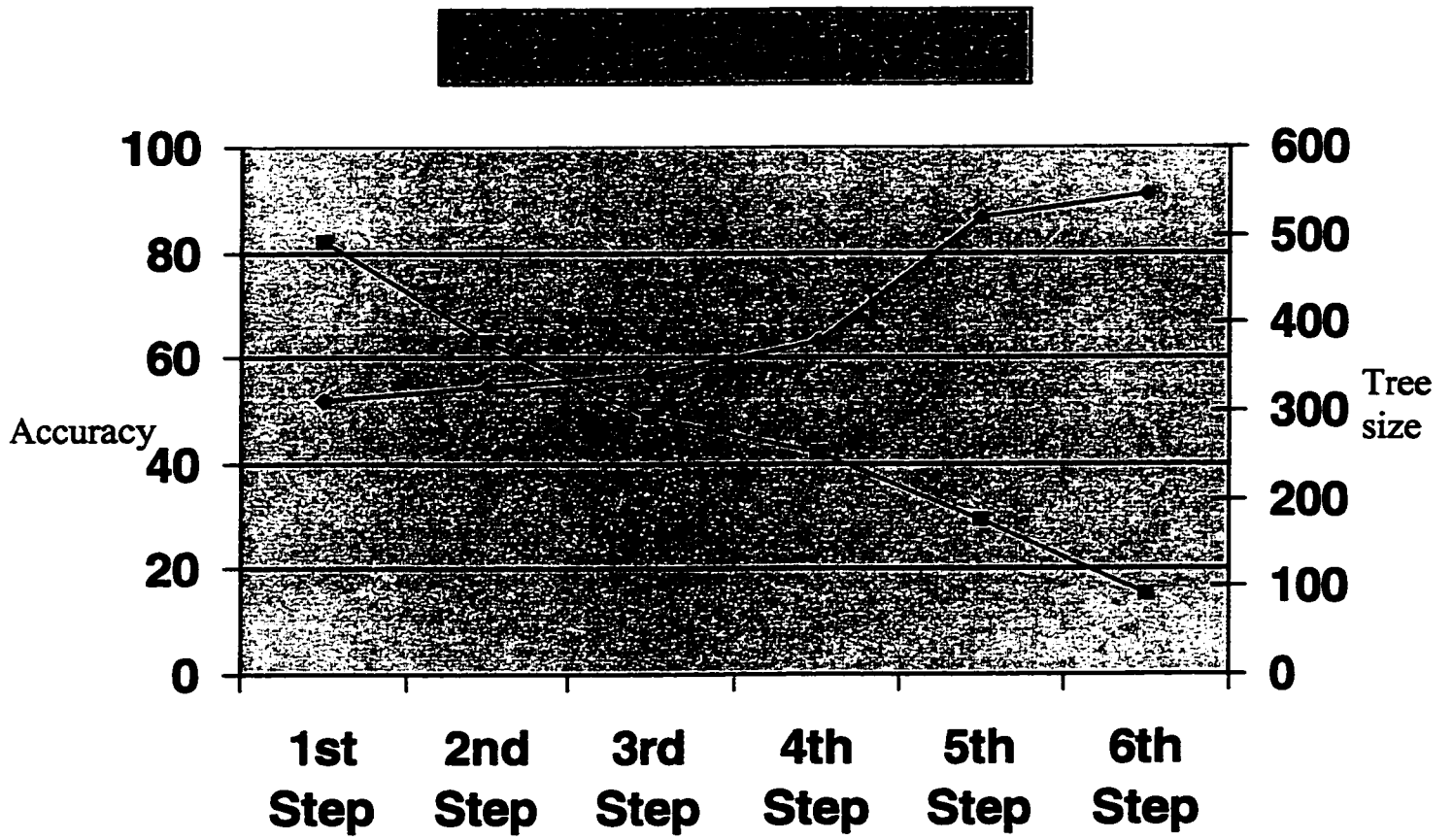


Figure 7.1: A summary of the results of the six-step case study

- Figure 7.2 summarized the results of the ten different tests.

