UNIVERSITY OF BAHRAIN

College of Information Technology



Modeling the Accuracy-cost Relationship in Decision Trees

A Thesis Submitted in Partial Fulfilment of the Requirements for the Master Degree in Information Technology

Submitted by
Mona Mansoor Merza Al Hamad
University number: 20021438

Supervised by
Dr. Ahmed M. Zeki
(Assistant Professor)
University of Bahrain

Kingdom of Bahrain
April, 2019



UNIVERSITY OF BAHRAIN

College of Information Technology

"Modeling the Accuracy-cost Relationship in Decision Trees"

A Thesis Submitted in Partial Fulfilment of the Requirements for the M.Sc. Degree in Information Technology

Submitted by

Mona Mansoor Mirza Ali 20021438

Supervised by

Dr. Ahmed M. Zeki

Kingdom of Bahrain

April 2019

University of Bahrain
Deanship of scientific Research



جامعة البحرين
عمادة البحث العلمي

Dissertation Deposit Form

Author: Muna Mansoor Merza Ali

Thesis Title: Modeling the Accuracy-cost Relationship in Decision Trees

Year: Second Semester 2018-2019

College: Information Technology

Email:mona.alhamad@gmail.com Contact No:34466256

Author's Declaration: I agree to the following conditions:

The above thesis will be available (in the University of Bahrain library and externally) and reproduced as necessary at the discretion of the University of Bahrain. It may also be digitized by the University of Bahrain and made available via the university's repository or via other parties on the Internet.

This condition shall apply to ALL copies including electronic copies.

The above thesis has been provided on the understanding that it is copyright protected material and that no quotation from this thesis may be published without proper acknowledgement.

Signature of the Author:

Date: 26 June 2019

DECLARATION

I declare that this work is the result of my own investigation and that it has not already been accepted in substance for any degree, nor is it currently submitted for any degree.

Signed: ---------- Date: 15/4/2019

Mona Mansoor Mirza Ali Al Hamad (Candidate)

The Thesis defense committee considers the thesis titled:

Modeling the Accuracy-Cost Relationship in Decision Trees

Submitted by Mona Mansoor Mirza Ali Al Hamad to the College of Information Technology is satisfactory and acceptable for the Master of Science in Information Technology.

The Defense Committee:

Dr. Ahmed M. Zeki
(Supervisor)
Department of Information Systems
University of Bahrain

Dr. Andri Mirzal
(Member)
Arab Gulf University
Kingdom of Bahrain

Dr. Mustaffa Hammad
(Member)
Department of Computer Science
University of Bahrain

Abstract

Decision Trees have been applied widely for classification in many fields such as finance, marketing, engineering, and medical diagnosis. Due to the wide range of applications, it is crucial to fully understand the various aspects of decision trees including the different type of costs associated with the classification task. It is also crucial to understand the relationship between the classifier's accuracy and cost, as balancing the two is a major concern nowadays in many fields such as the medical diagnosis field.

This research explores the way of solving the accuracy/cost trade-off in a decision tree classifier to get a low cost and an accurate classifier. It aims at modeling the relationship between the classification accuracy and the cost in a decision tree and determining the optimal balance between the two that gives a satisfactory accuracy at the lowest possible cost.

During the research, different pruning methods have been utilized to control the level of tree pruning and improve the classification accuracy in a decision tree classifier. Different pruning parameters have to be used carefully as they may lead to noticeable difference in the classifier's performance. Therefore, this research tested various pruning parameters options in the standard decision tree algorithm, and then the Pareto dominance approach was used to select the best setting as it is known for its powerfulness and effectiveness in addressing different data mining problems.

The experimental results indicate the efficiency of using different post-pruning and pre-pruning methods in reducing the average total classification cost significantly with a very slight reduction in the accuracy in some cases. The resulting model between the classification accuracy and the average total cost was proportional inverse. When the proposed solutions compared to the standard decision tree algorithm with its default pruning settings, the reduction in average total classification cost reached 21.14%, 19.79%, and 5.27% in breast cancer, heart disease, and thyroid disease datasets, respectively. The resulting accuracy rate was reduced only by 1.6% in breast cancer dataset. In heart disease dataset, the resulting classification accuracy was increased by 6.68% and the same accuracy rate was obtained in thyroid disease dataset.

Publications

AlHamad, M. & Zeki, A. (2018). Accuracy vs. Cost in Decision Trees: A Survey. International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 18-19 November 2018, Bahrain.

AlHamad, M. & Zeki, A. (n.d.). Modeling the Accuracy-cost Relationship in Decision Trees (Journal). In progress.

Dedication

This thesis work is dedicated to my beloved father who was waiting for this work to be accomplished and made me work hard but unfortunately he passed away before seen the fruit of this research, may Allah (s.w.) shower him with his mercy.

I also dedicate this thesis to my great mother, brothers, and sisters. I dedicate and give a special thanks to my husband "Talib" and wonderful daughters "Amal" and "Maryam".

Acknowledgements

First of all, I would like to express my sincere gratitude to my thesis supervisor Dr. Ahmed M. Zeki, the assistant professor at the University of Bahrain, for his continuous support, for his patience, and for his valuable comments given on this thesis. He directed me to the right direction whenever I needed guidance.

I also would like to thank Professor Dr. Nabil Al Hewahi, the coordinator of the Masters program at the college of Information Technology, for his guidance and for facilitating all the required procedures.

My sincere thanks also goes to Dr. Jihene Harrath, Dr. Jaflah Al Ammari, and Dr. Hayat Ali the assistant professors at the University of Bahrain. The door was always opened whenever I need an advice or help related to my research.

Last but not least, I would like to acknowledge my husband and my family for their sincere support and continuous encouragement during the whole period of my master thesis. The successful achievement I reached would not become real without them. Thank you.

Mona Al Hamad

Table of Contents

Abstract..... A

Publications..... B

Dedication..... C

Acknowledgements..... D

Table of Contents..... E

List of Tables..... K

List of Figures..... M

List of Abbreviations..... R

List of Appendices..... S

Chapter One

Introduction

1.0 Overview..... 1

1.1 Background of the Study..... 2

1.2 Problem Statement..... 3

1.3 Objectives of the Study..... 4

1.4 Research Questions..... 4

1.5 The Model of the Study..... 5

1.6 Significance of the Study..... 7

1.7 Scope and Limitations of the Study..... 7

1.8 Structure of the Thesis..... 8

1.9 Definition of Terms..... 8

1.10 Conclusion..... 9

Chapter Two

Classification Cost vs. Accuracy: A Theoretical Background

2.0 Overview..... 11

2.1 What is Data Mining? 11

2.2 Knowledge Discovery in Databases..... 12

2.3 Data Mining Applications..... 15

2.4 Taxonomy of Data Mining Methods.....16

2.4.1 Classification.....17

2.4.1.1 Performance Measures for a Classifier.....18

2.4.1.2 Classification by Decision Tree Induction..... 19

2.4.1.3 Decision Tree Construction Algorithms..... 21

2.4.2 Regression..... 22

2.4.3 Clustering 22

2.4.4 Association Rules Mining..... 23

2.5 Data Mining Tools..... 23

2.6 Costs Involved in the DM Process 23

2.6.1 Types of Costs Involved in Inductive Learning Process 24

2.6.1.1 Misclassification Cost (MC) 24

2.6.1.2 Test Cost (TC) 25

2.6.1.3 Intervention Cost 26

2.6.1.4 Unwanted Achievement Cost 26

2.6.1.5 Computational Cost 27

2.6.1.6 Human-computer Interaction Cost..... 28

2.6.2 What Types of Cost to Consider? 29
2.6.3 Calculating the Total Classification Cost in DTs 30
2.6.4 Data Mining Cost Units 32
2.7 Relationship between Classification Cost Types and Accuracy 32
2.8 Cost-sensitive Models..... 35
2.8.1 Misclassification Cost Reduction 36
2.8.2 Test Cost Reduction 36
2.8.3 Misclassification and Test Cost Reduction..... 38
2.8.4 Computational Cost Reduction 40
2.9 Conclusion..... 43

Chapter Three

Research Methodology

3.0 Overview..... 45
3.1 Research Methodology..... 45
3.2 Datasets Used in the Experiments..... 49
3.2.1 Heart Disease Dataset 49
3.2.2 Thyroid Disease Dataset 51
3.2.3 Wisconsin Breast Cancer Dataset 52
3.2.4 Datasets Properties..... 53
3.3 DT Classifiers: J48..... 54
3.4 Classification Cost Calculation..... 55
3.4.1 Misclassification and Test Cost Settings 56
3.4.2 Average Total Classification Cost Calculation..... 59

3.5 Accuracy Calculation.....60

3.6 Accuracy vs Cost Optimization..... 60

3.6.1 The Conventional Weighted Formula Approach..... 61

3.6.2 Pareto Approach..... 61

3.6.3 Lexicographic Approach..... 63

3.7 Decision Tree Pruning.....64

3.7.1 Post-Pruning Methods..... 64

3.7.1.1 Error-based Pruning..... 64

3.7.1.2 Reduced Error Pruning.....65

3.7.1.3 Minimum Error Pruning.....66

3.7.1.4 Cost Based Pruning..... 67

3.7.2 Pre-pruning Methods 68

3.7.2.1 Minimum No. of Object Pruning..... 68

3.7.2.2 Chi-square Pruning 69

3.7.3 Pruning Parameters in Weka..... 69

3.8 Conclusion..... 70

Chapter Four

Results and Analysis

4.0 Overview.....72

4.1 Experimental Framework.....72

4.2 Experimental Setup..... 76

4.3 Results Discussion..... 79

4.3.1 Heart Disease Dataset 79

4.3.1.1 Setting the Confidence Factor Parameter..... 79

4.3.1.2 Setting the Minimum Number of Objects Parameter..... 82

4.3.1.3 Setting the Number of Reduced Error Pruning's Folds..... 85

4.3.2 Thyroid Disease Dataset 88

4.3.2.1 Setting the Confidence Factor Parameter..... 88

4.3.2.2 Setting the Minimum Number of Objects Parameter..... 91

4.3.2.3 Setting the Number of Reduced Error Pruning's Folds..... 93

4.3.3 Breast Cancer Dataset 96

4.3.3.1 Setting the Confidence Factor Parameter.....96

4.3.3.2 Setting the Minimum Number of Objects Parameter..... 99

4.3.3.3 Setting the Number of Reduced Error Pruning's Folds..... 101

4.3.4 Datasets Comparision 104

4.4 Results Validation..... 107

4.5 The Impact of Dataset Characteristics on Accuracy/cost of a Classifier 111

4.6 Use of Pareto Dominance Approach to Resolve the Accuracy/cost Trade-off 114

4.7 Results Comparison..... 124

4.8 Conclusion 125

Chapter Five

Conclusion and Recommendation

5.0 Overview 129

5.1 Summary of the Approach Followed..... 129

5.2 Summary of Results.....131

5.3 Comparison with Other Researchers Results.....137

5.4 The Thesis Contribution 138

5.5 Future Directions.....139

References 140

Appendices.....148

Abstract (Arabic).....149

List of Tables

Table 2.1: Confusion Matrix for Binary Classification 18

Table 2.2: Classifier's Evaluation Metrics 19

Table 2.3: An Example of a Classification Cost Matrix 25

Table 2.4: Some Medical Tests and their Costs 26

Table 2.5: Type of Costs involved in inductive Learning Process 29

Table 2.6: The TC Vector 31

Table 2.7: Algorithms for Maximizing accuracy and Minimizing cost 42

Table 3.1: Attributes in HD Dataset 50

Table 3.2: Attributes in Thyroid Disease Dataset 51

Table 3.3: Attributes in Breast Cancer Dataset 52

Table 3.4: Dataset Properties 54

Table 3.5: HD Dataset Attribute's TC 56

Table 3.6: Thyroid Disease Dataset Attribute's Cost 57

Table 3.7: Breast Cancer Dataset's Attributes Cost 58

Table 3.8: MC Matrix of the HD and Breast Cancer Datasets 58

Table 3.9: MC Matrix of the Thyroid Disease Dataset 59

Table 4.1: Experimental Stages 78

Table 4.2: Experimental Results on the Thyroid Dataset 89

Table 4.3: The Resulting Relations using All Datasets 105

Table 4.4: Correlation Matrix for all Datasets 108

Table 4.5: Summery of all Correlation Matrices 110

Table 4.6: The Correlation Matrix of Dataset Properties and the Classifier's Accuracy/Cost 112

Table 4.7: The Model Summery in Regression Analysis Using Accuracy 113

Table 4.8: The Model Summery in Regression Analysis Using ATC 114

Table 4.9: Non-dominated Solutions 117

Table 4.10: Selected Solutions from Pareto Front 118

Table 4.11: Percentage of ATC Reduction Compared to Highest ATC 118

Table 4.12: Percentage of ATC and Accuracy Reduction Compared to the Standard C4.5 120

Table 4.13: Research Results Comparison 125

Table 5.1: Answers to the Research Questions 136

List of Figures

Figure 1.1: Research Methodology 7

Figure 2.1: The KDD Process 14

Figure 2.2: Taxonomy of Data Mining Techniques 16

Figure 2.3: DT Example 20

Figure 2.4: A DT Classifier (Xu, Min, & Zhu, 2013) 31

Figure 2.5: Accuracy vs. Tree Size 34

Figure 2.6: Error-cost Relationship in a DM Model 35

Figure 3.1: Research Methodology 48

Figure 3.2: An Example of Pareto Dominance 63

Figure 3.3: Reduced Error Pruning Method 66

Figure 3.4: Cost Based Pruning Method 68

Figure 4.1: Experimental Framework 76

Figure 4.2: The DT Constructed using the CF Pruning in HD Dataset 79

Figure 4.3: The Tree size vs. Accuracy in the HD Dataset 80

Figure 4.4: Average MC vs. Accuracy in HD Dataset 81

Figure 4.5: TC vs. Accuracy in HD Dataset 81

Figure 4.6: ATC Cost vs. Accuracy in HD Dataset 82

Figure 4.7: The DT Constructed using the MinNumObj Pruning in the HD Dataset 83

Figure 4.8: Tree Size vs. Accuracy in the HD Dataset 83

Figure 4.9: Average MC vs. Accuracy in the HD Dataset 84

Figure 4.10: Average TC vs Accuracy in the HD Dataset 84

Figure 4.11: The ATC vs Accuracy in the HD Dataset 85

Figure 4.12: The DT Constructed using the NumFold Pruning in the HD Dataset 86

Figure 4.13: Tree Size vs. Accuracy in the HD Dataset 86

Figure 4.14: The MC Cost vs. Accuracy in the HD Dataset 87

Figure 4.15: The Average TC vs Accuracy in the HD Dataset 87

Figure 4.16: The ATC vs Accuracy in the HD Dataset 88

Figure 4.17: The DT Constructed using the CF Pruning in the Thyroid Disease Dataset 89

Figure 4.18: The Average MC vs Accuracy in the Thyroid Disease Datase..90

Figure 4.19: Average TC vs. Accuracy in the Thyroid Dataset 90

Figure 4.20: ATC vs. Accuracy in Thyroid Disease Dataset 91

Figure 4.21: The DT Constructed using the MinNumObj Pruning in the Thyroid Disease Dataset 91

Figure 4.22: Tree Size vs. Accuracy in the Thyroid Disease Dataset 92

Figure 4.23: The Average MC vs Accuracy in the Thyroid Disease Datas..92

Figure 4.24: The Average TC vs Accuracy in the Thyroid Disease Dataset..93

Figure 4.25: The ATC vs Accuracy in the Thyroid Disease Dataset 93

Figure 4.26: Tree Size vs. Accuracy in the Thyroid Dataset..... 94

Figure 4.27: The DT Constructed using the NumFold Pruning in the Thyroid Disease Dataset 94

Figure 4.28: The Average MC vs Accuracy in the Thyroid Disease Datase..95

Figure 4.29: The Average TC vs Accuracy in the Thyroid Disease Dataset..95

Figure 4.30: ATC vs Accuracy in the Thyroid Disease Dataset 96

Figure 4.31: The DT Constructed using the CF Parameter in the Breast Cancer Dataset..... 97

Figure 4.32: Tree Size vs. Accuracy in the Breast Cancer Dataset..... 97

Figure 4.33: The Average MC Vs Accuracy in the Breast Cancer Dataset..98

Figure 4.34: Average TC vs. Accuracy in Breast the Cancer Dataset 98

Figure 4.35: The ATC vs. Accuracy in the Breast Cancer Dataset..... 99

Figure 4.36: The DT Constructed using the MinNumObj Pruning in the Breast Cancer Dataset..... 99

Figure 4.37: Tree Size vs. Accuracy in the Breast Cancer Dataset..... 100

Figure 4.38: The Average MC vs Accuracy in the Breast Cancer Dataset..100

Figure 4.39: The Average TC vs Accuracy in the Breast Cancer Dataset...101

Figure 4.40: The ATC vs Accuracy in the Breast Cancer Dataset..... 101

Figure 4.41: Tree Size vs. Accuracy in the Breast Cancer Dataset..... 102

Figure 4.42: Tree Size vs. Accuracy in the Breast Cancer Dataset..... 102

Figure 4.43: The DT Constructed using the NumFold Pruning in the Breast Cancer Dataset..... 103

Figure 4.44: The Average TC vs Accuracy in the Breast Cancer Dataset..103

Figure 4.45: The ATC vs Accuracy in the Breast Cancer Dataset..... 104

Figure 4.46: Use of faster_pareto2 Function in Matlab..... 116

Figure 4.47: Non-dominated Solutions in HD when using the CF 121

Figure 4.48: Non-dominated Solutions in HD when using the MinNumObj 121

Figure 4.49: Non-dominated Solutions in HD when using the NumFold.. 121

Figure 4.50: Non-dominated Solutions in Thyroid when using the CF 122

Figure 4.51: Non-dominated Solutions in Thyroid when using the MinNumObj 122

Figure 4.52: Non-dominated Solutions in Thyroid when using the NumFold.....122

Figure 4.53: Non-dominated Solutions in Breast Cancer when using the CF.....123

Figure 4.54: Non-dominated Solutions in Breast Cancer when using the MinNumObj 123

Figure 4.55: Non-dominated Solutions in Breast Cancer when using the NumFold 123

Figure 5.1: Modeling the Relationship between the ATC and the Accuracy 133

List of Abbreviations

ATC	Average Total Cost
AUC	Area Under the Curve
CF	Confidence Factor
CS-C4.5	Cost Sensitive C4.5
DT	Decision Tree
FN	False Negative
FP	False Positive
HCI	Human Computer Interaction
HD	Heart Disease
KDD	Knowledge Discovery in Databases
MC	Misclassification Cost
MD	Data Mining
MinNumObj	Minimum Number of Objects
ML	Machine Learning
NN	Neural Network
NumFold	Number of Folds
OR	Operation Research
RoF algorithm	Rotation Forest algorithm
SBC	Selective Bayesian Classifier
SVM	Support Vector Machine
TC	Test Cost
TN	True Negative
TP	True Positive

List of Appendices

Appendix (1) Pseudo Code for the Decision Tree Algorithm

Chapter One

Introduction

1.0 Overview

A lot of works have been done in the field of cost sensitive learning in order to build inexpensive learning models but most of the built models have a degraded classification accuracy (Qiu, Jiang, & Li, 2017).

In addition, many other researches focused on increasing the classification accuracy rather than reducing the cost, including attribute selection cost, total misclassification cost and total test cost (Kong, Jiang, & Li, 2016).

Balancing between classification cost and accuracy is an important issue in many fields. For example, in the medical diagnosis field, an agent must balance between the cost of a test and the contribution of a test toward an accurate diagnosis. It means that the agent must determine when further tests worth their costs. On the other hand, when tests are inexpensive compared to the classification error cost, it becomes logical to perform all tests. Therefore, the cost should be assigned to tests and misclassifications as well (Turney, 1995).

This research aims at balancing the classification cost and the accuracy of data mining models built using decision trees (DTs). The behavior of a DT will be analyzed in terms of the average total cost of classification and the accuracies across various datasets in order to find a way for achieving sufficient classification accuracy at the lowest possible classification cost. Several experiments will be conducted on multiple datasets to verify and compare the results and the results will be discussed in details.

1.1 Background of the Study

Data mining (DM) can be defined as "the non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Freitas et al., 2007). Most DM techniques rely on inductive learning (i.e. learning by example), which help in inferring general rules and predicting future activities from previous knowledge (AlMana & Aksoy, 2014).

Inductive learning algorithms including DT algorithms have been successfully build classification models (classifiers) with high classification accuracy (Qiu, Jiang, & Li, 2017). However, those algorithms do not consider different types of classification costs, including the test cost (TC) and the misclassification cost (MC) when constructing the tree (Kong et al., 2016). Where MC is the cost incurred by misclassification error and TC is the cost of acquiring the attribute values, also called attribute cost or feature cost (Ling et al., 2004).

Although other classification costs are also available such as human–computer interaction costs, intervention cost*, and computation cost, Turney (2000), and Chen et al. (2016) mentioned that MC and TCs are the most important costs among all. Therefore, throughout the research the total of TC and MC will be used to refer to the total classification cost.

In real world applications, the cost of DM can be very high especially if 100% accuracy is required. As 100% accuracy is not required in most of the cases such as in marketing where customers are not required to be

*Cost of intervention is the cost of changing the values of attributes that were used in classification.

classified 100% correctly, the cost of DM in this case is much lower (He, et al., 2017).

In cost sensitive learning, the focus is to build inexpensive and reliable inductive learning models to perform a particular task. However, most of the cost sensitive learning algorithms have not only decreased the total classification cost but also decreased the classification accuracy as well (Qiu, Jiang, & Li, 2017).

In this study, the behavior of the DT will be analyzed in terms of the average total cost of classification (ATC) and the accuracy across various datasets in order to find a way for achieving sufficient classification accuracy, which will be determined throughout this research based on the experimental results, at the lowest possible ATC.

1.2 Problem Statement

Decision Tree (DT) is a powerful supervised DM technique used for classification and prediction. Even though the traditional DT algorithms such as ID3, C4.5, CART, etc. proved their success in maximizing the classification accuracy, they do not consider different cost types.

In cost-sensitive learning, the main focus is to minimize the total classification cost, that is the sum of the MC and TC, instead of increasing the classification accuracy. However, few works were directed toward reducing the total classification cost while having sufficient classification accuracy at the same time (Kong et al., 2016). Some researchers in the field of test-cost sensitive DT tried to balance between the classification accuracy and TC. They were successful in improving the test-cost sensitive version of ID3 and C4.5 algorithms by reducing the total TC compared to the original C4.5 algorithm but the classification accuracy

was degraded as well (Qiu, Jiang, & Li, 2017). In medical diagnoses and other fields, reducing the cost is not the only concern, achieving higher classification accuracy is also important.

This research aims at finding a proper tradeoff between classification accuracy and the ATC of building a DT. The DTs behavior will be studied in terms of the ATC and the accuracy across various medical datasets.

1.3 Objectives of the Study

The research objectives are:

1. To model the relationship between the ATC and the classification accuracy of a DT classifier.
    - To determine the optimum balance between classification accuracy and the ATC.
    - To determine the percentage of saving in ATC that gives satisfactory accuracy.
2. To determine whether the accuracy and ATC of the classifier deviate significantly on multiple datasets.
    - To study the impact of the dataset characteristics on the accuracy and ATC of the constructed DTs.
    - To study the behavior of a DT using different pruning methods.

1.4 Research Questions

This research aims at modeling the relationship between the ATC and the accuracy using DTs. Following are the research questions for the study:

Main question: Can the relationship between the ATC and classification accuracy in DT classifier be modeled?

Minor question (1): What is the optimum balance between the classification accuracy and the ATC?

Minor question (2): What is the percentage of saving in ATC that gives satisfactory accuracy?

Minor question (3): Using different pruning methods, are the accuracy and the cost of a classifier significantly affected by the dataset properties?

1.5 The Model of the Study

The proposed model for the study will be as follows:

1. **Literature Review**: In this stage, the problem of cost sensitive learning will be studied, various costs involved in the inductive learning process will be defined and C4.5 algorithm will be studied to define the cost measure used in it.
2. **Data Selection**: The data will be selected from UCI ML repository for benchmarking purposes, as UCI repository contains different types of benchmark datasets on different fields such as business, engineering, social sciences, life sciences, and many others. Many researchers rely on this repository for obtaining datasets to verify their research results.
3. **Dataset Cleaning**: As some datasets may have missing values, all records with missing values will be deleted.
4. **Conducting Experiments**:
    i. J48 algorithm will be used to build a DT from the training dataset using Weka.
    ii. The ATC and accuracy will be measured at different stages of the constructed DT.
5. **Modeling the Accuracy/cost Relationship of a DT classifier:**

i. A model will be constructed to illustrate the ATC against the classification accuracy at different stages of the tree construction.
ii. The model will be used to determine the optimum balance between cost reduction and accuracy using one of the optimization approaches: Pareto dominance approach.

6. **Evaluation and Comparison:**
   i. The experimental results will be presented, analyzed, and discussed.
   ii. The correlation analysis will be conducted to verify the relationship between the ATC and the accuracy and to determine the significance of the relationship.
   iii. The correlation and multiple regression analysis will be conducted to determine the impact of the dataset characteristics on the accuracy and the ATC of a classier.
   iv. Series of experiments will be conducted on various datasets to validate the results.
   v. The research results will be compared with other researchers' results to verify their effectiveness.

Figure 1.1 shows the steps to be followed in conducting the research:

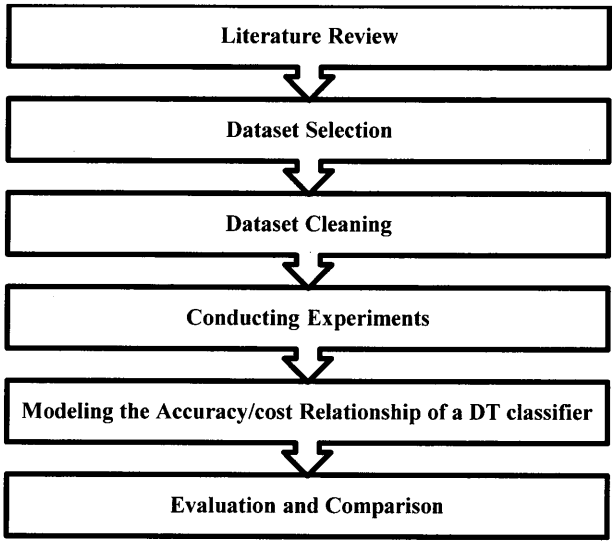


Figure 1.1: Research Methodology

1.6 Significance of the Study

The significance of the study comes from the followings:

- Constructing a model that relates the two main classification cost types with the classification accuracy.
- Making proper trade-off between maximizing the classification accuracy and minimizing the classification cost in DTs using one of the optimization approaches.

1.7 Scope and Limitations of the Study

The scope and limitations of the study can be summarized as follows:

- The behavior of DTs will be studied using C4.5 algorithm on three different datasets, as it is known for its high classification accuracy and as it is considered as one of the top ten algorithms in DM (Wu & Kumar, 2009).

- The ATC of a DT classifier will be modeled against the classification accuracy.
- The experimental datasets will be collected from UCI ML repository (UCI Machine Learning Repository, 2018).
- The two types of costs that will be considered are TC and MC.

1.8 Structure of the Thesis

The contents of the thesis will be as follows:

Chapter 1 Introduction

Chapter 2 Literature Review

Chapter 3 Research Methodology

Chapter 4 Results & Analysis

Chapter 5 Conclusion and Recommendations

1.9 Definition of Terms

Attribute: also called a feature that describes an instance in a dataset (Berry & Browne, 2006).

Classification Accuracy: the percentage of instances, from the testing set, that were correctly classified by a classifier (Kamber, 2007).

Cost-sensitive learning: an inductive learning methods that take into consideration different cost types (Ling et al., 2004).

Data Mining (DM): the process of extracting useful and understandable pattern in data (Freitas et al., 2007).

Inductive learning (learning by examples): the process of building a model through generalization from an adequate number of training examples and apply it on future/unseen examples (Rokach & Maimon, 2008).

Misclassification Cost (MC): the cost incurred when an instance not correctly classified (Marbán, et al., 2008).

Test Cost (TC): the cost of obtaining the value of attributes in a dataset (Marbán, et al., 2008).

Total classification cost: the total of MC and TC (Turney, 1995), (Ling, Sheng, & Yang, 2006), (Wang et al., 2010).

1.10 Conclusion

This chapter started with an overview of the research and gave a background of the study. Then, it discussed the problem statement and the reason behind conducting this research.

This chapter also presented the research objectives and questions that will be answered throughout the research. The main objectives of this research are to model the relationship between the ATC and the classification accuracy of a DT classifier, to determine the optimal balance between the two, and to determine the percentage of reduction in the ATC that gives satisfactory accuracy. It also aims at examining the impact of the dataset characteristics on the accuracy/cost pairs using different DT pruning methods.

The research methodology was also discussed in details. It composed of six stages, including the literature review, data selection, conducting

experiments, modeling the accuracy/cost relationship in a DT classifier, and evaluation and comparision stages.

The chapter also presented the research significance, scope, and the structure of the thesis. Five chapters will be included in this research. The chapter was concluded with the definition of the main terms contained in this research.

The following chapter will give a theoretical background of the study and will discuss other researchers work in the field of cost-sensitive learning.

Chapter Two

Classification Cost vs. Accuracy: A Theoretical Background

2.0 Overview

This chapter will present the general concept of Data Mining (DM) and its essential role in discovering knolwedge in databases. In section 3, the DM's vast area of usage in many applications including: education, telecommunication, intrusion detection, business, and others will be discussed.

The reminder of this chapter is organized as follows: section 4 will present the taxonomy of DM methods namely: classification, clustering, assocition rules mining, and regression with stating some of the classification algorithms. Section 5 will introduce the most popular DM tools. Section 6 will discuss different types of cost involved in the DM process in general and in inductive learning process in particular. Section 7 will introduce the relationship between different classification cost types and the the accuracy of a classifier. Finally, section 8 will discuss the research work been conducted related to this research.

2.1 What is Data Mining?

Freitas et al. (2007) defined Data mining (DM) as "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". Most DM techniques rely on inductive learning (learning by examples) where a model is built through

generalization from an adequate number of training examples and applied on future/unseen examples (Rokach & Maimon, 2008).

DM is also considered as an essential stage in the Knowledge Discovery in Databases (KDD) Process (Kamber, 2007). The following section discusses different stages of the KDD process.

2.2 Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) can be defined as "an automatic, exploratory analysis and modeling of large data repositories" (Maimon & Rokach, 2005). It aims to extract useful, understandable, and new knowledge from large and complicated data repositories. DM is considered as the core of KDD as it analyzes and models large data using association, classification, clustering and other algorithms (Figueiredo et al., 2016).

The first stage in the KDD process is domain understanding and goal setting to have a good understanding of the end-user goals and to have a good background regarding the domain where the KDD process will be implemented (Tomar & Agarwal, 2014).

The second stage is selecting data and creating databases to ensure the data availability, get additional important data, and put all the data together (Cheng et al., 2018).

The third stage is to use the pre-processing techniques to resolve inconsistencies, missing and irrelevant data (Cheng et al., 2018). Pre-processing techniques include data cleaning, handling missing data, handling imbalanced datasets and outlier or noise removal. Different

methods can be used to handle missing values in a dataset including filters, imputation, and embedded methods in DM algorithm itself such as DTs.

As imbalanced datasets* affect the performance of the classifier by making it biased towards the class with the large number of instances, it should be handled to resolve this issue. Imbalanced datasets can be handled by either sampling or algorithm modification methods. In the sampling method, data instances are adjusted in such a way so that each class will have the same number of instances. This can be done by removing some instances from the majority class, this technique is known either as "under sampling", or by adding duplicated entries for the minority classes, which is known as "over-sampling". In the algorithm modification method, the classifiers are adjusted to solve the imbalanced dataset problem since it mainly affects the classifier's performance (Tomar & Agarwal, 2014).

Outliers can be handled by various methods such as clustering and use of regression functions (interested readers may refer to (Hodge & Austin, 2004)).

The next stage is the data transformation stage. It involves transferring the data to the required form for DM. The two common methods used for this purpose are dimension reduction and attributes transforming (Tomar & Agarwal, 2014).

The fifth stage is the data mining stage. It involves selecting the appropriate DM method such as clustering, association rules mining, classification, etc. A technique for extracting knowledge is then selected

---

* Imbalanced datasets are datasets with non-uniform distribution of data instances to the classes (Li et al., 2018).

such as DT or Support Vector Machine (SVM) depending on the selected DM method and the required precision. For example, DT can be used for classification but for higher precision, Neural Network (NN) could be used (Maimon & Rokach, 2005).

In interpretation and evaluation stage, the extracted patterns and models are compared to the goals defined in the first stage and documented. In addition, the performance of the model is also evaluated in this stage using accuracy, error rate, computational complexity, precision, and some other metrics. In classification, the confusion matrix is used for this purpose (Tomar & Agarwal, 2014).

In the dissemination of knowledge stage, the discovered knowledge from previous stages is integrated and visualized using any of the visualization tools (Maimon & Rokach, 2005).

Figure 2.1 summarizes the different stages involved in the KDD process (adopted and modified from (Figueiredo et al., 2016)).

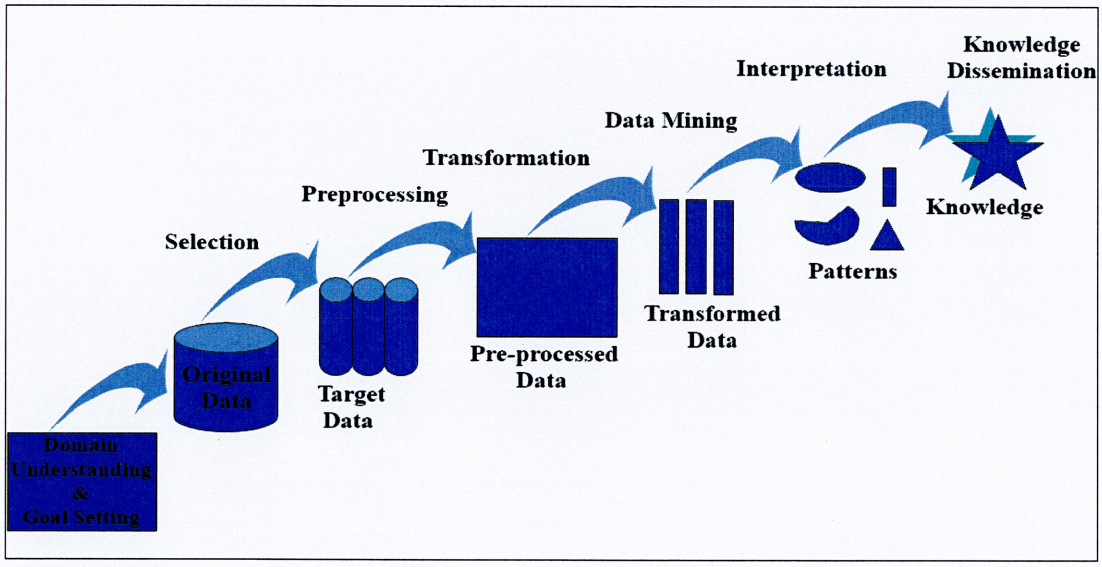


Figure 2.1: The KDD Process

2.3 Data Mining Applications

DM has been successfully utilized in applications of different areas including intrusion detection, education, telecommunication, business, and some other areas.

One of the most successful applications of DM is in intrusion detection systems to detect new attacks to network resources (including user accounts, file systems, system kernels, and others) (Ohta et al., 2008).

DM is also applied in education, which is referred to as "Educational DM". Asif et al. (2017) used DM techniques to predict the student performance progresses over the four years using DT and *k*-means algorithms. Mining educational data can also be used for detecting unwanted student behavior, grouping similar students together to create teams in a course project, helping educators in planning for future courses, recommending courses to students and in many other applications (Bakhshinategh et al., 2017).

DM also have been utilized by the telecommunication industry. It has many applications in this area including: detecting network problems, enhancing marketing by offering telecommunication services based on the customer usage obtained from mining call data (Weiss, 2005), and identifying fraudulent users and their usage patterns (Kamber, 2007).

Other applications of DM in business include market segmentation, customer churn, fraud detection, market basket analysis, and trend analysis (Sharma, 2016).

In biological DM, biological data including DNA, diseases, health records, biomedical images and others are explored to discover interesting

patterns and association. It can help insurers in detecting fraud and abuse and help physician in predicting different types of disease and their causes efficiently (Hussain, 2017).

Dogan and Tanrikulu (2012) mentioned that as DM systems are very important in areas like customer relationship management, credit evaluation, risk evaluations, medical treatments, classifiers in DM projects must be selected carefully, in terms of accuracy and speed.

2.4 Taxonomy of Data Mining Methods

Different methods were developed for the DM task. Figure 2.2 shows the two major types of DM methods, namely predictive or descriptive (Niha, 2017). Predictive DM refers to supervised methods that infer the existing dataset and predict unknown values in a new dataset. It includes classification and regression. Descriptive DM refers to unsupervised methods that focus on visualizing different features in a dataset including clustering (Tomar & Agarwal, 2014). The following subsections will discuss these methods in more details.

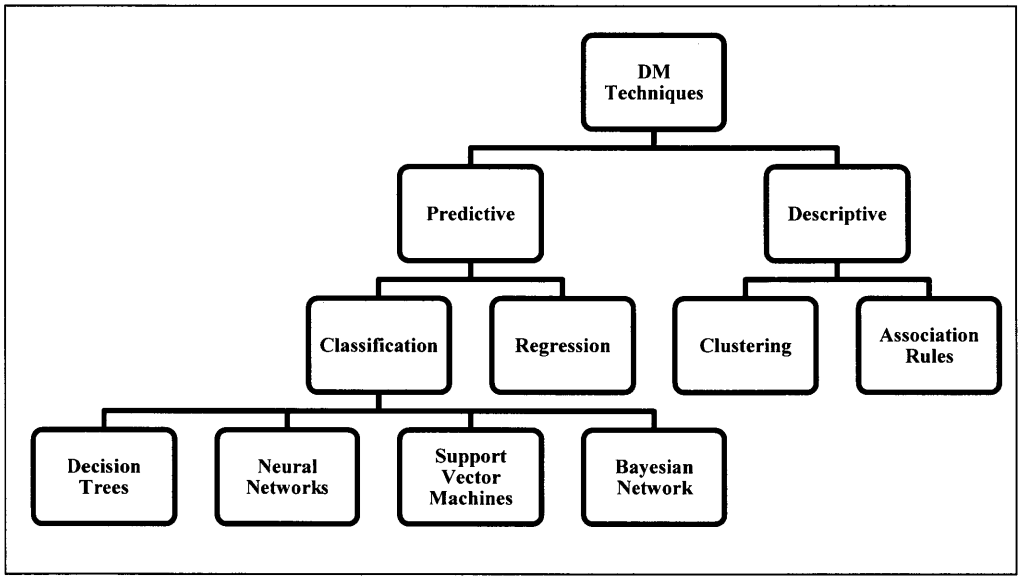


Figure 2.2: Taxonomy of Data Mining Techniques

2.4.1 Classification

Classification is one of the most widely used methods in various DM applications. The input for the classification is a training dataset where the class label is known for each instance and the output is a model that can assign a class lable to unlabled instances. The model can be presented in diferent ways including: classification rules (IF_THEN), decision tress, Neural Networks, or mathematical formulaes (Kamber, 2007). Classification is a supervised learning as the class label is known for the training set (Hussain, 2017).

Several classification algorithms exist including DTs, Neural Networks, genetic algorithm (GA), Support Vector Machines (SVM), and Bayesian classifiers (Hussain, 2017).

The *k*-fold cross-validation is commonly used in classification to improve the classifer accuracy. In *k*-fold cross-validation, the dataset is divided into *k* subsets. At each stage of the learning process, *k-1* subsets are used as the training set to build the classifier (model), while the remaining subset is used to test the accuracy of the constructed classifier, and the process reapeated *k* times, each time with different training and testing sets. The result is then combined together to construct the final classifier. The classifier's accuracy is then computed in percentage based on the number of objects classified correctly in the testing set (Bhowmick et al., 2009).

Classification has many applications in real world. Common applications include: fraud detection, credit card analysis, and medical diagnoses applications, among others (Hussain, 2017).

2.4.1.1 Performance Measures for a Classifier

Various measures are used for evaluating the correctness of a classifier. Confusion matrix is a tool that is used for this purpose. It is a table of size $m \times m$, given m classes, that evaluates the ability of a classifier to recognize objects of different classes (Kamber, 2007). It combines four different counts in case of binary classification where an object classified as one of two class labels. Those counts are: the number of positive objects classified correctly, also called true positives (TP), the number of negative objects classified correctly, also called true negatives (TN), the number of negative objects incorectly classified, also called false positives (FP), and the number of positive objects incorrectly classified, also called false negatives (FN) (Sokolova & Lapalme, 2009). Table 2.1 shows a binary confusion matrix.

Table 2.1: Confusion Matrix for Binary Classification

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

The most commonly used perfonmance measures rely on the values of the confusion matrix. Those measures includes: accuracy, precision, sensitivity (recall), specificity, and Area Under the Curve (AUC). Table 2.2 explains the definition and the way of computing each of these metrics (Kamber, 2007).

Table 2.2: Classifier's Evaluation Metrics

Measure	Definition	Formula
Accuracy	The percentage of objects from the testing set that were correctly classified by the classifier.	$\frac{TP + TN}{TP + FN + FP + TN}$
Precision	The percentage of positive objects correctly classified divided by the total number of positive objects classified by the system.	$\frac{TP}{TP + FP}$
Sensitivity (Recall)	The percentage of positive objects that were correctly recognized.	$\frac{TP}{TP + FN}$
Specificity	The percentage of negative objects that were correctly recognized.	$\frac{TN}{FP + TN}$
AUC	The ability of a classifier to avoid incorrect classifications.	$\frac{1}{2} \times (\text{Sensitivity} + \text{Specificity})$

2.4.1.2 Classification by Decision Tree Induction

Decision trees (DTs) are one of the most popular algorithms for classification in many application domains (Li, et al., 2018). Ohta et al. (2008) defined DTs as a "tree-shaped diagram representing a sequential decision process in which attribute values are successively tested to infer an unknown state".

As shown in Figure 2.3 (Farid, Zhang et al., 2014), a DT is composed of a root node, internal/test nodes and leaf nodes having a class or a label. The root node has no incoming edges, while all other nodes have only one incomming edge. Internal nodes also have one or more outgoing edge (Rokach & Maimon, 2008). Within each internal node the value of the attribute assigned to the node is tested to determine the next node to go to along the path, and when a leaf node is reached, its label will represent the current class (Ohta et al., 2008).

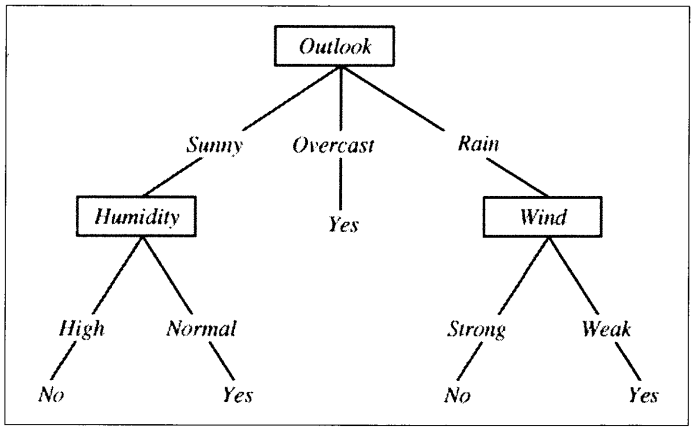


Figure 2.3: DT Example

DT algorithms include ID3 (the standard DT algorithm), C4.5, and CART algorithm. C4.5 is one of the top ten DM algorithms as mentioned by Wu and Kumar (2009) in their book 'The Top Ten Algorithms in Data Mining'.