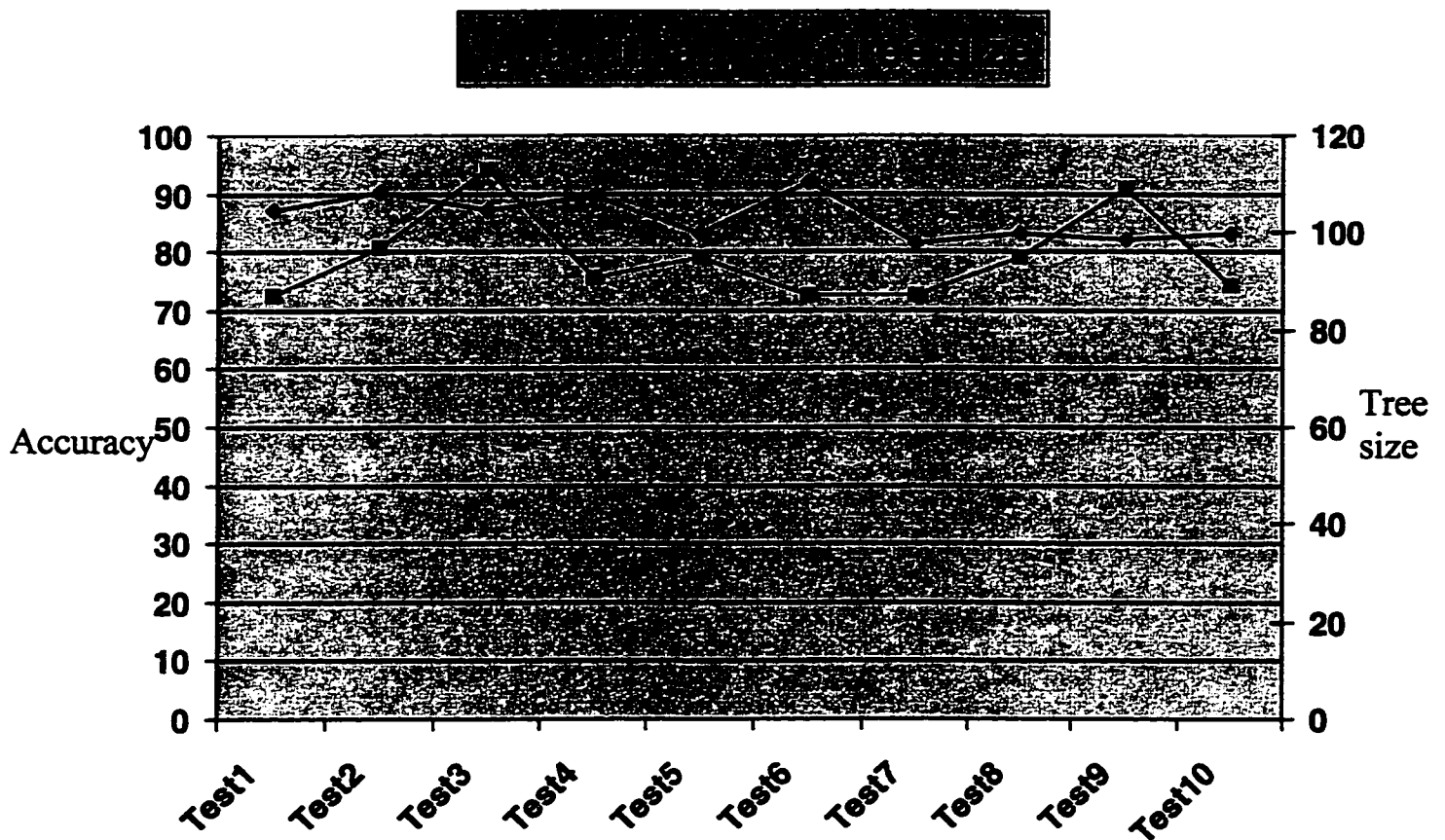


Figure 7.2: A summary of the results of the ten different tests

7.3 Future work

Although this study established the fact that formation facies prediction from well logs data is attainable using the decision tree learning approach

with an average accuracy of 89 %, this was only an initial study to evaluate the potentiality of this methodology. The following items are worth investigation for building conceivably more accurate DT models:

- There is a need for investigating the introduction of more essential variables than the three ones that we used in this study. For example, a resistivity well log, which is the recording of the resistance of formation water to natural or induced electrical current, might be a very useful log to be used in facies prediction.
- All of our training was done with less than 2800 examples. This may be considered a modest amount of data for such a complicated problem. A collection of more data at the training phase is needed for building a more precise DT model that is capable to have better accuracy.
- The “majority voting” approach is a well-known methodology that can be used to achieve better results when the domain of the problem is a complicated one. In this study we did not utilize this powerful concept and such utilization may be worth investigating.