DT has various advantages, some of its advantages include: easiness to understand, applied to various real world problems successfully, its ability to build classifiers from datasets containing numerical and non-numerical data, and easiness to be converted to classification rules (Kamber, 2007). In addition, DTs are most commonly used for classification due to their computational efficiency (Chen, Wu, & Tang, 2016), they require no domain knowledge or parameter settings and they provide high classification performance (Li, et al., 2018). On the other hand, DTs have some disadvantages. Specifically, any change made in the training data results in a change in the attribute selections and affects the whole tree accordingly. In addition, the output attributes must be single and categorical (Roiger & Geatz, 2003).

2.4.1.3 Decision Tree Construction Algorithms

Several DT algorithms exist to construct or model DTs. This section reviews the most popular DT algorithms namely: ID3, C4.5, and CART.

i. Iterative Dichotomiser (ID3)

ID3 is the standard DT algorithm developed by J. Ross Quinlan in 1986. In ID3, only a subset of the dataset is randomly selected to build the DT, while the remaining data are used to test the tree. If the constructed tree is able to classify all testing records correctly, then the tree is considered correct and the process finishes. Otherwise, another subset of the dataset including some of the incorrectly classified records is used to construct a new tree. The process continues until all records in the dataset are classified correctly by the constructed tree (Urso et al., 2018).

ii. C4.5 Algorithm

C4.5 is an improved version of ID3 by Quinlan in 1993. C4.5 algorithm has some advantages over ID3 in that it can handles both numerical and categorical attributes. It can also handle the missing attribute values. It makes the attribute selection using information-based criteria and uses pruning to trim the constucted tree inorder to handle the overfitting problem that may present in DTs (Urso et al., 2018). The psudo code for C4.5 algorithm is provided in appendix 1 (Rokach & Maimon, 2008).

iii. Classifier and Regression Trees (CART)

CART is also a DT algorithm developed by Breiman in 1984 (AlMana & Aksoy, 2014). It can perform classification and regression tasks based on the nature of the input data, whether categorical or numerical data, where the tree leaves -in case of regression trees- predict real numbers rather than

classes (Urso et al., 2018). Like C4.5, prunning is also used in CART to handle the overfitting problem. However, unlike ID3 and C4.5, CART allows for costructing binary trees only, where each internal node has exactly two outgoing edges. In CART algorithm, the attribute with the highest level of homogeneity is selected at each node (Berry & Browne, 2006).

2.4.2 Regression

Regression is also a supervised method that is normally used for numeric prediction (Rokach & Maimon, 2008). Different types of regression models exist: linear regression, nonlinear regression, multivariate linear regression, and multivariate nonlinear regression (Niha, 2017).

It differs from the classification methods in that it maps the input into a real-valued domain while classification methods map the input into a class label (Rokach & Maimon, 2008).

2.4.3 Clustering

Clustering is the process of grouping instances, so that instances in one group have similar characterstics and differ from instances in other groups (Kamber, 2007). It is also called unsupervised learning as the class lable is not provided for each instance. The most common clustering algorithms are: *k*-means, OPTICS, and STING. They can be used for organizing and categorizing data, data compression, and outlier detection (Hussain, 2017).

2.4.4 Association Rules Mining

This method aims at finding relations among attributes in large datasets. The most common application of association rule mining is market basket analysis to identify items that the customer likely buy together by analyzing his/her shopping cart. The most common association rules algorithms are Apriori, AprioriTid, and Tertius algorithms (Sharma, 2016).

2.5 Data Mining Tools

There are many open source tools used for DM. The most popular DM tools are RapidMiner, Weka, R-Programming, and Orange software.

The Waikato Environment for Knowledge Analysis (Weka) is the DM tool that will be used in this research. It is an open source DM tool written in Java developed by Waikato University in New Zealand. It is free under the General Public License (GPL), which gives it an advantage over its counterpart "RapidMiner" (Kulkarni & Kulkarni, 2016). It can be customized by users and have pre-processing, post-processing, and visualization tools. It can perform most of DM tasks such as classification, clustering, regression, feature extraction, visualization, and many others using C4.5, ID3, *K*-means, and Apriori algorithms. Weka is user-friendly and easy to manage due to its graphical user interfaces (Du, 2010), (Weka 3: Data Mining Software in Java, n.d.).

2.6 Costs Involved in the DM Process

Different types of cost are involved in the DM process including data cleaning cost, data integration cost, transformation costs of the integrated

data into useful datasets for the DM and the cost of building and running the DM model (Pipino, 2004).

The following subsections will discuss the different types of cost involved in building a DM model and different cost units.

2.6.1 Types of Costs Involved in Inductive Learning Process

Different types of costs are involoved while building a DM model. Turney (2000) presented a taxonomy for all possible cost types included in inductive learning, where each instance is assigned to a class label. According to Turney's taxonomy the cost types are: the misclassification cost, test cost, intervention cost, computation cost, and human-computer interaction costs.

2.6.1.1 Misclassification Cost (MC)

The misclassification cost is the cost incurred when an instance is not classified correctly, i.e. the penalty of classifying an instance in class X while it belongs to class Y (Marbán et al., 2008).

A classifier may infer wrong class. As discused in section 2.4.1.1, the two types of errors that may arise are false positive (FP) and false negative (FN). The classification cost matrix (refer to Table 2.3 for an example (Deng & Song, 2014)) is used to estimate the cost of FPs and FNs and calculate the MC (Ohta, Kurebayashi, & Kobayashi, 2008). It is a $c \times c$ matrix, where c is the number of different class lables. In classification cost matrix, the cost of correct classification is usually considered zero (Li, et al., 2018). The cost of FPs and FNs commonly considered the same but they might have different values in some cases (Turney, 2000).

Table 2.3: An Example of a Classification Cost Matrix

Actual Class \ Predicted Class	Good	Bad
Good	0	300
Bad	800	0

2.6.1.2 Test Cost (TC)

Test cost is the cost of obtaininig the value of an attribute (Marbán et al., 2008). For example, in classifying a medical disease, a patient might be described in terms of temperature, blood test, pulse rate, and others, which have their associated monetary cost (Zhao et al., 2015). It worths to incur the cost of all tests if the MC is much higher than the cost of tests (Turney, 1995).

Turney (1995) stated that the TC can be calculated by using the TC vector, which lists all possible tests and their corresponding cost. It can be calculated in terms of money spent, time needed to collect data, implementation difficulty of a test, or other required resources as shown in Table 2.4 (Zhou, Zhou, & Li, 2016). The TC may depend on the time it was done, a preceding test result, for example, the cost of a blood test may depend on the patient age, or other conditional circumenstances (Turney, 2000).

The MC and TC can be defined to have the same cost scale, such as dollars (Ling et al., 2004).

Table 2.4: Some Medical Tests and their Costs

Cost / test	G	S	ALT	AST	HBV-DNA	Gene-type	MTCT
Price(¥)	100–200	100–200	89	89	140	390	–
Time(day)	3–7	3–7	1	1	2-3	4-5	–
Difficulty(0–10)	7	7	3	3	4	5	2
Side effect(0–10)	3–4	3–4	1	1	1	1	0

2.6.1.3 Intervention Cost

The Intervention cost refers to the cost of changing the values of an attribute that were used in the classification process (Marbán et al., 2008). For example, in a petroleum manufacturing, if the induced rule says, "If sensor's X value is more than sensor Y, then the return of product Z will increase". Therefore, to increase the return of product Z, we might intervene the process so that X will always have value more than Y. The intervention process has an associated cost represented by the effort needed to change the feature values (Turney, 2000).

2.6.1.4 Unwanted Achievement Cost

The cost of unwanted achievements is the cost of errors obtained due to changing some attribute values, i.e. intervention. In the example given in the previous subsection, if the induced rule was used to make intervention in the manufacturing process so that X is always greater than Y, then any time the rule fails to increase Z's return will have an associated cost, also called "the cost of using the rule".

The cost of unwanted achievement can be constant. In this case, the classification cost matrix can be used to calculate it in a similar way as in MC. The cost of unwanted achievement can also be conditional depending on many factors such as the quantity of unwanted achievement, the cost

of repairing, disposing or refining the problem caused by unwanted achievement (Turney, 2000).

2.6.1.5 Computational Cost

Turney (2000) also defined the computational cost as the cost of the computer resources used in the inductive learning process. This include the size complexity, time complexity (execution time) and space complexity (memory space) used by a classifier. The cost of computational complexity might be endured during training, when learning the algorithm to classify, or during testing, when using the algorithm for prediction. For example, Neural Network (NN) algorithm has high computational cost during training and low computational cost during testing (Turney, 2000).

The computational complexity can be measured in different ways. The size complexity can be measured in terms of the number of lines of code or the number of bytes, as codes consume memory space, which have an associated cost. The time complexity can be measured in terms of the number of read/write head movements or by the frequency of change in read/write head direction. Last, the space complexity can be measured in terms of memory space needed for execution.

Dogan and Tanrikulu (2012) mentioned that the computational cost represents the classifier's speed, that is the CPU time used by the classifier to build the model or the time (in seconds), computational resources, and memory required to build the model.

Jensen et al. (2015) defined the computational cost as the number and type of mathematical operations, including arithmetic and comparison operations. They estimated the number of machine cycle required by the

CPU for each operation to calculate the computational cost. They defined the memory demand cost as the number and type of constants and variables needed.

2.6.1.6 Human-computer Interaction Cost

The human-computer interaction (HCI) cost is the cost of human involved in the learning process. This includes the cost of attributes selection, the cost of setting the algorithm parameters, the cost of converting the data to the appropriate format required by the algorithm, and the cost of analyzing the results of a model (Marbán et al., 2008). As Turney (2000) mentioned "we are not aware of any attempt to measure the HCI costs" which indicates that it is not an important cost type to consider. Table 2.5 summerizes the different types of costs involved in the inductive learning process and provides all reviewed researches that uses such type of costs.

Table 2.5: Type of Costs involved in inductive Learning Process

Cost Type	Definition	How to calculate?	References
MC	The cost incurred when an instance is not classified correctly.	The classification cost matrix is used to estimate the cost of FPs and FNs. It can be calculated in terms of money spent, time needed, or other required resources.	Turney (1995), Turney (2000), Ling et al. (2004), Marbán et al. (2008), Davis et al. (2006), Ohta et al. (2008), Deng and Song (2014), Zhao et al. (2015), Kong et al. (2016), Lu et al. (2017), Li et al. (2018)
TC	The cost of obtaining the value of an attribute.	The TC vector is used to estimate the TC. It can be defined to have the same cost scale as the MC.	Turney (1995), Turney (2000), Ling et al. (2004), Davis et al. (2006), Marbán et al. (2008), Zhao et al (2015), Zhou et al. (2016), Kong et al. (2016), Lu et al. (2017)
Intervention Cost	The cost of changing the values of an attribute that have been used in the classification process.	It can be estimated in terms of the effort needed to change the feature values.	Turney (2000), Marbán et al. (2008)
Unwanted Achievement Cost	The cost of errors obtained due to changing some attribute values "intervention".	The classification cost matrix can be used to estimat it.	Turney (2000)
Computational Cost	The cost of the computer resources used in the inductive learning process including the size complexity, execution time and memory space used by a classifier.	It can be measured in different ways including the number of lines of code or the number of bytes, CPU time, or memory space required for execution.	Turney (2000), Bhowmick et al. (2009), Dogan and Tanrikulu (2012), Jensen et al. (2015), He et al. (2017)
Human-computer Interaction (HCI) Cost	The cost of human involved in the learning process.	It can be estimated in terms of the time and effort required for selecting the attributes to be used, setting the algorithm parameters, converting the data to the appropriate format required by the algorithm, and analyzing the model results.	Turney (2000), Marbán et al. (2008)

2.6.2 What Types of Cost to Consider?

Among all types of costs mentioned in the previous subsections, Turney (2000) mentioned that MC and TCs are the most important classification costs that must be considered.

Turney (1995), Davis et al. (2006), Ling et al. (2004), Wang et al. (2010) and Kong et al. (2016) defined the total classification cost as the total of TC and the MC.

In inductive learning, most of the work done focused on the success rate (or error rate). Some researchers focused on non-uniform MCs, that is assigning different costs for different error types, and other researchers focused on TC without considering the MC (Freitas et al., 2007). However, most of the researchers agreed that the two main cost types for building a classification model are the TC and the MC.

2.6.3 Calculating the Total Classification Cost in DTs

As the DT algorithim will be the focus of this research, this section will discuess in details the way of calculating the two major classification cost types, namely MC and TC, in DTs.

Turney (1995) explained the method used for calculating the total classification cost for a DT, having a set of testing cases. It can be applied to all standard classification DTs, such as those created using EG2, CS-ID3, IDX, C4.5, and ICET algorithms.

Turney (1995), Xu, Min, & Zhu (2013) and Zhao et al. (2015) mentioned that the total classification cost of a DT is determind by calculating the average total cost (ATC) of the testing cases, where the dataset is divided into training set and testing set. The ATC is calculated by dividing the total classification cost (sum of MC and TC) for all testing cases by the number of testing cases as follows (Zhao et al., 2015):

$$ATC(U) = \sum_{x \in U} \frac{tc(x)+mc(x)}{|U|} \qquad 2.1$$

where x is a testing case, U is the testing dataset, $tc(x)$ and $mc(x)$ are the TC and MC of the testing case.

To calculate the TC of a certain case, the cost of each test along the path from the root to the leaf in a DT is added together. If the same test appears twice, it should be added only once (Turney, 1995). This is logical as in medical tests, patients are charged only once for a particular test (such as heart pulse rate) even though its results used to make various decisions. Figure 2.4 shows an example. If the set of attributes on the path of a testing case are *a2* and *a5* and their TCs are 1 and 7 from the TC vector shown in Table 2.6 (Xu, Min, & Zhu, 2013), their total TC will be 1+7= 8.

Table 2.6: The TC Vector

Attribute	a1	a2	a3	a4	a5	a6	a7	a8
TC	4	1	4	1	7	7	8	9

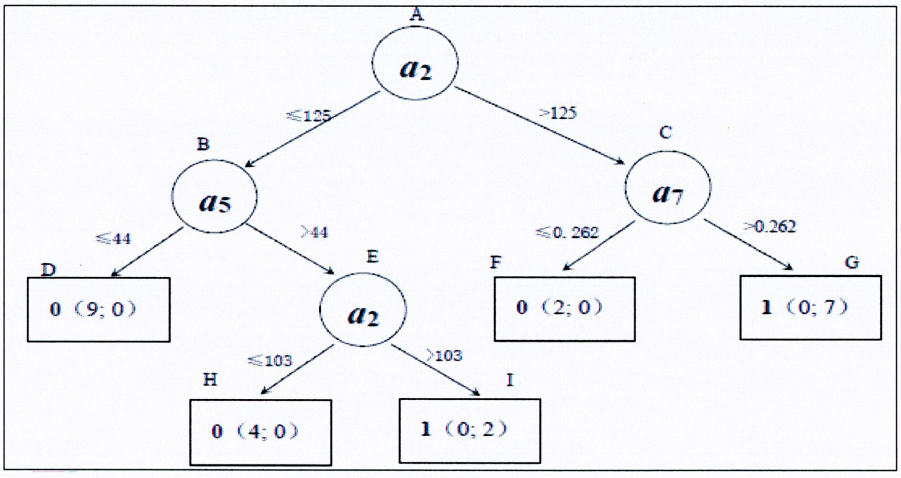


Figure 2.4: A DT Classifier (Xu, Min, & Zhu, 2013)

Refering to Figure 2.4, nine cases followed the path having *a2* and *a5* attributes. Therefore, their total TC will be 9×(1+7)= 72.

The total TC for the whole tree will be : 9×(1+7)+ 4×(1+7) + 2×(1+7)+ 2×(1+8)+ 7×(1+8)= 201 and the average total TC for the whole 24 testing cases= 201/24= 8.375.

The MC is calculated using the classification cost matrix (refer to Table 2.3). Having the predicted class of the case (the tree leaf) and the actual class of the case, the matrix used to determine the MC. This cost is added to the TC to get the total classification cost.

2.6.4 Data Mining Cost Units

The cost of building a DM model is domain-specific and can be measured in different ways (Turney, 1995). It can be measured in monetary units (dollars), time, or resources needed. In medical diagnosis, cost may include the quality of life of a patient. While in image recognition, cost may represent the CPU time needed for computation (Turney, 2000).

He et al. (2017) mentioned that the DM costs can be represented by the computational time needed to run the algorithm and the computational resources used to run the algorithm such as EC2 VM instance in case of mining cloud data.

2.7 Relationship between Classification Cost Types and Accuracy

Many researchers realized the relationship between different classification cost types and the accuracy of a classifier. Lu et al. (2017) mentioned the necessity to embed different types of costs -namely MC and TC- in classifiers to reduce their classification cost while having more reliable results at the same time. They also mentioned that in most cases the correct classification has no follow up cost.

In addition, most DM algorithms are exponential in computational complexity. For example, the process of mining big data may take several hours or even days to finish and consumes huge computational resources, especially if 100% accuracy is required (He, et al., 2017). Ahmadi et al. (2012) also mentioned, "high accuracy is achieved at the expense of increased computational complexity" as they defined total classification cost in terms of the computational complexity of features using SVM classifier.

Turney (1995) mentioned that when all TCs are ignored (set to zero), the total average cost of using a DT will be:

$$Cost = p.k \qquad 2.2$$

where $p \in [0,1]$ is the frequency of errors in the testing set, and k is the error cost from the classification cost matrix. Note that a simple classification cost matrix was used, which means that all errors have the same cost.

From the above equation, Turney calculated the accuracy percentage on the testing set as: $100 \times (1-p)$, which means that there is a linear relationship between the average classification cost and the accuracy percentage, i.e. reducing the cost will increase the accuracy (note that only the MC considered in this case).

When Turney(1995) considered just the TC and tried to find its relationship with the accuracy, the hypothesis he made "as the TC increases the accuracy will also increase" was confirmed through experimentations.

Bohanec and Bratko (1994) also clarified the relationship between the DT size and the classification accuracy. They noticed through experimentation that as the DT size (number of leaves) increases, the classification accuracy also increases, as shown in Figure 2.5. However, they noticed that nine nodes (out of 21 nodes full size tree) are sufficient to provide an accuracy of 99.57%, i.e. the remaining 12 nodes account for an accuracy less than 0.5%. They also found that the classifier's accuracy is affected by the properties of a dataset. There was a big difference in classification accuracy for two trees of the same size built using two different datasets.

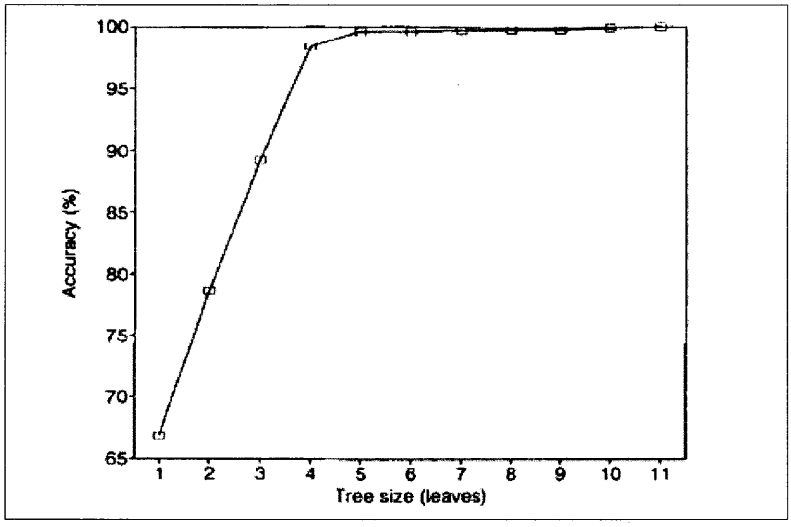


Figure 2.5: Accuracy vs. Tree Size

Bhowmick et al. (2009) confirmed the results of Bohanec and Bratko (1994), as they concluded that the larger the DT, the better the classification results, but the longer the time required to construct the classifier. The general rule they came up with is that: the cardinality of the feature set affects the cost of constructing and deploying the classifier. Zhou et al. (2016) also mentioned that adding more features in a DT will improve the accuracy but will also increase the cost of construction.

In addition, Maimon and Rokach (2005) clarified the relationship between the error rate of a DM model and the cost of building the model using the tradeoff curve shown in Figure 2.6 (Maimon & Rokach, 2005). The figure shows that as the cost of a model decreased (calculated in terms of the number of attributes used to build the model), the error rate will be increased. This also asserts that the relationship between the accuracy and TC is proportional as the error is the opposite of accuracy.

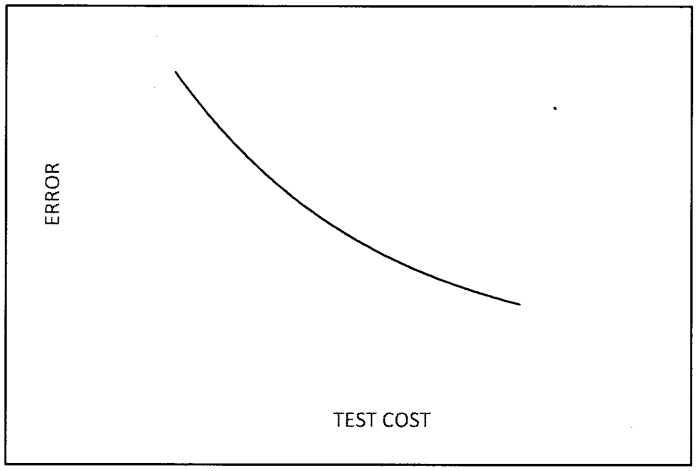


Figure 2.6: Error-cost Relationship in a DM Model

2.8 Cost-sensitive Models

Building a classifier to minimize the classification cost was the focus of many researchers, especially in the field of medical diagnosis. Many researchers proposed cost-sensitive algorithms to build DTs because in many cases it is more appropriate to minimize the classification cost than maximizing the classification accuracy. Some researchers considered the MC, while others considered the TC for building low cost classifiers. Few researchers considered both types of costs to reduce the overall classification cost. The following subsections review the work of some selected researchers.

2.8.1 Misclassification Cost Reduction

Many researchers focused on the MC only to reduce the total classifier's cost.