- Several authors have recently noted that C4.5's performance is weaker in domains with a preponderance of continuous attributes than for learning tasks that have mainly discrete attributes. However, there is a new release with improved use of continuous attributes in C4.5. Since most of the attributes used in this problem are of a continuous type, training and testing in such a release might positively help the final outcome and be worth questioning.
- We captured the geologist's knowledge and the known facts about the reservoir and we presented them as input variables to the DTLA. For example, variables *OAP_or_DOP* had been added to stand for the possibility that the specific point-of-depth for a case is of type OAP or DOP facies. A value of "Y" or "N" is set for *OAP_or_DOP* based on the following criteria:

$$\begin{array}{l}
 \textit{OAP_or_DOP} = \left\{ \begin{array}{ll}
 \text{Y} & \text{For all the examples occurring at 85 \%} \\
 & \text{or more from the top of the formation} \\
 \text{N} & \text{Otherwise}
 \end{array} \right.
 \end{array}$$

However, *OAP_or_DOP* might occur in some wells at 80 % or more from the top of the formation and in other wells at 90 % or more.

Therefore, *OAP_or_DOP* might be better evaluated based on something

like this:

$$OAP_or_DOP = \left\{ \begin{array}{l} 0.0 \text{ For all the examples occurring between} \\ \quad 0 \text{ to } 80 \% \text{ from the top of the formation.} \\ 0.5 \text{ For all the examples occurring between} \\ \quad 81 \text{ to } 85 \% \text{ from the top of the} \\ \quad \text{formation.} \\ 0.75 \text{ For all the examples occurring between} \\ \quad 86 \text{ to } 89 \% \text{ from the top of the} \\ \quad \text{formation.} \\ 1.0 \text{ Otherwise} \end{array} \right.$$

- Combining both neural networks with decision trees for general decision-making operations may be advisable.

Keywords

Core	A column sample taken from the reservoir formation.
Facies	The general appearance or aspect of rock.
Hydrocarbon	Oil and gas.
Formation	A bed or deposit composed throughout of substantially the same kind of rock.
Limestone	Sedimentary rock rich in calcium carbonate that sometimes serves as a reservoir rock for petroleum.
Lithologic	Characteristics of a rock in terms of mineral composition, structure, and so forth.
Permeability	Measurement of the ease with which fluids can flow through a porous rock.
Porosity	Measurement of the opening or space within a rock usually filled with fluid.
Well log	A measurement of the reflection of the electric or radioactive waves that have been generated by log devices.

REFERENCES

- [ABKM94] Mehmet Altunbay, D.C. Barr, A.F. Kennaird, D.K. Manning. "Numerical Geology: Predicting Depositional and Diagenetic Facies from Wireline Logs Using Core Data". SPE Asia Oil & Gas Conference, Melbourne, Australia, November 7-10, 1994, pages 507-515.
- [ADD90] H. Anxionnaz, P. Delfiner, J. Delhomme. "Computer-Generated Core-like Descriptions from Open-Hole Logs". The American Association of Petroleum Geologist Bulletin, V. 74, No. 4, April 1990, Pages 375-393.
- [Ali94] J.K. Ali. "Neural Networks: A New Tool for the Petroleum Industry". European Petroleum Computer Conference, Aberdeen, U.K., March 15-17, 1994, pages 217-231.
- [AMA90] F.G. Alabert, G.J. Massonnat, Elf Aquitaine. "Heterogeneity in a Complex Turbiditic Reservoir: Stochastic Modeling of Facies and Petrophysical Variability". SPE 65th Annual Technical Conference and Exhibition, New Orleans, LA, September 23-26, 1990, pages 775-790.
- [Bald91] J.L. Baldwin. "Using a Simulated Bidirectional Associative Neural Network Memory with Incomplete Prototype Memories to Identify Facies from Intermittent Logging Data Acquired in a Silliciclastic Depositional Sequence: A Case Study". SPE 1991 Annual Technical Conference and Exhibition, Dallas, TX, October 6-9, 1991, pages 273-286.
- [BMA95] S. Mohaghegh, B. Balan, S. Ameri. "State-of-the-art in Permeability Determination from Well Log Data: Part 1- A Comparative Study, Model Development". SPE Eastern Regional Conference and Exhibition, West Virginia, USA, September 18-20, 1995, pages 232-239.
- [Boom95] Robert J. Boomer. "Predicting Production Using a Neural Network". Petroleum Computer Conference, Houston, TX, June 11-14, 1995, pages 195-204.
- [BOW89] J.L. Baldwin, D.N. Otte, C.L. Wheatley. "Computer Emulation of Human Mental Processes: Application of Neural Network Simulator to Problems in Well Log Interpretation". SPE 64th Annual Technical Conference and Exhibition, San Antonio, TX, October 8-11, 1989, pages 481-486.
- [Chak93] S. Chakravarty. "A Characterization of Binary Decision Diagrams". IEEE Trans. On Computers, V. 42, March 1993, pages 129-137.
- [Cock90] J. R. Cockett. "Decision Tree Reduction". Journal ACM, V. 37, no. 4, October 1990, pages 815-842.
- [CT85] M. Cruz, H. Takizawa. "Automatic Facies Analysis in the Arab Formation, El Bundug Field, Offshore Abu Dhabi/Qatar". SPE Annual Technical Conference and Exhibition, Vagas, USA, September 22-25, 1985, pages 1-11.

- [DBB95] S. Dear, R. Beasley, K. Barr. "Use of a decision tree to select mud system for the Oso field, Nigeria". JPT, V. 47, October 1995, pages 909-912.
- [GE97] RB. Gharbi, A.M. Elsharkawy. "Neural Network Model for Estimating the PVT Properties of Middle East Crude Oils". 1997 Middle East Oil Show, Bahrain, March 15-18, 1997, pages 151-166.
- [GG92] H. Guo, S. Gelfand. "Classification Trees with Neural Network Feature Extraction". IEEE Trans. On Neural Networks, V. 3, November 1992, pages 923-933.
- [HNKE94] J. Hook, J. Nieto, C. Kalkomey, D. Ellis. "Facies and Permeability Prediction from Wireline Logs and Core – A North Sea Case Study". SPWLA 35th Annual Logging Symposium, June 19-22, 1994, pages 1-7.
- [Kung94] D. C. Kung. "On Decision Tree Verification and Consolidation". Department of Computer Science Engineering, The University of Texas, Arlington, TX, 76019-0015, 1994, pages 485-494.
- [MA95] S. Mohaghegh, S. Ameri. "Artificial Neural Network as a Valuable Tool for Petroleum Engineers". SPE 29220, 1995, pages 1-8.
- [MAA94] S. Mohaghegh, R. Arefi, S. Ameri. "A Methodological Approach for Reservoir Heterogeneity Characterization Using Artificial Neural Networks". SPE Annual Technical Conference and Exhibition, New Orleans, LA, September 25-28, 1994, pages 1-5.
- [MAAN96] S. Mohaghegh, R. Arefi, S. Ameri, R. Nutter. "Petroleum Reservoir Characterization with the Aid of Artificial Neural Networks". Journal of Petroleum Science and Engineering, v. 16, 1996, pages 263-274.
- [MBA95] S. Mohaghegh, B. Balan, S. Ameri. "State-of-the-art in Permeability Determination from Well Log Data: Part 2- Verifiable, Accurate Permeability Predictions, the TouchStone of All Models". SPE Eastern Regional Conference and Exhibition, West Virginia, USA, September 18-20, 1995, pages 104-109.
- [MBAD96] S. Mohaghegh, B. Balan, S. Ameri, D. McVey. "A Hybrid Neuro-Genetic Approach to Hydraulic Fracture Treatment Design and Optimization". SPE Annual Technical Conference and Exhibition, Denver, USA, October 6-9, 1996, pages 201-211.
- [MHA96] S. Mohaghegh, M. Hugh Hefner, S. Ameri. "Fracture Optimization eXpert (FOX) – How Computational Intelligence Helps the Bottom-Line in Gas Storage; A Case Study". SPE Eastern Regional Conference and Exhibition, Ohio, USA, October 23-25, 1996, pages 1-5.
- [MMAA95] S. Mohaghegh, D. McVey, S. Ameri, K. Aminian. "Predicting Well Stimulation Results in a Gas Storage Field in the Absence of Reservoir Data, Using Neural Networks". SPE 31159, September 1995, pages 1-8.