Sahin et al. (2013) proposed a new cost-sensitive DT algorithm that minimize the total MC of constructing a DT that can be used for detecting fraudulent transactions in credit cards. The new algorithm uses varying cost for each fraudulent transaction. The new algorithm was compared with some other DM algorithms such as the traditional DT, SVM and Artificial Neural Networks (ANN) algorithms that use fixed MCs. The performance of the models was measured in terms of the MC savings and the true positive rate (sensitivity). Although the new algorithm decreased the sensitivity by 0.5%, it also improved the MC savings approximately by 1%, which means in some cases, millions of dollars savings. However, the new algorithm just considered the MC and ignored the TC.

Li et al. (2018) also worked on minimizing the MC and maximizing the overall classifier performance of a DT of the minority classes in an imbalanced datasets. The accuracy of the proposed algorithm was compared with the existing cost-sensitive DT algorithms: IDX, CS-ID3, and CS-C4.5. The experimental results showed that the proposed algorithm improved the accuracy performance of the minority classes by 9.32%, 6.16% and 10.81%, respectively. The TC was not considered in their work.

2.8.2 Test Cost Reduction

Various researches focuced on minimizing the TC of a classifier, such type of researches is also called "test-cost sensitive learning" (Kong et al., 2016).

Davis et al. (2006) modified the existing C4.5 algorithm to build a DT with high accuracy and low TC. The extended algorithm was compared with the standard C4.5 algorithm. The results showed that the extended algorithm allows for high level of cost reduction (up to six times) compared to the cost-insensitive C4.5 with only 1% reduction in accuracy. However, they considered only the TC in calculating the overall classification cost.

Kong et al. (2016) modified the backward greedy search attribute selection method used in the Selective Bayesian Classifiers (SBC), which select attributes that maximize the accuracy regardless the cost. Their aim was to minimize the total TC while maintaining high classification accuracy. The expermental results showed that the total TC of the new algorithm was reduced from 5.9 to 1.67 (71.7%) on average compared to SBC algorithm with only 0.8% reduction in accuracy (83.26% for the new algorithm and 84.06% for SBC).

Zhou et al. (2016) proposed a new attribute selection method that considered the attribute "test" cost, assuming that different attributes have different associated costs. It selects the low cost of informative subset of attributes for constructing a DT inorder to minimize the overall TC of building the tree while having reasonable classification performance. The emperical results showed that the cost reduction reached 30% compared to the Random Forest (RF) and SVM algorithms with a very slight reduction in accuracy in some cases. Their proposed algorithm focuses only on the TC and does not consider the MC or any other cost types.

Qiu et al. (2017) used random attribute selection method, instead of greedy attribute selection used in standard DTs, to find the best attributes for each splitting in a DT. The aim was to balance between the classification

accuracy and efficiency, that is, the proportion of instances classified correctly and the cost of measuring attribute value. The new algorithm was tested on number of datasets collected from UCI and the results were compared with that of algorithm C4.5. The expermental results showed that the average classification accuracy of the new algorithm (82.73%) was almost the same as C4.5 (82.57%). In addition, the average total TC of the new algorithm (101.21) was much lower than that of C4.5 (146.73). However, the MC was not considered.

2.8.3 Misclassification and Test Cost Reduction

This section presents attempts of a number of researchers to minimize the total of MC and TC.

Ling et al. (2004) considered the sum of MC and TC in order to build a cost-sensitive DT. They used a new splitting criterion "the minimal total cost" for attribute selection in the tree construction, instead of minimal entropy used in C4.5. They also proposed a test strategy to minimize the TC in the test data, assuming that many of the test cases have missing values. Initially, they proposed four strategies, then they nominated the optimal one (the one that reduces the total cost to the lowest level).

When a test case goes down the tree and found unknown attribute value one of the following strategies will take place:

i. Strategy 1: Get the value of all unknown attributes sequentially in a tree built with minimal total cost of the training set.
ii. Strategy 2: Stop at the node with the unknown attribute value and use the ratio of positive and negative examples at that node from the training cases to predict the class of the test case.

iii. Strategy 3: Use C4.5 algorithm's strategy in dealing with unknown attributes, which is to proceed to all branches under the node with unknown attribute value and calculate the weighted sum of the class in each branch to determine the final class (interested reader may refer to the example in (Ling et al., 2004)).

iv. Strategy 4: Not to do any test, instead build a dynamic DT from the training examples for each test case using only the attributes with known values.

They found that the first strategy reduces the total cost more compared to others. The new algorithm was compared with C4.5 algorithm across five different datasets. The experimental results showed that the new algorithm have lowered the ATC approximately by 80% under different percentage of unknown attributes and under different TC values compared to C4.5. The complexity of both algorithms was similar. However, the classification accuracy was not considered.

Deng & Song (2014) incorporated the classification benefits, i.e. the benefits of making correct classification, in addition to the MC and TC to build a DT that is sensitive to all. The propsed algorithm was compared with C4.5 using a dataset with different missing attribute rates (between 10% to 60%). The experimental results showed that "as the missing attribute values increased, the missclassification cost will also increased". The new algorithm was successfully decreased the total classification cost up to 25% compared to C4.5. However, the researchers did not measure the accuracy of the proposed algorithm.

Zhao et al. (2015) proposed a method for constructing a cost-sensitive DT based on C4.5 algorithm. They focused on reducing the ATC, i.e. the average total TC and the average total MC, by using a propabilistic

pruning mechnism. They tested their modified algorithm using UCI datasets. The experimental results also showed that the pruning mechanisim reduced the TC without affecting the MC significantly. The ATC of the new algorithm was lower in all datasets compared to CS-C4.5, and CSGainRatio algorithms, but it was lower in five datasets (out of eight) compared to post pruning algorithm. The ATC of the new, post pruning, CS-C4.5, and CSGainRatio algorithms was 28.75, 31.9, 50.125 and 64.69, respectively. However, the accuracy of the extended algorithm was not measured and compared with other algorithms to varify its effectiveness.

Lu et al. (2017) extended the rotation forest (RoF) algorithm by embeding the two types of classification costs, namely MC and TC to improve the classification accuracy and reduce the average MC of a DT. The reduction in MC reached 30%, but the drop in accuracy also reached 10.2% compared to RoF algorithm. The drop in total classification cost was not measured in their research.

2.8.4 Computational Cost Reduction

Bhowmick et al. (2009) focused on the computational complexity of each features (i.e. the time required to compute each feature) to reduce the total computational complexity of constructing the classification model without sacrifying the accuracy. Their proposed method selects features in increasing order of their computational complexity. The experimental results showed that the new method speeded up the time for building the model by a factor of 125 on average while maintaining high accuracy as in SVM and NN.

He et al. (2017) made a trade-off between the accuracy and cost of mining big data on the cloud in order to determine the optimum balance that gives high accuracy at the lowest possible cost. They used the *k*-means algorithm as a case study and considered the computational cost of mining big data on the cloud. The experimental results showed that the algorithm consumed small amount of computational time to reach high accuracy, then it consumed large amount of computational time to improve the accuracy just by 1%. For example, when they set the *k* value to 32, the algorithm consumed 75.23% of the computational time to improve the accuracy from 99% to 100% and when *k* value was set to 64, it consumed 93.59% of the computational time to improve the accuracy from 99% to 100%. Thus, they recommended stopping the DM process at an earlier stage to save in computational time while having high accuracy.

Table 2.7 summarizes all reviewed research papers were done to minimize the total classification cost and maximize the classification accuracy in some classifiers.

Table 2.7: Algorithms for Maximizing accuracy and Minimizing cost

Reference	Classification Cost Type	Accuracy vs Cost	Algorithm Used
Ling, et al. (2004)	MC and TC	Lowered the ATC up to 80% compared to C4.5 but the classification accuracy was not measured.	DT
Davis, et al. (2006)	TC	High level of cost reduction (up to six times) with only 1% reduction in accuracy.	DT
Bhowmick, et al. (2009)	Computational cost	Speeded up the time for building the DT by a factor of 125 on average with high accuracy as in SVM and NN.	SVM, NN
Sahin, et al. (2013)	MC	Lowered the sensitivity by 0.5% but improved the MC savings approximately by 1%.	DT
Deng and Song (2014)	MC and TC	Lowered the total classification up to 25% compared to C4.5 but the accuracy was not measured.	DT
Zhao, et al. (2015)	MC and TC	Lowered the total classification cost by 9.9%, 42.6%, 55.6% compared to post pruning, CS-C4.5, and CSGainRatio algorithms, respectively but the accuracy was not measured.	DT
Kong, et al. (2016)	TC	The total TC was reduced by 71.7% on average with 0.8% reduction in accuracy compared to SBC algorithm.	Naive Bayes
Zhou, et al. (2016)	TC	The cost reduction reached 30% with very slight reduction in accuracy.	DT
Lu, et al. (2017)	MC and TC	The reduction in MC reached 30%, but the drop in accuracy also reached 10.2% RoF algorithm.	DT
He, et al. (2017)	Computational cost	Around 70% of saving in computational time with only 1% reduction in accuracy.	*K*-means
Qiu, et al. (2017)	TC	The total TC reduced by 31% while maintaining the same accuracy as C4.5.	DT
Li, et al. (2018)	MC	The accuracy of the minority classes improved by 9.32%, 6.16% and 10.81% compared to IDX, CS-ID3 and CS-C4.5, respectively.	DT

As it is clear, most of the above mentioned researches focued on balancing the TC and the classification accuracy or balancing the MC and the classification accuracy.

Only four researches found focusing on reducing both cost types and incrseaing the classification accuracy. However, they were successful in decreasing different types of cost but most of them have degraded the classification accuracy too or did not measure the accuracy at all. In this research, the relationship between the ATC and the classification accuracy will be modeled to determine the optimal balance between the two that give satsfactory results.

2.9 Conclusion

This chapter started with defining DM. Then it discussed the concept of KDD and the essential role of DM in the KDD process for extracting hidden, valid, and useful patterens. Different applications of DM were presented such as in education, telecommunication, intrusion detection, medical diagnosis and in business. The DM appilcation in business was found the most commonly used application of DM nowadays.

This chapter also discussed different DM methods including classification, clustering, associtions rules mining, and regression. In addition, metrics used for evalualting the performance of a classifier and some classification algorithms were also discussed in details as this research will focus on the classification method.

Moreover, Weka tool for DM was presented. Note that the Weka software is going to be used to conduct experiments in this research due to its various advantages such as: performing most of the DM tasks (classification, clustering, regression, feature extraction, and visualization), user-friendly and easy to manage due to its graphical user interface.

This chapter also introduced different types of cost involved in the DM process in general and in inductive learning process in particular. Among all costs, the MC and TCs were found the most important cost types in inductive learning that must be considered.

The relationship between the some of the classification cost types and the classification accuracy of a classifier was discussed. A propotional relationship was found between the TC and the accuracy and a proportional inverse relationship was found between the MC and the

accuracy. However, the researchers who worked on modeling the TC or MC against the accuracy have considered the same cost for all attributes or the same cost for all misclassifications, while in cost-sensitive learning different tests/misclassifications should be given different cost values. In addition, non of the researches found modeling the total of MC and TC against the classification accuracy.

Finally, the work related to the cost sensitive DM models were reviewed. It has been found that most of researchers either focused on balancing the TC and the accuracy or balancing the MC and the accuracy. Only a few researchers worked on balancing the total of MC and TC with the classification accuracy. However, the accuracy was either reduced as a result of cost minimization or not measured by the researchers.

The next chapter will discuss in details the research methodology, it will present the datasets and the algorithm that will be used in the experiments, and will discuss some necessary concepts to conduct the research.

Chapter Three

Research Methodology

3.0 Overview

This chapter will first introduce the research methodology that will be followed through this research. In section 2, the datasets that will be used in the experments during the research with their main properties will be discussed. In section 3, the Weka version of algorithm C4.5 (J48) including the basic algorithm steps will be discussed. Section 4 and 5 will explain the way of calculating the ATC and the classification accuracy of the J48 classifier. In section 6, some of the operation research approches to optimize the classifier's ATC and accuracy will be discussed in details. Finally, section 7 will discuss the some of the post-pruning and pre-pruning methods used in a DT classifier. It will also define some of the J48 pruning parameters used in Weka.

3.1 Research Methodology

The research methodology that will be followed in this research is a quantitative methodology as the research seeks precise measurments and uses statistical and mathematical data analysis. Following is a summery of the various stages involved in this research:

1. **Literature Review**: In this stage, the problem of cost sensitive learning will be studied, various costs involved in DM process in general and in building a classification model in particular will also be defined. The relationship between the total classification cost and the accuracy will be studied. C4.5 algorithm will also be

studied and the method of calculating the average total TC and MC of a classifier to get the average total cost (ATC) of classification will be defined. In addition, various researchers' attempts to make accuracy/cost trade-off will be presented, including their success rates. Chapter 2 presented the literature review.

2. **Dataset Selection:** The data will be collected from UCI (UCI Machine Learning Repository, 2018) for benchmarking purposes, as UCI contains different types of benchmark datasets on different fields such as business, engineering, social sciences, life sciences, and many others. Many researchers rely on this repository for obtaining datasets to verify their research results. Besides benchmarking, most of the datasets in this repository are ready to be mined, i.e. they are processed, and hence researchers can focus on their own research objectives rather than wasting time on data collection and processing.
3. **Dataset Cleaning:** the selected datasets will be cleaned. Although the selected datasets are preprocessed, some datasets have few missing values. All records with missing values will be deleted.
4. **Conducting Experiments:**
    i. J48 algorithm will be used to build DTs from the training dataset using Weka.
    ii. The average total MC, average total TC, ATC, and the classification accuracy will be calculated and recorded after setting one of the J48 classifier's pruning parameters with range of values.
    iii. The previous step will be repeated with other pruning parameters.
    iv. The experiments will be repeated using other datasets.

5. **Modeling the Accuracy/cost Relationship of the DT classifier:**
    i. A model will be constructed to illustrate the ATC against the classification accuracy at different stages of the tree construction.
    ii. The constructed model will be used to determine the optimum balance between the ATC and the accuracy using the Pareto dominance approach.
6. **Data Analysis and Comparison:**
    i. The experimental results will be presented, analyzed, and discussed.
    ii. A correlation analysis will be conducted to verify the relationship between the ATC and the accuracy and to determine the significance of the relationship.
    iii. A correlation and regression analysis will be conducted to determine the impact of the dataset properties on the classifier's accuracy/ATC.
    iv. Results obtained from conducting series of experiments on different datasets will be compared with other researchers' results to verify their effectiveness.

The following sections will discuss various stages in more details. Figure 3.1 visualizes the methodology followed in this research.

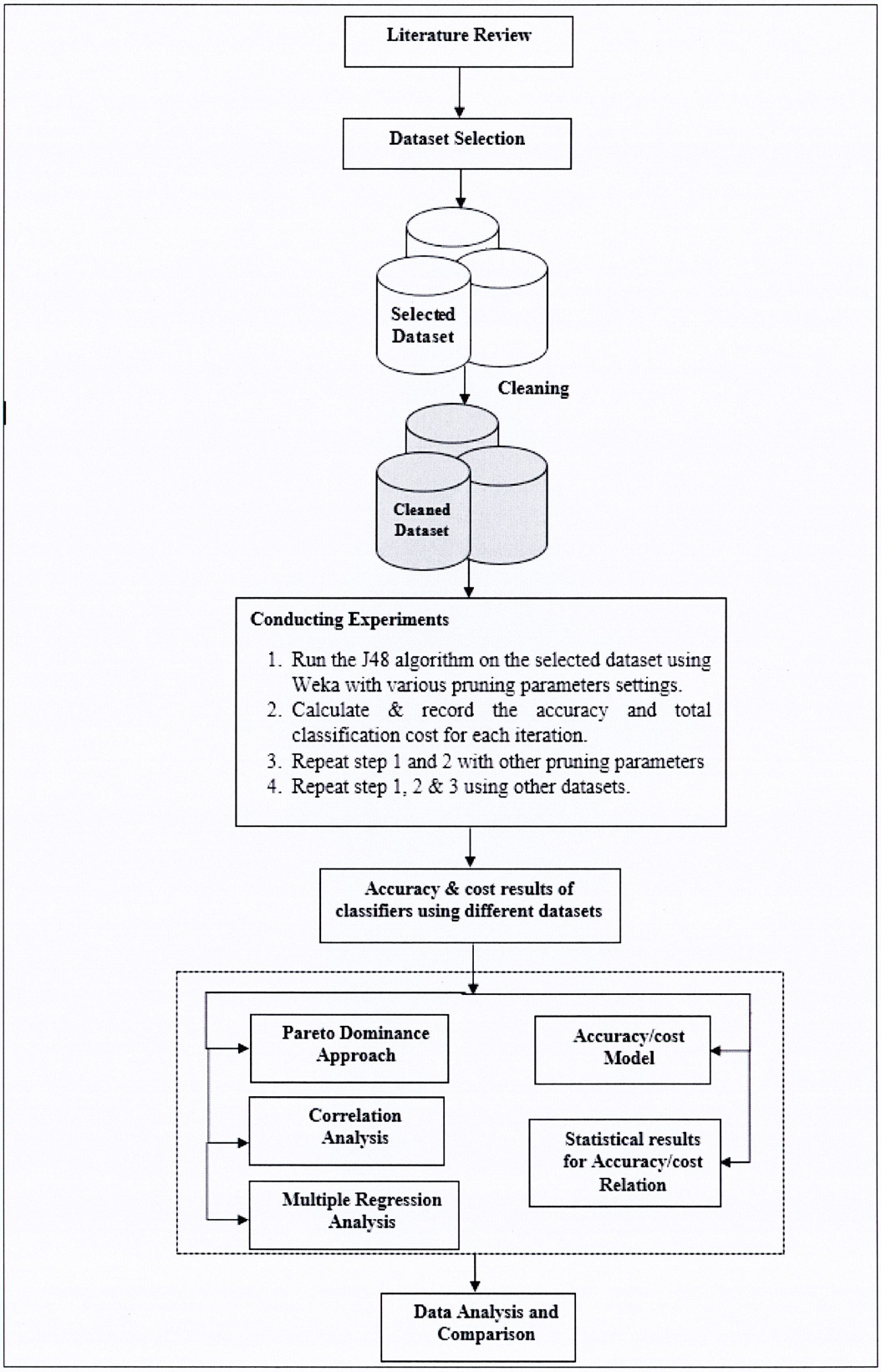


Figure 3.1: Research Methodology

3.2 Datasets Used in the Experiments

This section briefly describes the datasets that will be used for experimentation throughout this research. All datasets are from UCI ML repository (UCI Machine Learning Repository, 2018). The selected datasets are heart disease (HD), thyroid disease, and breast cancer datasets. They vary in the number of attributes, the number of cases, and the number of classes they have. Some datasets include balanced class distribution, while others are imbalanced.

3.2.1 Heart Disease Dataset

This dataset originally contains 76 numeric attributes but all of the researchers worked on this dataset used just 14 attributes in their experiments. The dataset contains information on the patient's age, sex, fasting blood sugar, heart rate, and other medical attributes. It has one attribute indicating whether a patient is having HD or not. In this dataset, there exist 303 records. All records with missing attribute's values have been deleted, hence, the remaining records were 297. Various researchers used this dataset including Pazzani, et al. (1994), Turney (1995), Ling et al. (2006), and Chen et al. (2016). The dataset's attributes are described in Table 3.1(UCI Machine Learning Repository, 2018).

Table 3.1: Attributes in HD Dataset

Attribute	Description
Age	Age of the patient in years.
Sex	Sex of the patient. - Value 1: male - Value 0: female
Cp	Chest pain type. - Value 1: typical angina - Value 2: atypical angina - Value 3: non-anginal pain - Value 4: asymptomatic
Trestbps	Resting blood pressure (in mm Hg on admission to the hospital).
Chol	Serum cholestoral in mg/dl.
Fbs	Fasting blood sugar.
Restecg	Resting electrocardiographic results. - Value 0: normal - Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) - Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
Thalach	Maximum heart rate achieved.
Exang	Exercise induced angina. - Value 1: yes - Value 0 : no
Oldpeak	ST depression induced by exercise relative to rest.
Slope	The slope of the peak exercise ST segment. - Value 1: upsloping - Value 2: flat - Value 3: downsloping
Ca	Number of major vessels (0-3) colored by flourosopy.
Thal	Finishing heart rate. - Value 3: normal - Value 6: fixed defect - Value 7: reversable defect

The class attribute in this dataset indicates the existence of heart disease (values 1 to 4) or the absence of it (value 0).

3.2.2 Thyroid Disease Dataset

The thyroid disease dataset is a big dataset that has 21 numeric attributes. All attributes are binary except six continous attributes (age, TSH, T3, TT4, T4U, and the class attribute). It has 3772 medical records with no missing attribute's values (UCI Machine Learning Repository, 2018). It classifies thyroid as normally functioning, under functioning (hypothyroid), or over functioning (hyperthyroid). Many researchers used this dataset including Turney (1995), Ling et al. (2006) and Weiss et al. (2013). Table 3.2 describes each attribute in the dataset (Turney , 1995).

Table 3.2: Attributes in Thyroid Disease Dataset

Attribute	Description
Age	Age in years
Sex	Gender
On Thyroxine	Patient on thyroxine
Query Thyroxine	Maybe on thyroxine
On Antithyroid	On antithyroid medication
Sick	Patient reports malaise
Pregnant	Patient pregnant
Thyroid Surgery	History of thyroid surgery
I131 Treatment	Patient on I131 treatment
Query Hypothyroid	Maybe hypothyroid
Query Hyperthyroid	Maybe hyperthyroid
Lithium	Patient on lithium
Goiter	Patient has goitre
Tumour	Patient has tumour
Hypopituitary	Patient hypopituitary
Psych	Psychological symptoms
TSH	TSH value, if measured
T3	T3 value, if measured
TT4	TT4 value, if measured
T4U	T4U value, if measured

The class attribute in this data set have 3 possible values: value 1 indicates hypothyroid, value 1 indicates hyperthyroid, and value 3 indicates that the thyroid is functioning normaly.

3.2.3 Wisconsin Breast Cancer Dataset

This dataset is one of the breast cancer datasets at UCI. It was obtained from the University of Wisconsin by W.H. Wolberg. It has 10 numeric attributes and a total of 699 records. Researchers used this dataset to build models that predict whether a sample taken from a patient is benign or malignant. The attributes found in this dataset are listed in Table 3.3 (UCI Machine Learning Repository, 2018). The class attribute in this dataset can have 2 values: value 2 indicates benign and value 4 indicates malignant.