- [MMAA96] D. McVey, S. Mohaghegh, K. Aminian, S. Ameri. "Identification of Parameters Influencing the Responses of Gas Storage Wells to Hydraulic Fracturing with the Aid of a Neural Network". SPE Computer Applications, April 1996, pages 54-57.
- [MS97] J. MacDonald, R. Smith. "Decision Trees Clarify Novel Technology Applications". Oil and Gas Journal, V. 95, February 1997, pages 69-71.
- [Murp96] JP. Murphy, Betty Olson. "Decision Tree Construction and Analysis". Journal AWWA, February 1996, pages 59-67.
- [PL88] R.C.A. Peveraro, J.A. Lee. "HESPER: An Expert System for Petrophysical Formation Evaluation". SPE European Petroleum Conference, London, UK, October 16-19, 1988, pages 100-109.
- [Quin90] J. R. Quinlan. "Decision Trees and Decision-Making". IEEE Trans. Systems, Man and Cybernetics, V. 20, No. 2, 1990, pages 339-346.
- [Quin92] J. R. Quinlan. "Learning with Continuous classes". Proceedings of Artificial Intelligence 92 Australian National Conference on Artificial, Singapore, 1992, pages 343-348.
- [Quin93] J. R. Quinlan. "A Case Study in Machine Learning". Proceedings ACSC 16th Australian Computer Science Conference, Brisbane, January 1993, pages 731-737.
- [Quin96] J. R. Quinlan. "Learning Decision Tree Classifiers". ACM Computing Surveys, V. 28, no. 1, March 1996, pages 71-72.
- [Quin97] J. R. Quinlan. "Decision Trees and Instance-based Classifiers". CRC Press, 1997, pages 1-11.
- [Qun96] J. R. Quinlan. "Improved Use of Continuous Attributes in C4.5". Journal of Artificial Intelligence Research 4, 1996, pages 77-90.
- [Qunl96] J. R. Quinlan. "Learning First-order Definitions of Functions". Journal of Artificial Intelligence Research, V. 5, 1996, pages 139-161.
- [SK86] R.A. Startzman, T.B. Kuo. "An Artificial Intelligence Approach to Well Log Correlation". SPWLA 27th Annual Logging Symposium, June 9-13, 1986, pages 1-10.
- [SM88] S. Sakurai, J. Melvin. "Facies Discrimination and Permeability Estimation from Well Logs for the Endicott Field". SPWLA 29th Annual Logging Symposium, June 5-8, 1988, pages 1-12.
- [SM93] A. Sankar, R. Mammone. "Growing and Pruning Neural Tree Networks". IEEE Trans. On Computers, V. 42, March 1993, pages 291-299.

- [TBM95] J. Ternyik, H. Bilgesu, S. Mohaghegh. "Virtual Measurement in Pipes, Part 2: Liquid Holdup and Flow Pattern Correlation". SPE Eastern Regional Conference and Exhibition, West Virginia, USA, September 18-20, 1995, pages 1-5.
- [UBC97] P. Utgoff, N. Berkman, J. Clouse. "Decision Tree Induction Based on Efficient Tree Restructuring". Machine Learning, V. 29, October 1997, pages 5-44.
- [Well88] J. M. Weller. "Stratigraphic Facies Differentiation and Nomenclature". Bulletin of the American of Petroleum Geologist, March 1988, pages 609-639.
- [WMMA95] A. White, D. Moinar, S. Mohaghegh, S. Ameri. "The Application of ANN for Zone Identification in a Complex Reservoir". SPE Eastern Regional Conference and Exhibition, West Virginia, USA, September 19-23, 1995, pages 1-5.

Appendix A

C4.5 [release 5] decision tree generator Sun Sep 20 12:03:56 1998

Options:

File stem <finalout>
Trees evaluated on unseen cases

Read 2663 cases (23 attributes) from finalout.data

Decision Tree:

```
oap_or_dop = y:
| gre_bd <= 0.543599 :
| | gre_ad <= -8.44684 :
| | | greM <= 12.9077 : DOP (4.0/1.0)
| | | greM > 12.9077 : OAP (3.0/1.0)
| | gre_ad > -8.44684 :
| | | gre_ad <= -6.45468 :
| | | | dep <= 5410 : OAP (5.0)
| | | | dep > 5410 : DOP (4.0)
| | | gre_ad > -6.45468 :
| | | | porM <= 0.2969 : OAP (199.0/1.0)
| | | | porM > 0.2969 :
| | | | | dep > 5438 : DOP (4.0)
| | | | | dep <= 5438 :
| | | | | | den > 2.224 : OAP (30.0)
| | | | | | den <= 2.224 :
| | | | | | | dep <= 5398 : DOP (2.0)
| | | | | | | dep > 5398 : OAP (2.0)
| gre_bd > 0.543599 :
| | den_bd > 0.05696 : SS (3.0)
| | den_bd <= 0.05696 :
| | | gre > 17.226 : SS (3.0/1.0)
| | | gre <= 17.226 :
| | | | greM > 11.7614 : OAP (17.0)
| | | | greM <= 11.7614 :
| | | | | denM > 2.32411 : OAP (11.0)
| | | | | denM <= 2.32411 :
| | | | | | porM <= 0.2394 : OAP (4.0)
| | | | | | porM > 0.2394 :
| | | | | | | gre_ad <= 2.1926 :
| | | | | | | | den_bd > 0.00268 : OAP (5.0)
| | | | | | | | den_bd <= 0.00268 :
| | | | | | | | | gre_ad <= 1.82784 : DOP (10.0/1.0)
| | | | | | | | | gre_ad > 1.82784 : OAP (2.0)
| | | | | | | gre_ad > 2.1926 :
| | | | | | | | gre_ad > 4.34745 : DOP (103.0/2.0)
| | | | | | | | gre_ad <= 4.34745 :
| | | | | | | | | dep <= 5364 : DOP (13.0)
| | | | | | | | | dep > 5364 : OAP (5.0/1.0)
```



```

oap_or_dop = n:
| region = 0: LICO (0.0)
| region = 1: SS (560.0)
| region = 2: LICO (0.0)
| region = 3:
|   pos_lico = y:
|     denM > 2.37631 : BB (35.0)
|     denM <= 2.37631 :
|       dep <= 5152 :
|         greM <= 8.2851 : LICO (10.0)
|         greM > 8.2851 : BB (12.0)
|       dep > 5152 :
|         greM <= 14.548 :
|           greM <= 12.1575 :
|             greM <= 10.7086 : LICO (345.0/1.0)
|             greM > 10.7086 :
|               greM <= 10.844 : OAP (7.0)
|               greM > 10.844 : LICO (107.0)
|             greM > 12.1575 :
|               greM <= 12.3513 : OAP (7.0)
|               greM > 12.3513 : LICO (32.0)
|           greM > 14.548 :
|             greM <= 16.5215 : OAP (10.0)
|             greM > 16.5215 : LICO (12.0)
|   pos_lico = n:
|     greM <= 12.1575 :
|       porM <= 0.259818 :
|         greM <= 11.437 :
|           greM <= 8.04775 :
|             porM <= 0.227111 :
|               denM > 2.32512 : BB (182.0)
|               denM <= 2.32512 :
|                 dep <= 5087 : BB (70.0)
|                 dep > 5087 : LG (11.0)
|             porM > 0.227111 :
|               denM <= 2.29543 : BB (68.0/1.0)
|               denM > 2.29543 : LICO (12.0)
|           greM > 8.04775 :
|             porM <= 0.194438 : BB (56.0)
|             porM > 0.194438 :
|               greM > 10.8155 : BB (24.0/1.0)
|               greM <= 10.8155 :
|                 den_bd > 0.05992 : LG (43.0)
|                 den_bd <= 0.05992 :
|                   dep <= 4968 : BB (12.0)
|                   dep > 4968 :
|                     dep > 5201 : BB (7.0)
|                     dep <= 5201 :
|                       dep > 5109 : LG (60.0)
|                       dep <= 5109 :
|                         denM > 2.32973 : LG (16.0)
|                         denM <= 2.32973 :
|                           denM > 2.29543 : BB (50.0)
|                           denM <= 2.29543 : [S1]

```

```

| | | | | greM > 11.437 :
| | | | | | dep <= 5113 : BB (12.0)
| | | | | | dep > 5113 : LICO (24.0)
| | | | | porM > 0.259818 :
| | | | | | dep <= 5156 : LG (4.0)
| | | | | | dep > 5156 : LICO (25.0)
| | | | greM > 12.1575 :
| | | | | dep <= 5190 : LG (376.0)
| | | | | dep > 5190 :
| | | | | | dep <= 5236 : LICO (8.0)
| | | | | | dep > 5236 : OAP (6.0)

```

Subtree [S1]

```

porM <= 0.214 : BB (4.0)
porM > 0.214 : LG (27.0)

```

Simplified Decision Tree:

```

oap_or_dop = y:
| gre_bd <= 0.543599 :
| | porM <= 0.2969 : OAP (208.0/5.0)
| | porM > 0.2969 :
| | | dep > 5438 : DOP (9.0/1.3)
| | | dep <= 5438 :
| | | | den > 2.224 : OAP (32.0/2.6)
| | | | den <= 2.224 :
| | | | | dep <= 5398 : DOP (2.0/1.0)
| | | | | dep > 5398 : OAP (2.0/1.0)
| gre_bd > 0.543599 :
| | den_bd > 0.05696 : SS (3.0/1.1)
| | den_bd <= 0.05696 :
| | | gre > 17.226 : SS (3.0/2.1)
| | | gre <= 17.226 :
| | | | greM > 11.7614 : OAP (17.0/1.3)
| | | | greM <= 11.7614 :
| | | | | denM > 2.32411 : OAP (11.0/1.3)
| | | | | denM <= 2.32411 :
| | | | | | porM <= 0.2394 : OAP (4.0/1.2)
| | | | | | porM > 0.2394 :
| | | | | | | gre_ad <= 2.1926 :
| | | | | | | | den_bd > 0.00268 : OAP (5.0/1.2)
| | | | | | | | den_bd <= 0.00268 :
| | | | | | | | | gre_ad <= 1.82784 : DOP (10.0/2.4)
| | | | | | | | | gre_ad > 1.82784 : OAP (2.0/1.0)
| | | | | | | gre_ad > 2.1926 :
| | | | | | | | gre_ad > 4.34745 : DOP (103.0/3.8)
| | | | | | | | gre_ad <= 4.34745 :
| | | | | | | | | dep <= 5364 : DOP (13.0/1.3)
| | | | | | | | | dep > 5364 : OAP (5.0/2.3)
oap_or_dop = n:
| region = 0: LICO (0.0)
| region = 1: SS (560.0/1.4)
| region = 2: LICO (0.0)
| region = 3:

```

```

pos_lico = y:
| denM > 2.37631 : BB (35.0/1.4)
| denM <= 2.37631 :
| | dep <= 5152 :
| | | greM <= 8.2851 : LICO (10.0/1.3)
| | | greM > 8.2851 : BB (12.0/1.3)
| | dep > 5152 :
| | | greM <= 14.548 :
| | | | greM <= 12.1575 :
| | | | | greM <= 10.7086 : LICO (345.0/2.6)
| | | | | greM > 10.7086 :
| | | | | | greM <= 10.844 : OAP (7.0/1.3)
| | | | | | greM > 10.844 : LICO (107.0/1.4)
| | | | greM > 12.1575 :
| | | | | greM <= 12.3513 : OAP (7.0/1.3)
| | | | | greM > 12.3513 : LICO (32.0/1.4)
| | greM > 14.548 :
| | | greM <= 16.5215 : OAP (10.0/1.3)
| | | greM > 16.5215 : LICO (12.0/1.3)
pos_lico = n:
| greM <= 12.1575 :
| | porM <= 0.259818 :
| | | greM <= 11.437 :
| | | | greM <= 8.04775 :
| | | | | porM <= 0.227111 :
| | | | | | denM > 2.32512 : BB (182.0/1.4)
| | | | | | denM <= 2.32512 :
| | | | | | | dep <= 5087 : BB (70.0/1.4)
| | | | | | | dep > 5087 : LG (11.0/1.3)
| | | | | porM > 0.227111 :
| | | | | | denM <= 2.29543 : BB (68.0/2.6)
| | | | | | denM > 2.29543 : LICO (12.0/1.3)
| | | greM > 8.04775 :
| | | | porM <= 0.194438 : BB (56.0/1.4)
| | | | porM > 0.194438 :
| | | | | greM > 10.8155 : BB (24.0/2.5)
| | | | | greM <= 10.8155 :
| | | | | | dep > 5201 : BB (7.0/1.3)
| | | | | | dep <= 5201 :
| | | | | | | dep > 5109 : LG (60.0/1.4)
| | | | | | | dep <= 5109 :
| | | | | | | | denM > 2.32973 : LG (35.0/1.4)
| | | | | | | | denM <= 2.32973 :
| | | | | | | | | denM > 2.29543:BB (53.0/1.4)
| | | | | | | | | denM <= 2.29543 :
| | | | | | | | | | porM <= 0.214:BB (13.0/1.3)
| | | | | | | | | | porM > 0.214: LG (51.0/1.4)
| | | greM > 11.437 :
| | | | dep <= 5113 : BB (12.0/1.3)
| | | | dep > 5113 : LICO (24.0/1.3)
| | porM > 0.259818 :
| | | dep <= 5156 : LG (4.0/1.2)
| | | dep > 5156 : LICO (25.0/1.3)
greM > 12.1575 :
| | dep <= 5190 : LG (376.0/1.4)
| | dep > 5190 :
| | | dep <= 5236 : LICO (8.0/1.3)

```

| | | | | dep > 5236 : OAP (6.0/1.2)

Tree saved

Evaluation on training data (2663 items):