Table 3.3: Attributes in Breast Cancer Dataset

Attribute	Domain
Clump Thickness	1 - 10
Uniformity of Cell Size	1 - 10
Uniformity of Cell Shape	1 - 10
Marginal Adhesion	1 – 10
Single Epithelial Cell Size	1 – 10
Bare Nuclei	1 – 10
Bland Chromatin	1 – 10
Normal Nucleoli	1 – 10
Mitoses	1 - 10

Sixteen records have been removed from the dataset because of having a single missing attribute value, researchers followed this method include Dogan and Tanrikulu (2012) and Lim and Loh (2000). Researchers used this dataset include Sheng and Ling (2006), Zhao et al. (2015) and Qiu et al. (2017).

All medical attributes in all datasets will be renamed as A, B, C, etc. through the expermentation for simplicity as many researchers did including Ling, et al. (2004).

3.2.4 Datasets Properties

Datasets vary in their properties/characteristics. Different dataset properties have different effects on the DM process. This section briefly discusses the main properties of a dataset including the size of a dataset, type, dimensionality, sparsity, and the abstraction level as this research will be studying the impact of the dataset characteristics on the constructed classifier's accuracy and ATC.

The **dataset size** can be measured in terms of the number of instances (sometimes called cardinality) or the total number of bytes. This property may affect the speed of the DM process and the memory space required. The **type of a dataset** property refers to the type of data file that is used to repersent the dataset, i.e. the file extension. Weka accepts Tables in ARFF or CSV format (Du H. , 2010).

On the other hand, **dimensionality** property refers to the number of attributes in a dataset (sometimes called degree). High dimentional dataset can make the DM process quite difficult and resource consuming in terms of the computation speed and memory consumed. Attributes may also differ in their types. The two common types of attributes are numeric (i.e. real numbers) and nominal (i.e. categorical) (Witten & Frank, 2005).

Sparsity property reflects the degree to which data values in a dataset are distributed. Data may not distribute uniformaly, instead data points might be very close to each other. Therefore, it would be very difficult to find out patterns in a highly sparsity dataset (Zuber & Faltings, 2006).

The **abstraction** property of a dataset means the level at which data is presented. Data can be presented as recorded (i.e. raw data), it can be summerized or approximated. For example, the student height can be

measured in centimeters and then summerized as short, average, or tall. This property can affect the patterns to be discovered (Du H. , 2010). Table 3.4 summerizes the properties of the selected datasets.

Table 3.4: Dataset Properties

Dataset	Number of instances	Number of classes	Number of attributes (including the class attribute)	Class Distribution (Positive/ Negative)
HD	297	2	14	160/137
Thyroid disease	3772	3	21	3488/191/93
Wisconsin breast cancer	683	2	10	444/239

3.3 DT Classifiers: J48

The J48 algorithm will be used to construct DTs. It is the Java implementation of the famous C4.5 algorithm. It is an open source classifier provided by the DM tool "Weka" (Freitas et al., 2007). It provides some features such as handling missing values in the training dataset, tree prunning after the tree is completely constructed, handling continous attribute values, and using the gain ratio for attribute selection rather than information gain used in ID3 to overcome the problem of selecting attributes with more instances in the training set (Kaur & Chhabra, 2014).

Following are the basic algorithm steps for inducing a DT from a training set (Kamber, 2007):

1. If all instances in the dataset belong to the same class, then the tree will have a single leaf node labled with that class.
2. If not, the gain information is calculated for each attribute.
3. Next, the algorithm calls the attribute selection method, gain ratio in this case, to select the best splitting attribute at each node (the one with the highest gain ratio).

4. The process continues to form a tree. It stops if one of the following events occur:
    - All instances in the dataset belong to the same class.
    - No more attributes remain for partitioning instances. In this case, the majority voting is used to label the leaf node with the most common class.
    - No more instances remain.

As mentioned above, the attribute selection method in case of J48 classifier is the gain ratio. The gain ratio of an attribute (A) can be calculated as shown in equation 3.1.

$$\textbf{\textit{Gain Ratio}} = \frac{\textbf{\textit{Gain}}(A)}{\textbf{\textit{SplitInfo}}(A)} \qquad 3.1$$

where *Gain (A)* is the amount of reduction in the information required for classification by knowing the value of A, the higher is the better. Where as the *SplitInfo(A)* represents the information gained by partitioning the taining dataset into x partiotions representing the x outcomes of a test on attribute A (Patel & Upadhyay, 2012).

The gain ratio is an extension to the information gain method used in ID3 algorithm, which is biased to selecting the attributes with the larger number of values (Kamber, 2007).

3.4 Classification Cost Calculation

This section discusses the way of calculating the ATC. It first introduces the MC matrix and the TC vectors, which provide the basis for calculating the ATC of building a classifier. Next, it explains how to calculate the ATC of a classifier.

3.4.1 Misclassification and Test Cost Settings

This section identifies the MC matrix and the TC vector that will be used in the experiments.

The costs of the HD dataset's attributes that will be used in experiments are shown in Table 3.5 (UCI Machine Learning Repository, 2018). The cost of tests is in Canadian dollar as indicated in UCI ML reposetory (UCI Machine Learning Repository, 2018) and used by many reserchers including Turney (1995), Sheng et al. (2005), Chen et al. (2016), and others.

Table 3.5: HD Dataset Attribute's TC

Attribute	TC ($)
age	1.00
sex	1.00
cp	1.00
trestbps	1.00
chol	7.27
fbs	5.20
restecg	15.50
thalach	102.90
exang	87.30
oldpeak	87.30
slope	87.30
ca	100.90
thal	102.90

The TCs for the thyroid disease dataset was also collected from UCI ML repository. As shown in Table 3.6 (UCI Machine Learning Repository, 2018), the cost of the first sixteen tests is nominal ($1). Originally, the TCs for HD and thyroid datasets were obtained from Ontario Health Insurance Program's fee schedule (Ling et al., 2006).

Table 3.6: Thyroid Disease Dataset Attribute's Cost

Attribute	TC ($)
Age	1.00
Sex	1.00
On Thyroxine	1.00
Query Thyroxine	1.00
On Antithyroid	1.00
Sick	1.00
Pregnant	1.00
Thyroid Surgery	1.00
I131 Treatment	1.00
Query Hypothyroid	1.00
Query Hyperthyroid	1.00
Lithium	1.00
Goiter	1.00
Tumour	1.00
Hypopituitary	1.00
Psych	1.00
TSH	22.78
T3	11.41
TT4	14.51
T4U	11.41

Unfortunatly, the attributes TC for the breast cancer dataset were not provided by UCI, like most of the other datasets. And as finding the real TC values is not an easy task and time consuming, the uniform distribution was applied using SPSS (Statistical Package for the Social Sciences) to generate random TCs between 0 and 100. Many researchers followed this strategy including Yang et al. (2006), Wang et al. (2010), Min, et al. (2011), Zhao et al. (2015), and Qiu et al. (2017). It is logical to follow this strategy since the same attributes cost settings will be used for the experimental evaluations in all of the conducted experiments. Table 3.7 shows the attribut's costs for the breast cancer dataset.

Table 3.7: Breast Cancer Dataset's Attributes Cost

Attribute	TC($)
Clump thickness	13.96
Uniformity of cell size	43.13
Uniformity of cell shape	61.22
Marginal adhesion	29.08
Single epithelial cell size	15.57
Bare nuclei	69.95
Bland chromatin	34.63
Normal nucleoli	44.56
Mitoses	5.24

The MC matrix that will be used for HD and brest cancer datasets will be adopted from (Sheng & Ling, 2006), (Ling et al., 2006), and (Weiss et al., 2013) as the UCI ML repository doesn't supply the MCs. Table 3.8 shows the MC matrix that will be used for HD and breast cancer datasets in this research. It shows different costs for different types of errors depending on the predicted class and the actual class of an instance. It also shows that it is more costly to predict that someone as healthy (0) by miskate than prdicting him/her incorrectly as patient (1). Obviously, correct classification has no cost (the matrix diagonal). Note that the MCs shown in Table 3.8 was recommended by researchers from the Heart-Failure Research Group in the local medical school (Weiss et al., 2013).

Table 3.8: MC Matrix of the HD and Breast Cancer Datasets

	Predict 0 (healthy)	**Predict 1 (patient)**
Actual 0 (Healthy)	0	FP $ 600
Actual 1 (Patient)	FN $ 1000	0

Since the thyroid dataset has three types of classes (hypothyroid, hyperthyroid, or normal), Turney (1995) recommended the MC matrix shown in Table 3.9. Turney assigned the minimum of FN and FP costs in cases where someone diagnosed as hyperthyroid while the correct

dignosis is hypothyroid (or the opposite) as this situation has lower consequences than diagnosing him/her as normal (healthy). Note that FP and FN costs are $600 and $1000, respectively as in HD and breast cancer datasets.

Table 3.9: MC Matrix of the Thyroid Disease Dataset

	Predicted 3 (Normal)	**Predicted 2 (Hyperthyroid)**	**Predicted 1 (Hypothyroid)**
Actual 3 (Normal)	0	FP cost $ 600	FP cost $ 600
Actual 2 (Hyperthyroid)	FN cost $ 1000	0	Minimum (FN cost, FP cost) $ 600
Actual 1 (Hypothyroid)	FN cost $ 1000	Minimum (FN cost, FP cost) $600	0

Both the TC and the MC are assumed to have the same cost scale ($), so that adding them together to get the total classification cost is feasible.

3.4.2 Average Total Classification Cost Calculation

The total cost of a classifier is the sum of the average total MC and the average total TCs of the testing dataset as mentioned in section 2.6.3. Weka uses the MC and the confusion matricies to get the MC of a classifier using the following formula:

$$\textbf{\textit{Total misclassification cost}} = \boldsymbol{FN \times C_{FN} + FP \times C_{FP}} \qquad 3.2$$

where FN is the number of false negative cases, and C_{FN} is the cost of false negative cases in the testing dataset. While FP is the number of false positive cases, and C_{FP} is the cost of false positive cases (Jiang, Li, & Wang, 2014).

For the TC, Weka does not provide a systematic way for calculating it as the TC was studied much less than the MC (Zhou et al., 2016). However, as many researchers indicated, it is the total cost of all tests in the path through which a test case goes from the root node to the leaf node (Turney, 1995), (Sheng et al., 2005), (Davis et al., 2006). If a test apears more than once through the path, then it will be charged just once (Turney, 1995). Tests which are done together or in a particular order can have a discounted cost (Weiss et al., 2013), but this is out of the scope of this research.

The ATC can be calculated by adding the total MC and the total TC for all testing cases, divided by the number of cases in the testing dataset.

3.5 Accuracy Calculation

As indicated in section 2.4.1.1, the accuracy of a classifier depends on the number of correctly classified instances in the testing dataset and is calculated using the confusion matrix as shown in eqaution 3.3.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \quad 3.3$$

3.6 Accuracy vs Cost Optimization

This section will discuss some of the operation research (OR) approaches that can be exploited to optimize two or more DM objectives (criterias) simultaneously or what is also called multi-objective optimization approaches. One approach to do so is to convert the multi-objectives into a single one using a weighted formula. Another approach is to prioterize each objective, this approch is called "lexicographic approach". Pareto approach involves returning a set of non-dominated solutions to the user

(Freitas, 2004). The follwing subsections discuss the three approches in more details in order to make a trade-off between the accuracy and the ATC of a DT classifier.

3.6.1 The Conventional Weighted Formula Approach

This approach is the most widely used approach due to its simplicity and easiness to use. It converts the multi-objectives into a single one by giving a numerical weights to each objective and then add the values of the weighted criteria together to get a single value. The weights are set by the user and reflect the importance of the objectives (Maimon & Rokach, 2005). An example of the score(s) of a model with n objectives is:

$$Score = w_1 \times O_1 + w_2 \times O_2 + ... + w_n \times O_n \qquad 3.4$$

where w_n is the weight allocated to objective O_n and n is the number of objectives (Marler & Arora, 2004).

This approche have some disadvantages including the subjectiveness of objectives weights', as it is the user option to select the appropriate weight for each objective (Freitas, 2004). This approach also mixes different types of objectives in one formula (e.g. minimizes the number of selected attributes and maximizes the classifier's accuracy), which might not be meaningful to the user. In addition, as this approach returns a single solution, the user misses the chance to explore the varying trade-offs (Maimon & Rokach, 2005).

3.6.2 Pareto Approach

In this approach the multi-objectives problem will not be transfered into a single one, instead, each objective is tested individually. In Pareto

dominance approach, solution *S1* dominates another solution, let's say *S2*, if and only if it does better than *S2* in at least one of the optimised objectives and *S1* does no worse than *S2* in all other objetives being optimised. In other words, solution *S1* dominates solution *S2* iff:

$\forall j = 1, 2, \ldots, n, f_j(S1) \leq f_j(S2)$, and there exists $z \in \{1, 2, \ldots, n\}$ such that $f_z(S1) < f_z(S2)$. Note that the all objectives in the aforementioned example to be minimized.

In addition Solution *S1* is called Pareto-optimal solution if it is not dominated by any other solution (Jin & Sendhoff, 2008).

The difference between the conventional approach and this approach is that the former approach gives a single solution while this approach gives a set of solutions, which makes it a more complex approach as the user have to select the best non-dominated solution from the set of all non-dominated solutions based on his/her preferences (Marler & Arora, 2004).

Figure 3.2 clarifies the concept of Pareto dominance where the two objectives to be optimized are to maximize the model accuracy and simplicity. In this figure, model A is dominated by model B and D. Model C is dominated by model D, and finally model E is dominated by model F. Thus, models B, D and F are said to be non-dominated solutions, also called "Pareto front" (Freitas, 2004).

Researchers recommended using this approach to solve multi-objective optimization problems in DM include Freitas (2004), Jin and Sendhoff (2008), and Kamila et. al (2016).

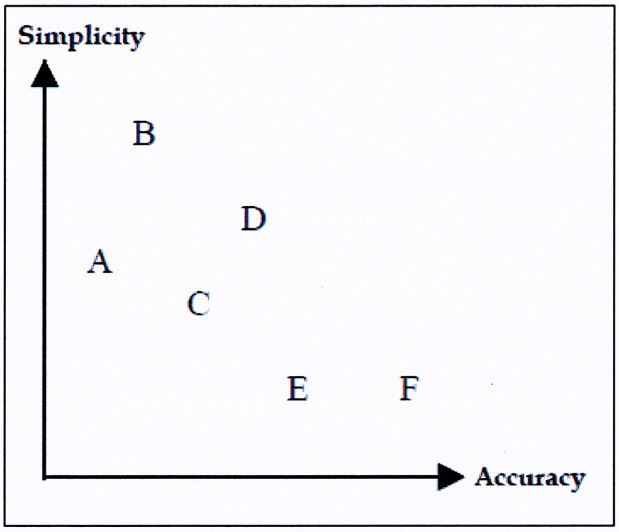


Figure 3.2: An Example of Pareto Dominance

3.6.3 Lexicographic Approach

This approach orders objectives based on their importance and then concentrate on optimizing objectives with high priority. In particular, it just tells that one objective is more important than another instead of assigning weights to objectives (Marler & Arora, 2004). For example, when comparing two DM models to select the best one, first they are compared based on their performance measure of the highest priority objective and the one that performs better is selected. Otherwise, they are compared based on their performance measure of the next high priority objective and so on. Note that to consider one model is performing better than the other in a particular objective, the difference in their performance has to exceed a particular threshold value. Therefore, this approach requires assigning a threshold which is not an easy task. The standard deviation-based thresholds can be used for this purpose, which in turn require the specification of one more parameter, i.e. the degree of confidence. Like the conventional approach, this approach also returns a single solution to the user (Freitas, 2004).

3.7 Decision Tree Pruning

Patel & Upadhyay (2012) defined pruning as the process of reducing the size of a tree that is too large or too deep. It is required to avoid the overfitting (specific adjustment) problem and large DTs. It also participates in generating a more comprehesive learning models (Neto, et al., 2017).

Two types of DT pruning methods exist: post-pruning and pre-pruning. In post-pruning, the DT is constructed first, then the reduction of non important tree branches and levels take place. On the other hand, in pre-pruning the DT keeps tracking for overfitting during the tree construction process (Witten & Frank, 2005). The following subsections will list some of the post-pruning and pre-pruning methods. It will also discuss some of the J48 pruning parameters in Weka.

3.7.1 Post-Pruning Methods

There are various methods for post-pruning a DT. This section briefly introduce some of those methods namely: error-based pruning, reduced error pruning, minimum error pruning, and cost based pruning methods. Other post-pruning methods include: minimum description length pruning, pessimistic pruning, among others.

3.7.1.1 Error-based Pruning

This pruning method is used in C4.5 algorithm. It uses the upper bound of the statistical confidence interval to estimate the expected error rate as follows (Rokach & Maimon, 2008):

$$\varepsilon_{UB}(T,S) = \varepsilon(T,S) + Z_{\alpha} \times \sqrt{\frac{\varepsilon(T,S)\times(1-(\varepsilon(T,S))}{|S|}} \qquad 3.5$$

where $\varepsilon(T,S)$ represents the misclassification error rate of a tree T on the training set S, Z is the inverse of the standard normal distribution, and α is the required significance level.

The followings are estimated at each non-leaf node (t) in a bottom-up order:

1. The misclassification error rate of a subtree rooted at node t.
2. The misclassification error rate after pruning the subtree rooted at node t.
3. The resulting misclassification error rate, when node t is replaced by a subtree rooted at the most frequent node of t, i.e. the node that most of the instances in the training set traverse through.

The errors in the previous three steps are then compared. If the result of the first step is the lowest, the tree kept as it is. If the second step gave the lowest error rate, then the tree is pruned at node t. Otherwise, node t is replaced by the subtree rooted at the most frequent node of t (Rokach & Maimon, 2005).

3.7.1.2 Reduced Error Pruning

This is the simplest method for DT pruning that was developed by Quinlan. In this method, each DT node is nominated for pruning by removing the subtree rooted at that node and converting it to a leaf node while passing over the tree from the bottom to top (Rokach & Maimon, 2008). Pruning is done if the error rate of the resulting tree (after pruning)

is lower than or equal to that of the original tree (before pruning). In this method the dataset is divided into three parts: the training set, the testing set, and the validation set (pruning set) used for error estimation and making pruning decision (Rokach & Maimon, 2005). As indicated by Patel & Upadhyay (2012), this method works well as good as the other pruning methods in terms of the accuracy and also better than others in terms of the resuting tree size.

The method steps are summerized in Figure 3.3.

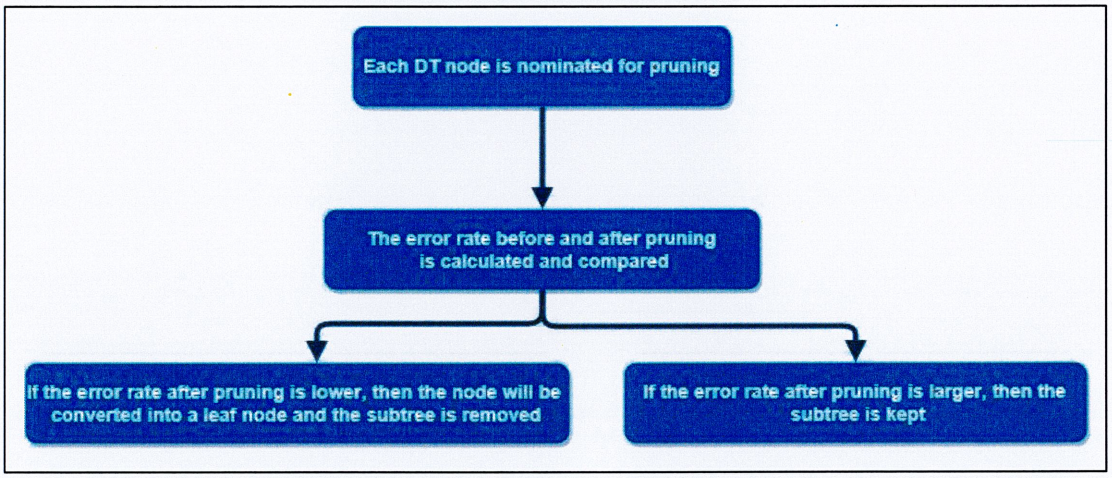


Figure 3.3: Reduced Error Pruning Method

3.7.1.3 Minimum Error Pruning

This method was proposed by Niblett and Brotko in 1986 (Cestnik & Bratko, 1991). It aims at minimizing the expected error rate of pruning on an independent dataset. It works as follows:

First, it calculates the expected error rate of pruning at each non-leaf using equation 3.6:

$$E(t) = \frac{n_t - n_{c,t} + k - 1}{n_t + k} \qquad 3.6$$

where k is the number of classes, n_t is the number of cases in node t, $n_{c,t}$ is the number of cases labeled with class c in node t.

Then, it calculates the expected error rate at that node if the subtree was not pruned.

Finally, it compares both rates. If the expected error rate was greater after pruning the subtree rooted at the non-leaf node, then the subtree is kept as it is. Otherwise, it is pruned and the node becomes a leaf node (Patel & Upadhyay, 2012).

3.7.1.4 Cost Based Pruning

In this method, the error rate as well as the MC is considered at each non-leaf node to make pruning decisions. The classification cost matrix and the propability distribution of a given class is used to calculate the expected cost at each node by multiplying the two. For example, if the cost of misclassifing sick patient as healthy is ten times the cost of misclassifing a healthy patient as sick according to the classification cost matrix, then each of a child nodes rooted at that node might predict sick to minimize the expected MC and pruning will take place. In this method pruning a DT depends heavily on whether pruning minimizes the MC or not (Bradford et al., 1998). Figure 3.4 summerizes this method.

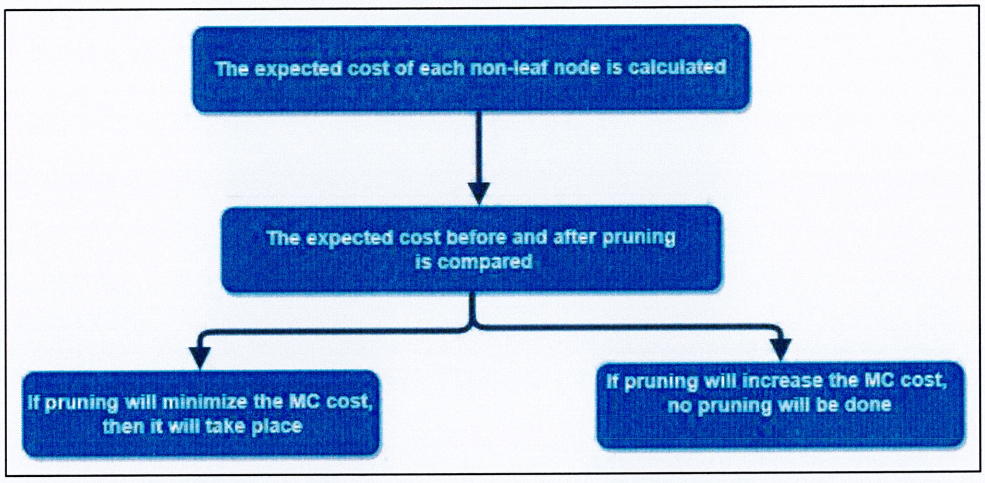


Figure 3.4: Cost Based Pruning Method

3.7.2 Pre-pruning Methods

Pre-pruning methods are also called "forward pruning" or "online pruning". It involves adding a condition to prevent unnecessary branches during the DT construction. The condition assesses the benefits of splitting instances at a particular node. If the goodness value exceeds a predefined threshold, the split will take place. Otherwise, it will be prevented (Patel & Upadhyay, 2012).

The two major pre-pruning methodes are the minimum number of objects pruning and the Chi-square pruning, will be discussed in the following subsections.

3.7.2.1 Minimum No. of Object Pruning

In this method, the minimum number of instances required at leaf nodes is specified, so that all leaf nodes in a DT will have greater than or equal to the specified number of instances. In addition, all splits that lead to having leaf nodes with lower number of instances than the pre-defind threshold will be rejected. This will prevent the overfitting problem

(Rajput & Arora, 2013). It also results in smaller and simpler DT (Stiglic et al., 2012). In Weka, the minimum number of instances threshold is specified using the MinNumObj parameter (Neto, et al., 2017).

3.7.2.2 Chi-square Pruning

In this pruning method, a statistical test is applied on data to determine whether splitting the instances on a particular feature is statistically significant or not (Rajput & Arora, 2013). It depends on the impact of the split on the class distribution of the partitions resulting from the split. If the class distribution after splitting is consistent with the class distribution at the current node, pruning will take place (Patel & Upadhyay, 2012).

3.7.3 Pruning Parameters in Weka

In Weka, there are some pruning parameters such as MinNumObj, numFold and confedince factor.

The **MinNumObj** specifies the minimum number of instances at each leaf node in a DT. If the number of instances at a child leaf node is less than the MinNumObj, then the child node and the parent node are combined in one node (Patel & Upadhyay, 2012). Increasing the MinNumObj not only reduces the tree size but also decreases the accuracy of the DT (Neto, et al., 2017).

The **NumFold** parameter can be used only if the reduced error pruning parameter is set to true. The default pruning method in J48 is the error-based pruning method, unless the reduced error pruning parameter is set to true. This parameter specifies the data amount used for reduced error pruning. Regardless the number of folds specified, only one fold is used for reduced error pruning. For example, if the NumFold is set to 4, only

one fold will be used for pruning and the other 3 folds will be used for constructing the DT (Patel & Upadhyay, 2012).

The **Confidence Factor** (CF) parameter in J48 classifier in Weka can also contributes to the reduction of the tree size. Stiglic et al. (2012) defined the CF as "a threshold of allowed inherent error in data while pruning the decision tree". It determines the confidence level for the confidence intervals used in error-based pruning method for error estimation. Lowering the CF means having less confidience in the training data and more pessimistic error estimations, which leads to heavier pruning (Stiglic et al., 2012), (Rajput & Arora, 2013).

3.8 Conclusion

This chapter presented the research methodology that will be followed in this research. It was a quantitative methodology as it will be focusing on precise objective measurments, and uses statistical and mathematical ways of data analysis. The various stages involved in the methodology was described, including: litrature review, data selection, data cleaning, conducting experiments, building accuracy/ATC models, and data analysis and comparision.

The J48 algorithm, which is the Java implementation of C4.5 algorithm in Weka, was introduced. One of the main features this classifier provide is the DT pruning during and after the tree construction. This chapter also mentioned the attribute selection method used in J48 which is gain ratio and how it is calculated.

The selected datasets for experiments was fully discribed including HD, thyroid disease, and breast cancer datasets. The selected datasets differ in their properties including whether they are balanced or imbalanced

datasets, the number of instances, the number of attributes, and the number of classes they have.

The way of calculating the accuracy and the ATC in J48 classifier were also defined. The TC vector and the MC matrix were introduced, which provide the basis for calculating the total TC and the total MC of a DT classifier.

In addition, some approaches for accuracy-cost optimization were discussed. The Pareto dominance approach will be used to address the accuracy-ATC trade-off in a DT classifier as it does not require the user to assign weights or ranks to objectives compared to lexicographic and weighted sum approaches and hence is not subjective. In addition, many researchers recommended using Pareto dominance approach to solve multi-objectives problems in DM and discussed the shortcomings of using the other two approaches, especially the weighted sum approach. Finally, some post-pruning and pre-pruning methodes to address the overfitting problem and minimize redandecies in a DT were also defined. The pruning parameters of the J48 algorithm were discussed: the MinNumObj, NumFold, and CF parameters. All three parameters will be used to conduct experiments.

Next chapter will discuss in details the conducted experiments, the results obtained, and a comparision and analysis of the results.

Chapter Four

Results and Analysis

4.0 Overview

This chapter starts by discussing the experimental framework followed to conduct the experimental work. The phases of experiments applied here are sound and were followed by many researchers. In section 2, the set of experiments performed on the selected datasets will be described in details. Section 3 will illustrate and compare the experimental results obtained from tuning the pruning parameters including the confidence factor (CF), minimum number of instances at leaf nodes (MinNumObj), and the number of folds (NumFold) used in reduced error pruning method. A correlation analysis will take place in section 4 to validate the experimental results.

In section 5, the impact of the dataset's characteristics on the classification accuracy and the ATC will be studied by conducting correlation and multiple regression analysis. In addition, the accuracy/cost trade-off will be resolved in section 6 using Pareto dominance approache.

In section 7, the results of this research will be compared to ther researchers results. Finally, section 8 will provide a summery to this chapter.

4.1 Experimental Framework

This section introduces the experimental framework maintained throughout the research. It starts with the dataset selection, then the

decision tree (DT) construction takes place. Three pruning parameters will be used to reduce the tree size with various settings. For each setting , the average total cost (ATC) of classification and accuracy of the resulting tree will be calcuated to model the relationship between the two for each selected dataset. The next stage is to use the Pareto dominance approach to determine the Pareto-optimal solutions that resolve the trade-off between the classification accuracy/ATC pairs. The final stage includes presenting, discussing, and analyzing the results. Following are more details about each stage of the experimental framework:

1. **Dataset selection**: in the implementation phase three medical datasets will be used as one of the objectives of this research is to study the impact of the dataset characteristics on the classifier's accuracy and ATC. The selected datasets will be real datasets collected from UCI as most of the researchers in cost-sensitive learning did, including Sheng et al. (2005), Dogan & Tanrikulu (2012), Zhou et al. (2016), and Chen et al. (2016).

   A processed data will be selected as data preprocessing is out of this research scope, but as some datasets have few missing values, all records with missing values will be deleted. The selected datasets will have all numeric attributes except for the class attribute (nominal). They differ from each other in terms of their size, dimentionality, number of classes, and the class distribution. The sparsity property of datasets will not be considered as it is more applicable to consider in clustering studies. No summerized attributes exist in all datasets.
2. **DT construction**: Weka software will be used to run J48 algorithm (the Java implementation of C4.5 algorithm) on the selected datasets. Weka is the prefered DM softwere tool for teaching and

learning (Du, 2010), (Dogan & Tanrikulu, 2012) and research. All algorithm settings will be setteled in this stage including the MC matrix and the number of folds in cross-validation.

3. **DT pruning**: three of the J48 algorithm's pre-pruning and post-pruning paramters in Weka will be used with various settings, the CF (post pruning), MinNumObj (pre pruning), and NumFold (post pruning). To the best of our knowledge, none of the researchers went for this option for optimizing the accuray/ATC pairs in a DT. All researchers found resolved the trade-off by updating the attribute selection criterion of the DT algorithm including Kong et al. (2016), Zhou et al. (2016), and Qiu et al. (2017).
4. **Cost calculation**: the ATC of the constructed DT will be calculated after tuning the classifier's pruning parameters number of times and readings will be recorded. Weka gives the total and avereage MC of a DT but it does not calculate the total TC. Therefore, it will be calculated externally based on the tree path that will be followed within the tree by each testing case in the testing dataset and the TC vector as indicated in section 2.6.3 in chapter 2. The average of all TCs will be calculated, then added to the average total MC to get the ATC.
5. **Accuracy calculation**: accuracy is the main performance measure of a DT classifier in this research, as it is the commonly used criterion to reflect the quality of a model in supervised learning (Jin & Sendhoff, 2008). The accuracy of a DT classifier will be calculated by Weka. It is important to measure the overall accuracy and ATC at different stages when building a DM model to verify its effectivness and help in determining the optimal balance between the two (He et al., 2017).

6. **Modeling the accuracy/ATC relationship**: the obtained readings of accuracy and the ATC will be used to model the relationship between the two. Researchers followed this approach include: Davis et al. (2006), Zhou, et al. (2016), and He et al., (2017). However, Davis et al. and Zhou et al. just modeled the TC and the accuracy while He et al. modeled the computational cost with the accuracy. Three models will be created for each dataset as three pruning parameters will be used.
7. **Use of optimization approach**: the Pareto dominance approach will be used to determine the optimal balance between the ATC and the accuracy. This approach was used by many researchers to solve multi-objective optimization problems in many fields (Du et al., 2007), specially in the DM filed including (Zhang, 2013) and (Kamila, 2016).
8. **Overall results analysis, comarision and presentation**: the three constructed models for each dataset will be studied and the relationship between the ATC and the accuracy will be validated by calculating the Pearson correlation coefficients.

   The models will also be compared together and Pearson correlation and multiple regression analysis will be conducted to determine the impact of the dataset properties on the classifier's accuracy and the ATC. In addition, the percentage of cost savings that gives satisfactory accuracy will be calculated and compared with other researchers results.

Figure 4.1 shows a folw chart summerizing the experimental framework stages.

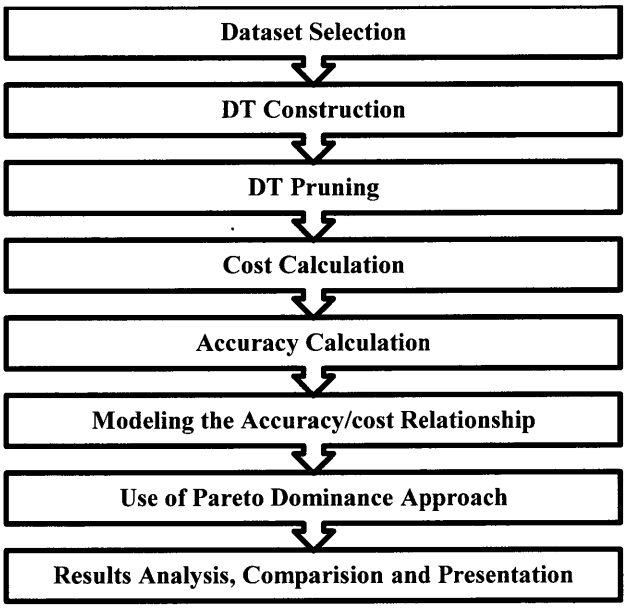


Figure 4.1: Experimental Framework

4.2 Experimental Setup

This section describes the experiments conducted on the selected datasets and the experiments' settings. The main purpose of the experiments is to reveal the behavior of DTs with different pruning parameters settings and use the experimental results to model the relationship between the accuracy and ATC of a DT classifier.

The Weka version used to run the J48 algorithm was Weka 3.8.2, which was released in 2017. In addition, the 10-fold cross validation option for the testing datasets was used in all experiments as it is the most acceptable setting (Bhowmick et al., 2009). The FP and FN costs were set to \$600 and \$1000, respectively and the TC vectors indicated in section 3.4.1 were used.

SPSS version 22.0 was used to generate a random attribute costs for the breast cancer dataset, and for conducting the correlation and multiple regression analysis. All experiments were conducted on a desktop

computer with the following specifications: Windows 10 pro, 64-bit Operating System, Intel Core i7, 3.60 GHz CPU, and 8 GB RAM.

Following is a detailed describtion of the conducted experiments and the expermental settings in each stage:

Stage 1: the CF parameter was set to values ranging from 0.1 to 1 with an increment of 0.1 to minimize the tree size (Witten, Frank, & Hall, 2011). The minimum number of instances (MinNumObj) was held at 2, which is the default setting, and the reduced error pruning option was also set to its default option (false) to prevent pruning the tree using the reduced error pruning method. The algorithm was run after tuning the CF parameter and a batch of DTs was produced.

Stage 2: for the pre-pruning tests, the minimum number of instances (MinNumObj) was tested. The values of MinNumObj used in the experiments was from 5 to 40 with an increment of 5 (Patel & Upadhyay, 2012). The CF parameter was set to 0.25 (default setting) and the reduced error pruning option was kept false (the default option). A new set of DTs was built after running the J48 algorithm.

Stage 3: in this stage, the reduced error pruning method was used. The reduced error pruning parameter was set to true and the number of folds, which indicates the amount of data used for reduced error pruning, was set to 3 (the default setting), 5, 7, 9, 11, 13, and 15 (Patel & Upadhyay, 2012). The CF and MinNumObj parameters were set to their default options. Another batch of DTs was built again.

Stage 4: for each one of the constructed trees, the accuracy, the average total MC, the average total TC, and the ATC were calculated and recorded.

Stage 5: stages 1 to 4 were repeated with the same settings for the three selected datasets.

A total of 25 experiments were conducted on each dataset. Ten different settings for the CF parameter, eight settings for the MinNumObj parameter, and seven settings for the NumFold parameter were used as indicated in stage 1, 2, and 3, respectively. Thus, the total number of experiments conducted for the three selected datasets was 75 experiments (25×3). Table 4.1 summerizes all experimental stages.

Table 4.1: Experimental Stages

Experimental Stages	Pruning Parameter Setting	Stage Description	Other Experimental Settings	Output of Stages
Stage 1	- Set the CF parameter to values ranging from 0.1 to 1 with an increment of 0.1. - Set the MinNumObj to 2 (default setting). - Set the reduced Error Pruning to false (default option).	Run J48 algorithm with particular settings for pruning parameter.	10-fold cross validation option used, FP cost was set to 600, and FN cost was set to 1000. The TC vector indicated in section 3.4.1 was used.	A batch of DTs
Stage 2	- Set the MinNumObj parameter to values from 5 to 40 with an increment of 5. - Set the CF parameter to 0.25 (default setting) - Set the reduced error pruning option to false (default option).			A second batch of DTs
Stage 3	- Set the reduced error pruning option to true. - Set the NumFold parameter to values 5, 7, 9, 11, 13, and 15. - Set the MinNumObj to 2 (default setting). - Set the CF parameter to 0.25 (default setting).			A third batch of DTs
Stage 4	-	Calculate and record the classifier's accuracy, average total MC, average total TC, and ATC for each constructed tree.	-	Accuracy and cost results.
Stage 5	-	Repeat stages 1 to 4 with the other two datasets.	-	Accuracy and cost results.

4.3 Results Discussion

This section discusses in details the results obtained from the conducted experiments on heart disease (HD), breast cancer, and thyroid datasets.

4.3.1 Heart Disease Dataset

This section discusses the experimental results obtained by tuning the CF, MinNumObj, and NumFolds pruning parameters while constructing a DT using the HD dataset.

4.3.1.1 Setting the Confidence Factor Parameter

The CF parameter was the first parameter to test in the HD dataset. The initial tree constructed when the value of the CF parameter was set to 0.1 is shown in Figure 4.2. Note that A, B, C, etc. are the attributes representing some medical tests.

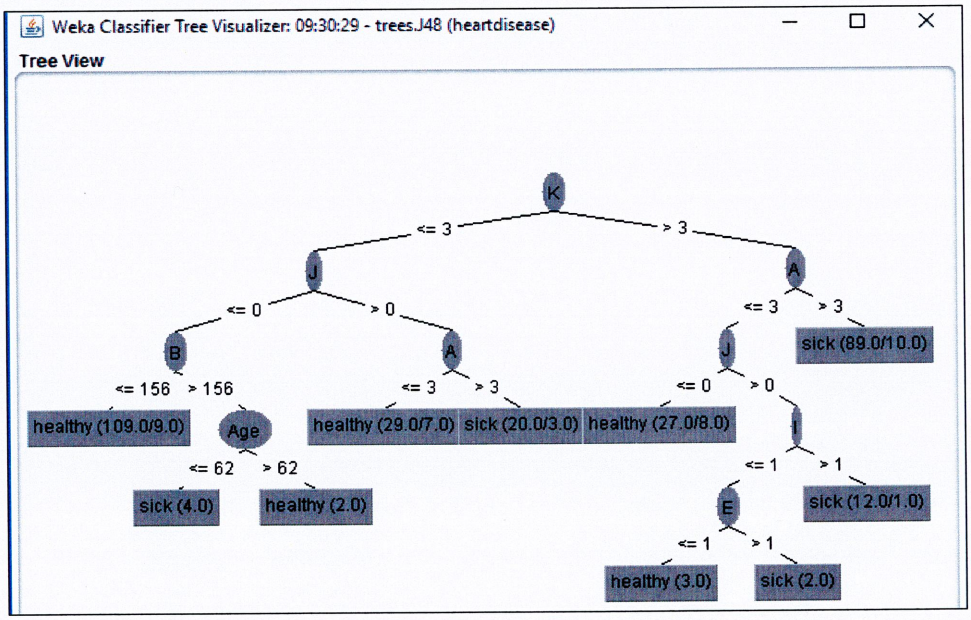


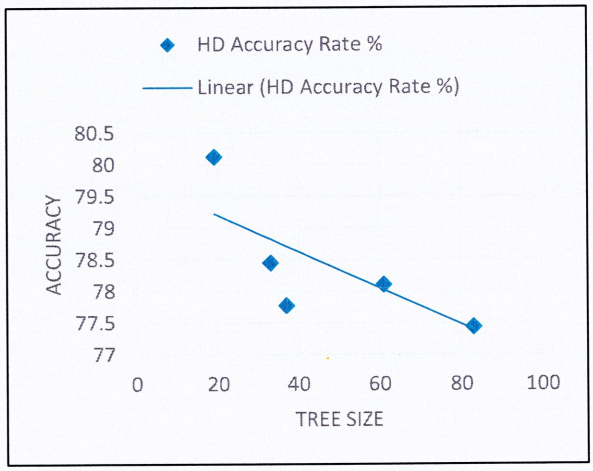
Figure 4.2: The DT Constructed using the CF Pruning in HD Dataset

In the HD dataset, increasing the CF increases the DT size. In addition, when the value of the CF becomes above 0.5, the accuracy of the classifier starts decreasing and the resulting tree remains the same as shown in Figure 4.3(a). This is because setting the CF value more than 0.5 is equivelent to turning pruning off.

The ten runs on the HD dataset reveals that, although the tree size was increasing, the accuracy was slightly decreasing as the percentage of allowed inherent error in data was increasing while pruning the tree (i.e. the value of CF was increasing). Thus, there was no proportional relationship between the tree size and the accuracy as expected, as shown in Figure 4.3(b). The effect of changing the CF makes the relation almost proportional inverse.

CF Settings	Tree Size	Accuracy Rate %
0.1	19	80.13
0.2	33	78.45
0.3	37	77.78
0.4	61	78.11
0.5	61	78.11
0.6	83	77.44
0.7	83	77.44
0.8	83	77.44
0.9	83	77.44
1	83	77.44

(a) Experimental Results



(b) Plotted Results

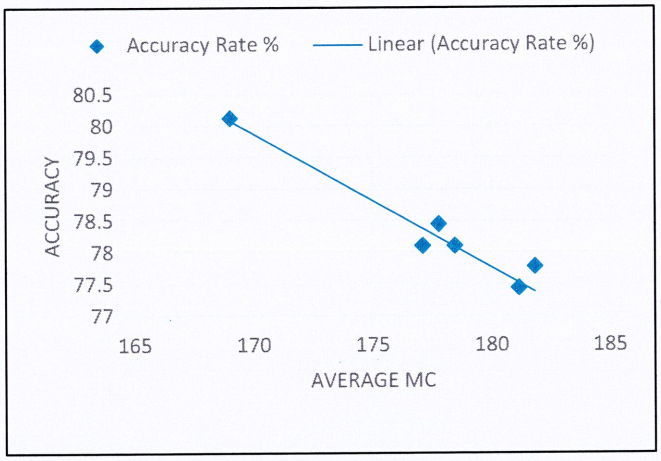
Figure 4.3: The Tree size vs. Accuracy in the HD Dataset

As demonstrated in Figure 4.4, the relationship between the accuracy and the average total MC resulting from tuning the CF was monotonically decreasing in the HD datasets. This means that as the average total MC increases, the accuracy rate will be decreased. This is logical since the more misclassifications will eventually reduce the accuracy. This result

confirms results of Turney (1995) as mentioned in chapter two, section 2.7.