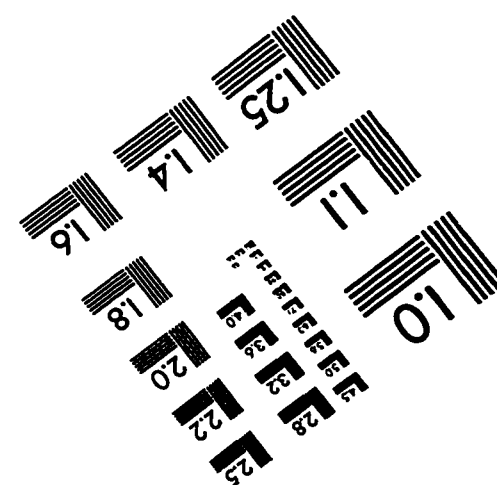
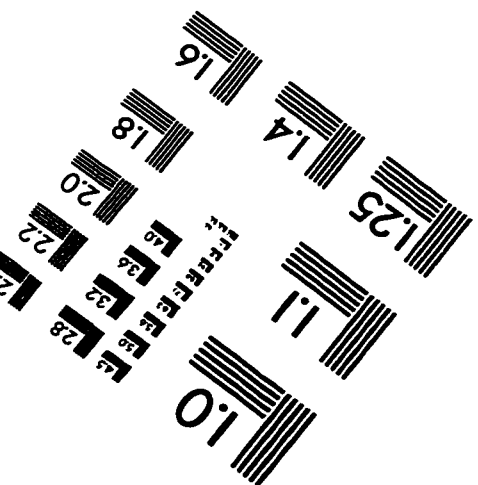
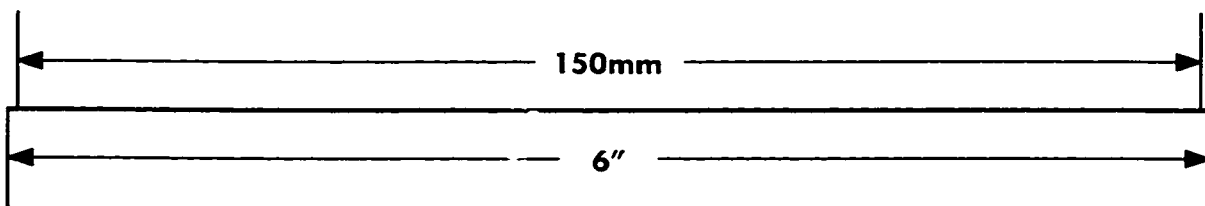
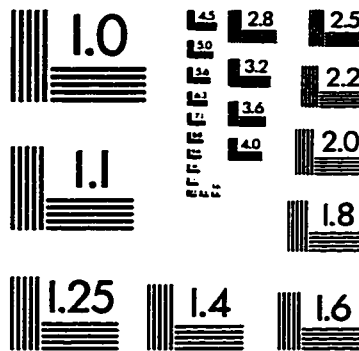
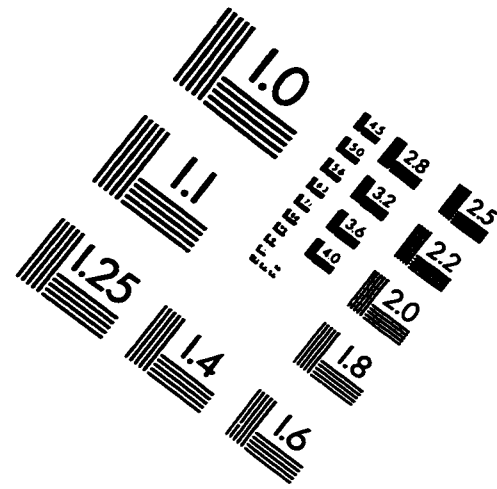
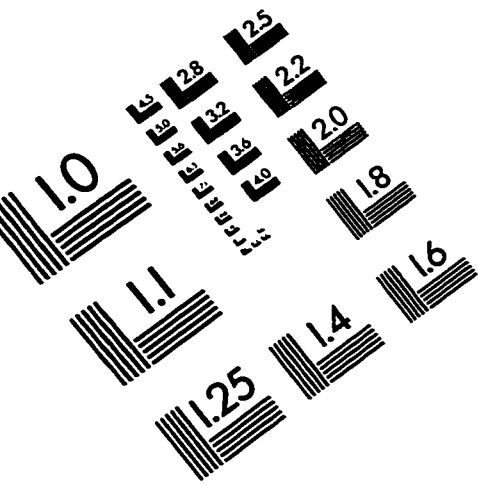
Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
107	11 (0.4%)	95	12 (0.5%)	(2.8%) <<

Evaluation on test data (681 items):

Before Pruning		After Pruning		
Size	Errors	Size	Errors	Estimate
107	63 (9.3%)	95	63 (9.3%)	(2.8%) <<

(a)	(b)	(c)	(d)	(e)	(f)	<-classified as
188						(a): class SS
	95	18				(b): class BB
	5	118				(c): class LG
	1	24	114			(d): class LICO
			3	67	4	(e): class OAP
1				7	36	(f): class DOP

IMAGE EVALUATION TEST TARGET (QA-3)



APPLIED IMAGE, Inc
1653 East Main Street
Rochester, NY 14609 USA
Phone: 716/482-0300
Fax: 716/288-5989

© 1993, Applied Image, Inc., All Rights Reserved