Average MC	Accuracy Rate %
169.02	80.13
177.78	78.45
181.82	77.78
178.45	78.11
177.1	78.11
181.14	77.44
181.14	77.44
181.14	77.44
181.14	77.44
181.14	77.44

(a) Experimental Results



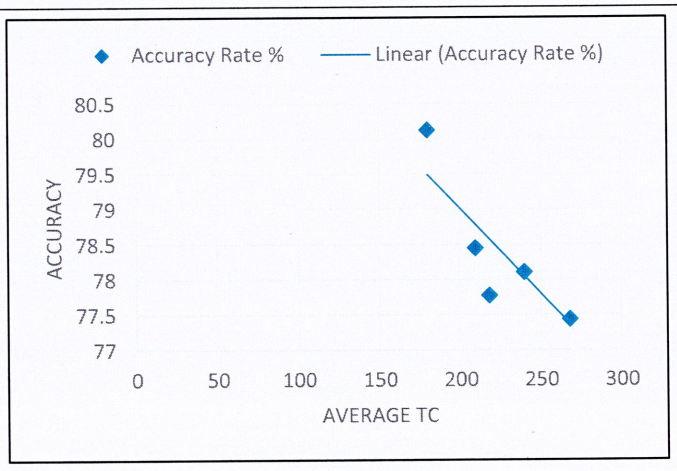
(b) Plotted Results

Figure 4.4: Average MC vs. Accuracy in HD Dataset

In normal cases, increasing the tree size will increase the average total TC. However, as mentioned above, increasing the CF will increase the tree size and decreases accuracy. Therefore, the relationship between the average TC and accuracy becomes monotonically decreaseing as shown in Figure 4.5.

Average TC	Accuracy Rate %
179.84	80.13
209.14	78.45
217.89	77.78
239.52	78.11
239.52	78.11
267.9	77.44
267.9	77.44
267.9	77.44
267.9	77.44
267.9	77.44

(a) Experimental Results



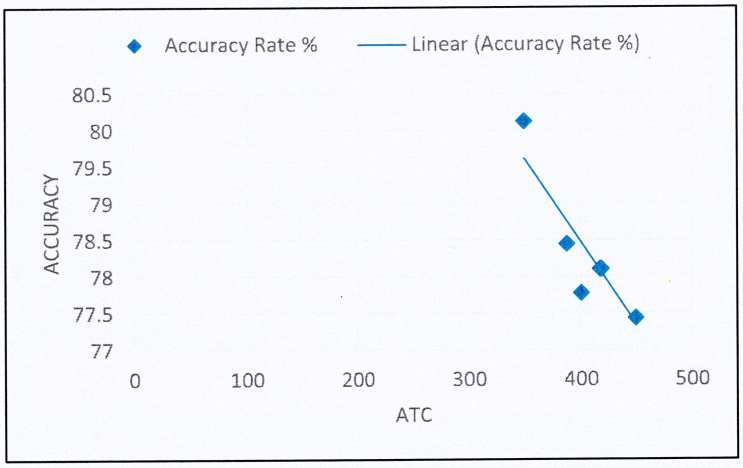
(b) Plotted Results

Figure 4.5: TC vs. Accuracy in HD Dataset

Figure 4.6 shows the resulting relationship between the ATC and accuracy, when both classification cost types were considered. It was also monotonically decreasing as the relationship between average total TC/accuracy and average total MC/accuracy were also monotonically decreasing, considering that the ATC of classification is the sum of the two.

ATC	Accuracy Rate %
348.86	80.13
386.92	78.45
399.71	77.78
417.97	78.11
416.62	78.11
449.04	77.44
449.04	77.44
449.04	77.44
449.04	77.44
449.04	77.44

(a) Experimental Results



(b) Plotted Results

Figure 4.6: ATC Cost vs. Accuracy in HD Dataset

4.3.1.2 Setting the Minimum Number of Objects Parameter

When the MinNumObj parameter was tuned, the tree size was rapidly decreasing in HD dataset (as well as all others) as the minimum number of instances (MinNumObj) at leaf nodes grows. The DT constructed with the MinNumObj initial setting (equals to 5) is illistrated in Figure 4.7.

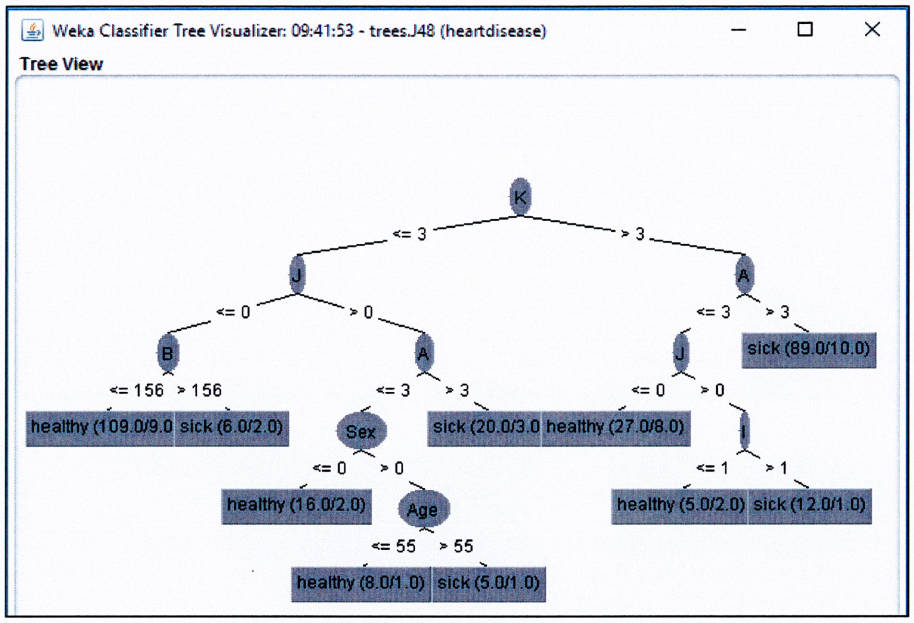
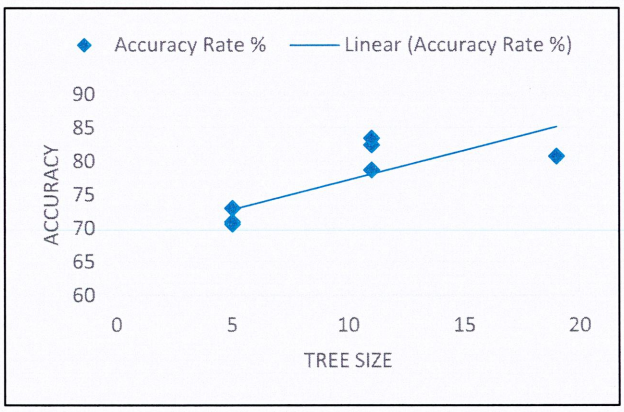


Figure 4.7: The DT Constructed using the MinNumObj Pruning in the HD Dataset

At start in the HD dataset, the accuracy starts increasing as the tree size decreased reaching a peak value of 83.5% but then it drops gradually to 71.04% as the tree size continues decreasing as shown in Figure 4.8(a). However, the overall relationship looks almost proportional as illistrated in Figure 4.8(b).

Tree Size	Accuracy Rate %
19	80.81
11	82.49
11	83.501
11	78.79
5	73.06
5	70.71
5	70.71
5	71.04

(a) Experimental Results



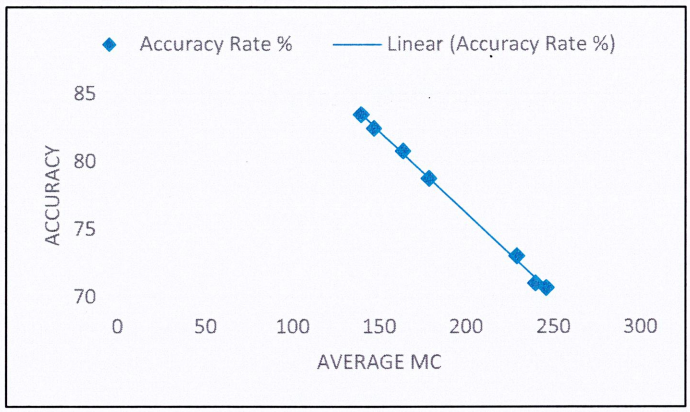
(b) Plotted Results

Figure 4.8: Tree Size vs. Accuracy in the HD Dataset

In addition, the relationship between the average total MC and the accuracy was almost proportional inverse as expected and justified in the previous subsection, 4.3.1.1. Figure 4.9 demonstrates the relationship.

Average MC	Accuracy Rate %
163.64	80.81
146.8	82.49
139.39	83.5
178.45	78.79
228.96	73.06
245.79	70.71
245.79	70.71
239.73	71.04

(a) Experimental Results



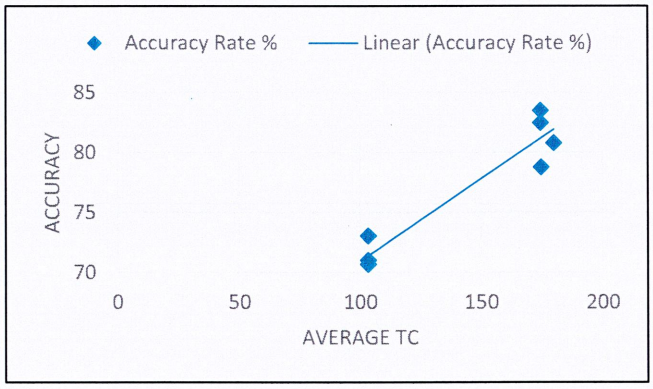
(b) Plotted Results

Figure 4.9: Average MC vs. Accuracy in the HD Dataset

Figure 4.10 showes the relationship between the average total TC and the accuracy in the HD dataset. It is almost proportional as expected because increasing the number of tests (attributes) increases the chance towards making the right decision (or classification). However, in very few cases the average total TC seems to be decreasing while the accuracy was increasing. This is because different costs (weights) were assigned to different types of tests.

Average TC	Accuracy Rate %
179.7	80.81
174.18	82.49
174.18	83.5
174.47	78.79
103.35	73.06
103.35	70.71
103.35	70.71
103.35	71.04

(a) Experimental Results



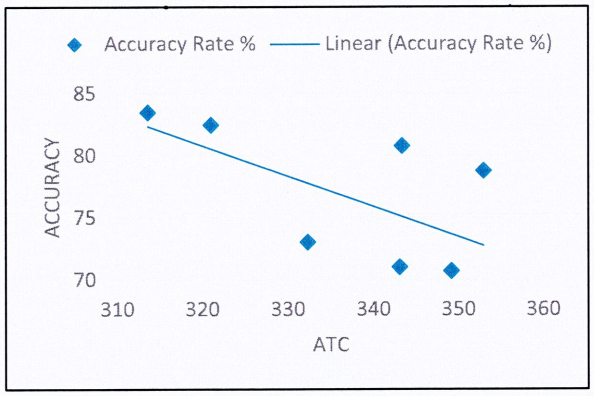
(b) Plotted Results

Figure 4.10: Average TC vs Accuracy in the HD Dataset

The ATC was more affected by the MC, thus the resulting relationship between the ATC and the accuracy was monotically decreasing as shown in Figure 4.11.

ATC	Accuracy Rate %
343.34	80.81
320.98	82.49
313.57	83.5
352.92	78.79
332.31	73.06
349.14	70.71
349.14	70.71
343.08	71.04

(a) Experimental Results



(b) Plotted Results

Figure 4.11: The ATC vs Accuracy in the HD Dataset

4.3.1.3 Setting the Number of Reduced Error Pruning's Folds

Increasing the amount of data used for reduced error pruning method (specified by the NumFold parameter) increases tree size in some cases and decreases it in other cases (fluctuating) in the HD dataset as well as in all other datasets. Figure 4.12 demonstrates the DT obtained when the NumFold parameter was set to 3.

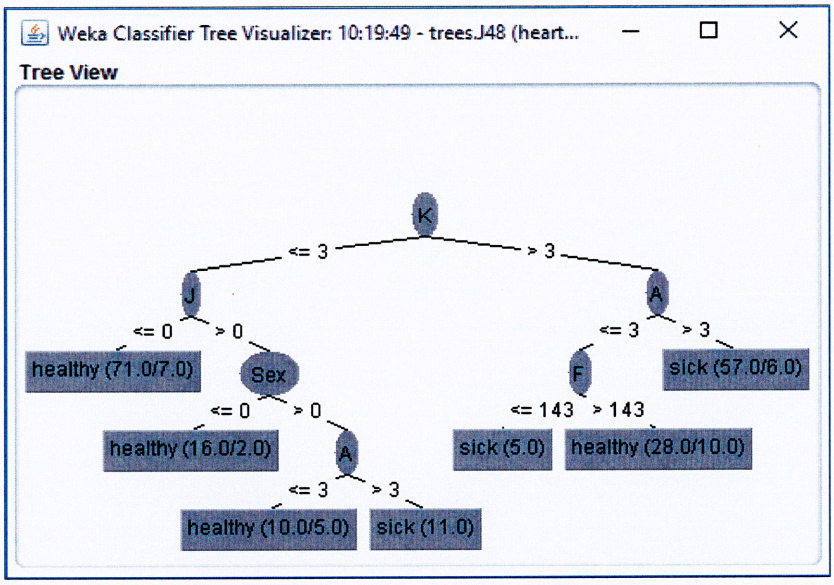
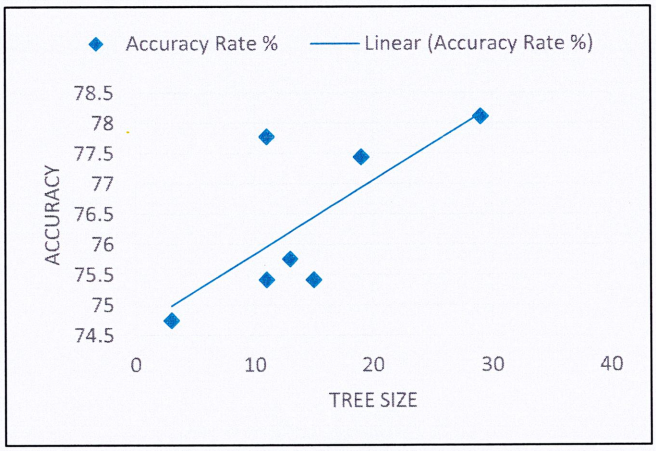


Figure 4.12: The DT Constructed using the NumFold Pruning in the HD Dataset

The relationship between the tree size and the accuracy resulting from tuning the NumFold parameter was almost proportional in HD as illustrated in Figure 4.13.

Tree Size	Accuracy Rate %
13	75.76
11	77.78
29	78.11
11	75.42
19	77.44
15	75.42
3	74.75

(a) Experimental Results



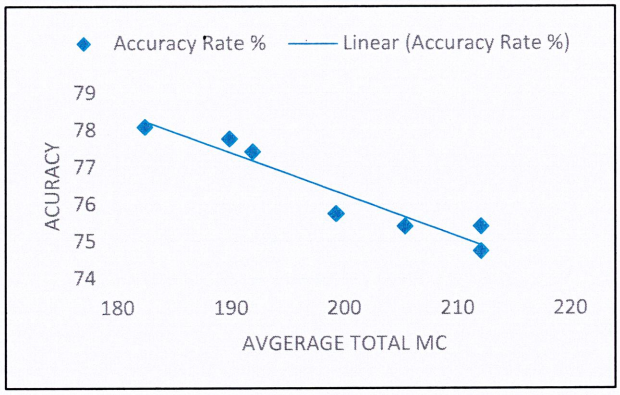
(b) Plotted Results

Figure 4.13: Tree Size vs. Accuracy in the HD Dataset

Like when tuning the CF and MinNumObj parameters in HD dataset, a proportional inverse relationship exists between the average total MC and the classification accuracy as shown in Figure 4.14.

Average MC	Accuracy Rate %
199.3	75.76
189.9	77.78
182.49	78.11
212.12	75.42
191.92	77.44
205.39	75.42
212.12	74.75

(a) Experimental Results



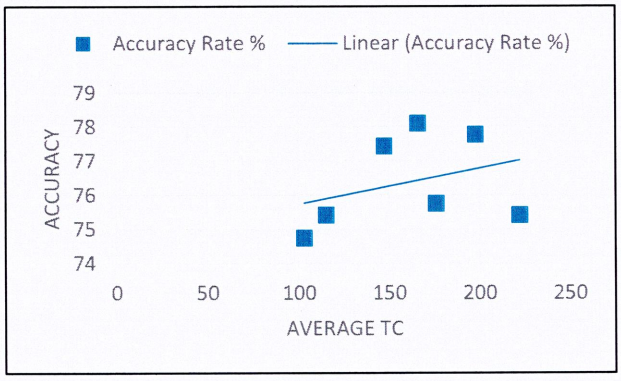
(b) Plotted Results

Figure 4.14: The MC Cost vs. Accuracy in the HD Dataset

In most of the cases, the relationship between the accuracy and the average total TC is almost proportional as shown in Figure 4.15. This is logical since the relationship between the tree size and the classifier's accuracy is proportional and the relationship between the tree size and TC is also proportional. However, in a few cases, the opposite occured since larger trees may have lower TC as the attribute costs differ for different attributes.

Average TC	Accuracy Rate %
175.83	75.76
197.57	77.78
165.76	78.11
222.03	75.42
147.01	77.44
114.91	75.42
102.9	74.75

(a) Experimental Results



(b) Plotted Results

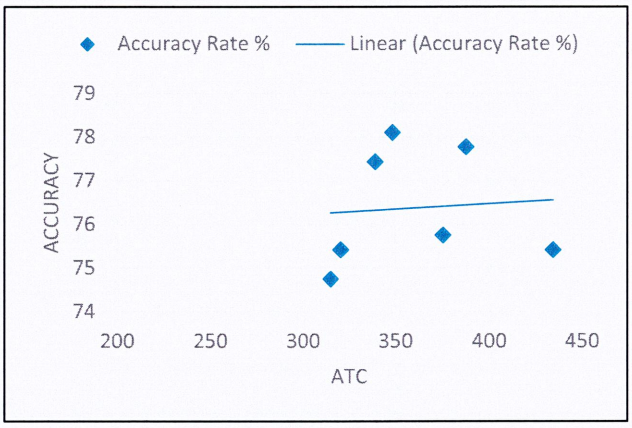
Figure 4.15: The Average TC vs Accuracy in the HD Dataset

Although, the association between the average total MC and the accuracy is proportional inverse, the resulting relationship between the ATC and

the classification accuracy is almost proportional as shown in Figure 4.16. It seems that the average total TC affects the ATC more as the relationship between the average total TC and accuracy was almost proportional.

ATC	Accuracy Rate %
375.13	75.76
387.47	77.78
348.25	78.11
434.15	75.42
338.93	77.44
320.3	75.42
315.02	74.75

(a) Experimental Results



(b) Plotted Results

Figure 4.16: The ATC vs Accuracy in the HD Dataset

4.3.2 Thyroid Disease Dataset

This section discusses the experimental results obtained when tuning the CF, MinNumObj, and NumFolds pruning parameters while constructing a DT using the thyroid disease dataset.

4.3.2.1 Setting the Confidence Factor Parameter

The first pruning parameter that was tuned in thyroid dataset is the CF parameter, a batch of DTs was obtained. Figure 4.17 shows the DT acquired when the CF parameter was set to 0.1.

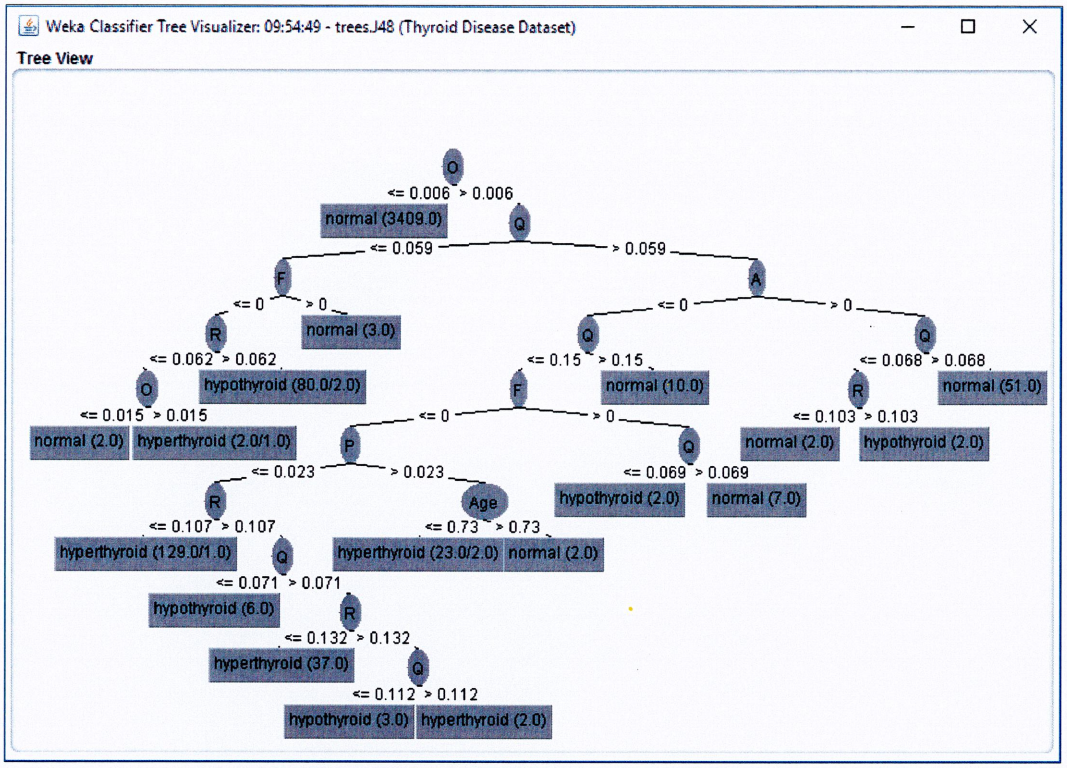


Figure 4.17: The DT Constructed using the CF Pruning in the Thyroid Disease Dataset

In thyroid dataset, changing the CF slightly changes the tree size as shown in Table 4.2. As a result, the accuracy was almost the same for different tree sizes, taking into consideration that the accuracy was very high ranging from 99.13 to 99.18.

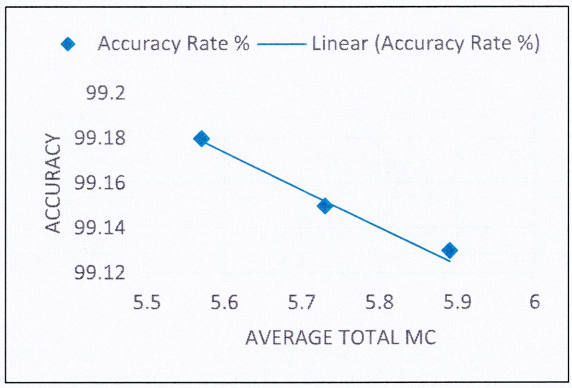
Table 4.2: Experimental Results on the Thyroid Dataset

CF Settings	Tree Size	Accuracy Rate %
0.1	35	99.13
0.2	35	99.18
0.3	35	99.18
0.4	35	99.18
0.5	35	99.18
0.6	39	99.15
0.7	39	99.15
0.8	39	99.15
0.9	39	99.15
1	39	99.15

The relationship between the accuracy and the average total MC was proportional inverse as expected, but the difference between accuricies was 0.04 on average. In addition, the average total TC was almost the same for all CF settings as the resulting trees were very similar, almost of the same size, and the classifier's accuracy was very high. The two relations are displayed in Figure 4.18 and 4.19, respectively. This confirms the results of Patel (2012).

Average MC	Accuracy Rate %
5.89	99.13
5.57	99.18
5.57	99.18
5.57	99.18
5.57	99.18
5.73	99.15
5.73	99.15
5.73	99.15
5.73	99.15
5.73	99.15

(a) Experimental Results

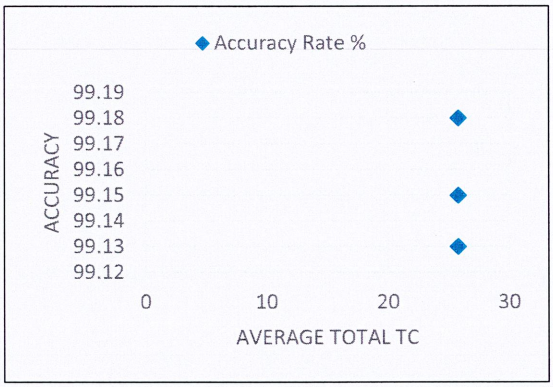


(b) Plotted Results

Figure 4.18: The Average MC vs Accuracy in the Thyroid Disease Dataset

Average TC	Accuracy Rate %
25.75	99.13
25.75	99.18
25.75	99.18
25.75	99.18
25.75	99.18
25.75	99.15
25.75	99.15
25.75	99.15
25.75	99.15
25.75	99.15

(a) Experimental Results



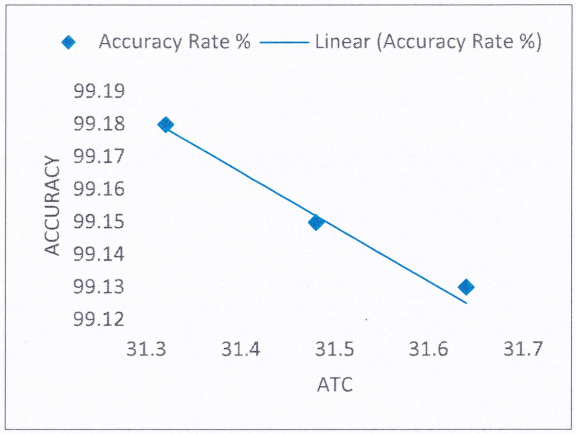
(b) Plotted Results

Figure 4.19: Average TC vs. Accuracy in the Thyroid Dataset

When the ATC was compared to accuracy, the resulting relationship was proportional inverse as shown in Figure 4.20.

ATC	Accuracy Rate %
31.64	99.13
31.32	99.18
31.32	99.18
31.32	99.18
31.32	99.18
31.48	99.15
31.48	99.15
31.48	99.15
31.48	99.15
31.48	99.15

(a) Experimental Results



(b) Plotted Results

Figure 4.20: ATC vs. Accuracy in Thyroid Disease Dataset

4.3.2.2 Setting the Minimum Number of Objects Parameter

Similar to the HD dataset, when the MinNumObj parameter was tested for the thyroid disease dataset, the tree size was decreasing as the MinNumObj grow. Figure 4.21 shows the DT constructed when the the MinNumObj was set to 5 in thyroid disease dataset.

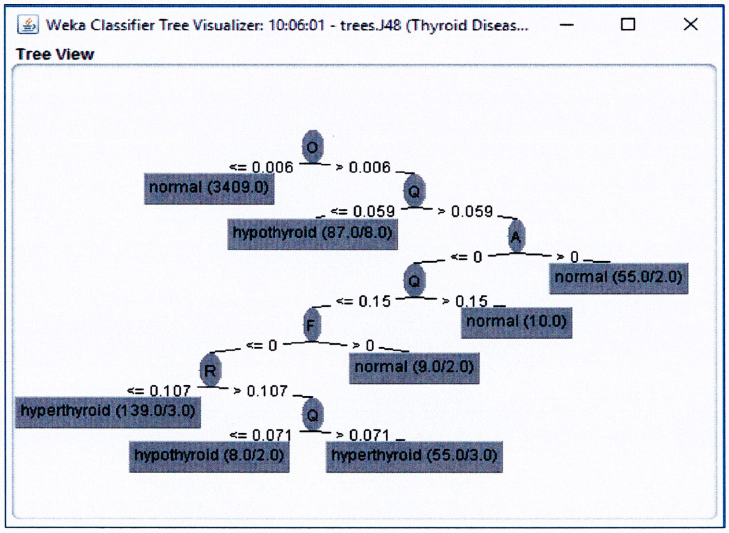
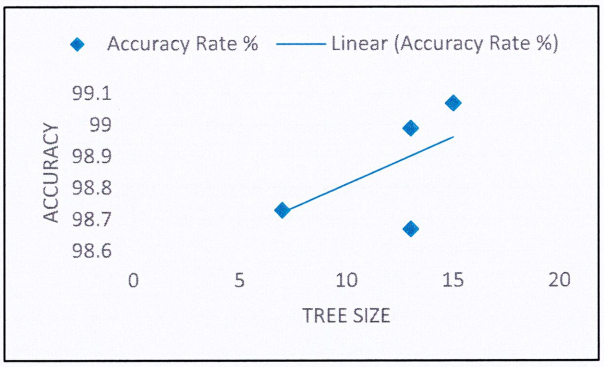


Figure 4.21: The DT Constructed using the MinNumObj Pruning in the Thyroid Disease Dataset

It has been found that the classifier's accuracy drops slightly as the tree size decreased in thyroid dataset, which indicates the existence of proportional relationship between the two as expected and confirmed by Neto, et al. (2017). Figure 4.22, shows the relationship between the tree size and the overall classifier's accuracy.

MinNumObj	Tree Size	Accuracy Rate %
5	15	99.07
10	13	98.99
15	13	98.67
20	7	98.73
25	7	98.73
30	7	98.73
35	7	98.73
40	7	98.73

(a) Experimental Results



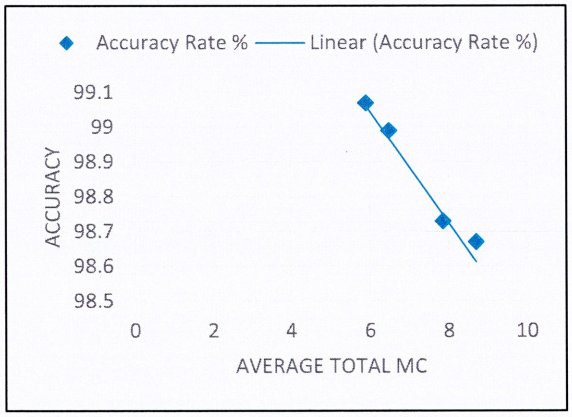
(b) Plotted Results

Figure 4.22: Tree Size vs. Accuracy in the Thyroid Disease Dataset

The relationship between the average total MC and the accuracy was proportional inverse as it supposed to be, and the relationship between the average total TC and the accuracy was almost proportional as expected and explained in section 4.3.1.2. The relationships are shown in Figures 4.23 and 4.24, respectively.

Average MC	Accuracy Rate %
5.89	99.07
6.47	98.99
8.7	98.67
7.85	98.73
7.85	98.73
7.85	98.73
7.85	98.73
7.85	98.73

(a) Experimental Results

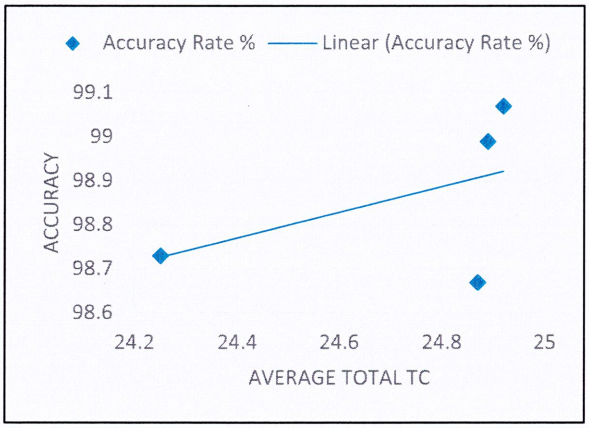


(b) Plotted Results

Figure 4.23: The Average MC vs Accuracy in the Thyroid Disease Dataset

Average TC	Accuracy Rate %
24.92	99.07
24.89	98.99
24.87	98.67
24.25	98.73
24.25	98.73
24.25	98.73
24.25	98.73
24.25	98.73

(a) Experimental Results



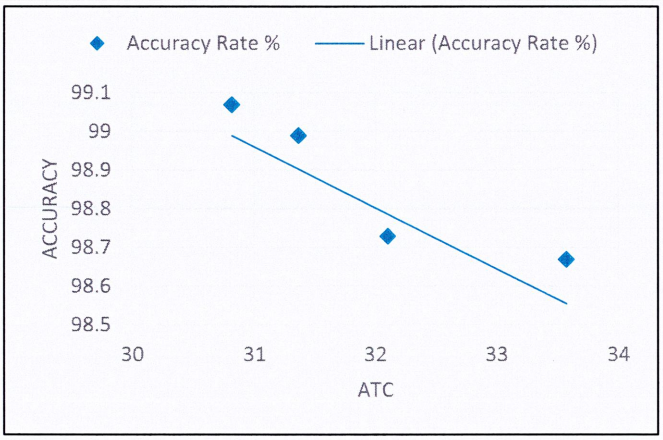
(b) Plotted Results

Figure 4.24: The Average TC vs Accuracy in the Thyroid Disease Dataset

The resulting relation between the ATC and the accuracy was almost proportional inverse as illistrated in Figure 4.25.

ATC	Accuracy Rate %
30.81	99.07
31.36	98.99
33.57	98.67
32.1	98.73
32.1	98.73
32.1	98.73
32.1	98.73
32.1	98.73

(a) Experimental Results



(b) Plotted Results

Figure 4.25: The ATC vs Accuracy in the Thyroid Disease Dataset

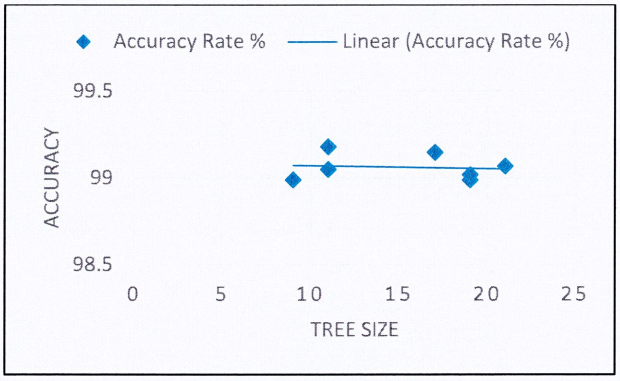
4.3.2.3 Setting the Number of Reduced Error Pruning's Folds

When the NumFold pruning parameter was tested in thyroid disease dataset, the relationship between the tree size and the accuracy found was

almost constant as illistrated in Figure 4.26. The accuracy was almost the same for different tree sizes as the classifier's accuracy was very high.

Tree Size	Accuracy Rate %
19	98.99
11	99.18
21	99.07
17	99.15
11	99.05
19	99.02
9	98.99

(a) Experimental Results



(b) Plotted Results

Figure 4.26: Tree Size vs. Accuracy in the Thyroid Dataset

Figure 4.27 shows the tree acquired when the NumFold parameter was set to its default setting (equals to 3).

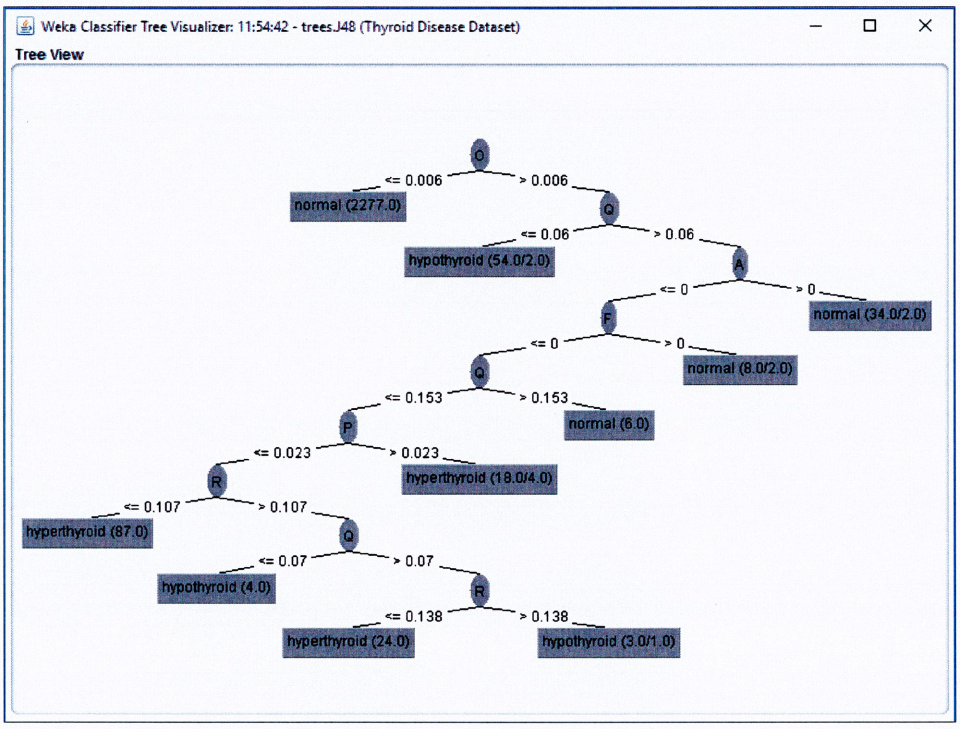
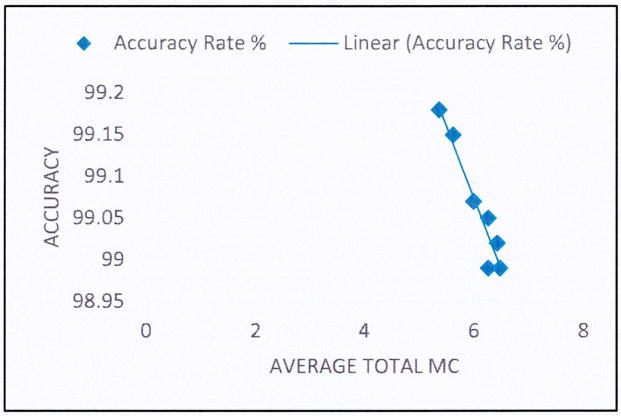


Figure 4.27: The DT Constructed using the NumFold Pruning in the Thyroid Disease Dataset

As shown in Figure 4.28, a proportional inverse relationship exists between the average total MC and the classification accuracy. However, the difference in accuracy was very small. In addition, Figure 4.29 shows a very slight change in the accuracy when compared to the average total TC for the same reason mentioned above.

Average MC	Accuracy Rate %
6.47	98.99
5.36	99.18
5.99	99.07
5.62	99.15
6.26	99.05
6.42	99.02
6.26	98.99

(a) Experimental Results

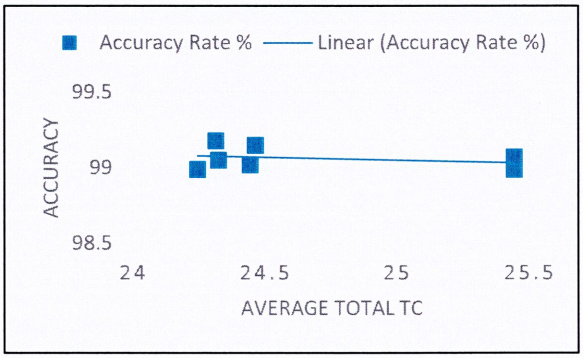


(b) Plotted Results

Figure 4.28: The Average MC vs Accuracy in the Thyroid Disease Dataset

Average TC	Accuracy Rate %
25.44	98.99
24.31	99.18
25.44	99.07
24.46	99.15
24.32	99.05
24.44	99.02
24.24	98.99

(a) Experimental Results



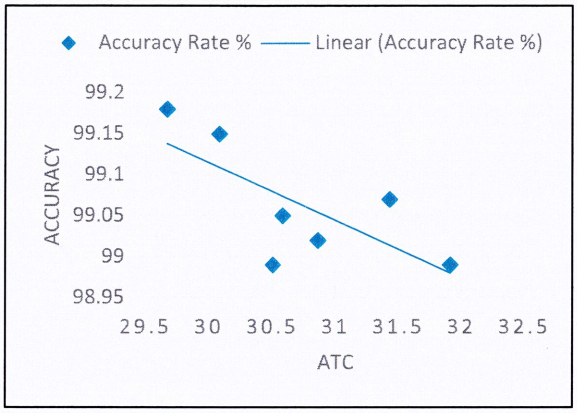
(b) Plotted Results

Figure 4.29: The Average TC vs Accuracy in the Thyroid Disease Dataset

The classifier's accuracy was almost proportional inverse with the ATC. However, a very slight change in the accuracy was noticed in Figure 4.30.

ATC	Accuracy Rate %
31.91	98.99
29.67	99.18
31.43	99.07
30.08	99.15
30.58	99.05
30.86	99.02
30.5	98.99

(a) Experimental Results



(b) Plotted Results

Figure 4.30: ATC vs Accuracy in the Thyroid Disease Dataset

4.3.3 Breast Cancer Dataset

This section discusses the experimental results obtained when tuning the CF, MinNumObj and NumFolds pruning parameters while constructing a DT using the breast cancer dataset.

4.3.3.1 Setting the Confidence Factor Parameter

When the CF was tuned in breast cancer dataset, the initial tree obtained (when CF parameter equals to 0.1) is shown in Figure 4.31. The experimental results showed that as the tree size increased, the accuracy was slightly decreased as with the other two datasets.

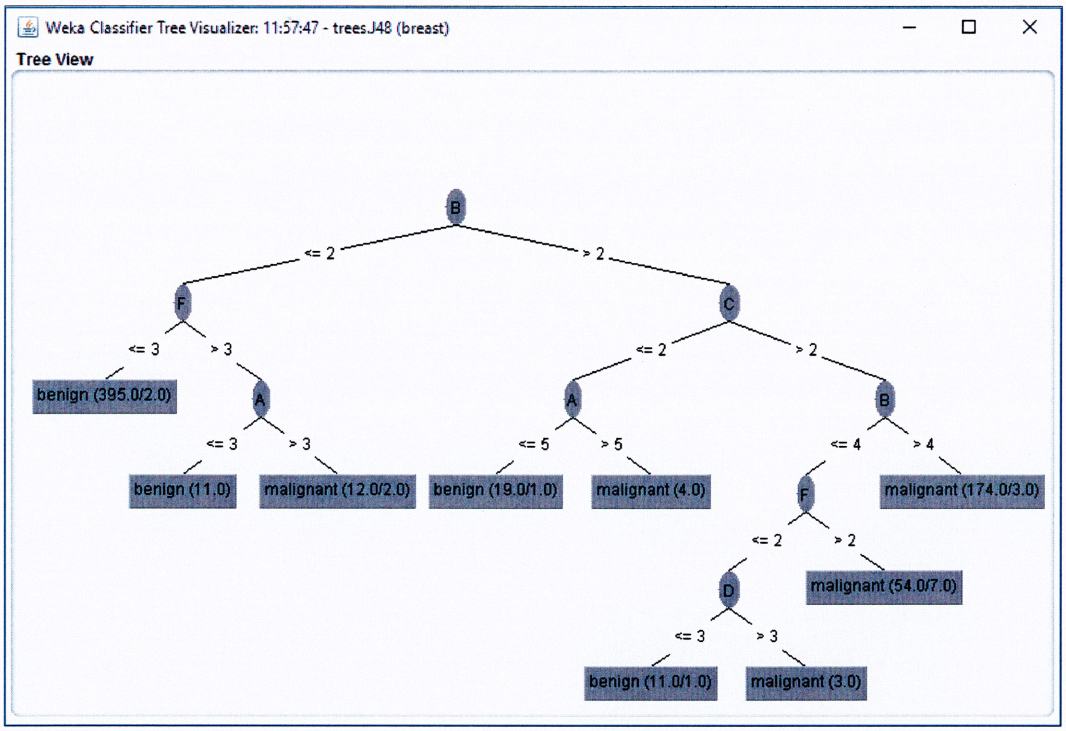
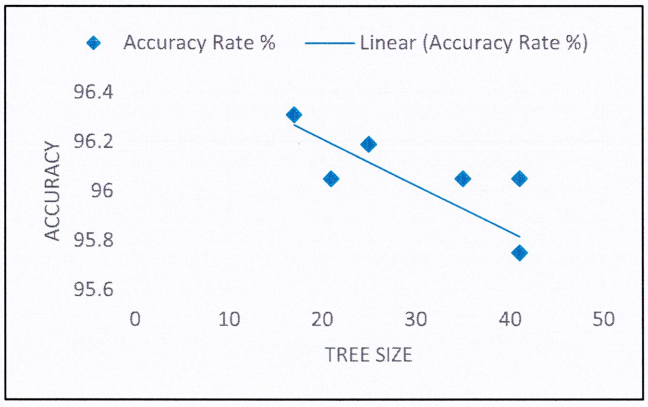


Figure 4.31: The DT Constructed using the CF Parameter in the Breast Cancer Dataset

The relationship between the tree size and the accuracy was almost proportional inverse, as Figure 4.32 shows.

CF Settings	Tree Size	Accuracy Rate %
0.1	17	96.31
0.2	21	96.05
0.3	25	96.19
0.4	35	96.05
0.5	41	96.05
0.6	41	95.75
0.7	41	95.75
0.8	41	95.75
0.9	41	95.75
1	41	95.75

(a) Experimental Results



(b) Plotted Results

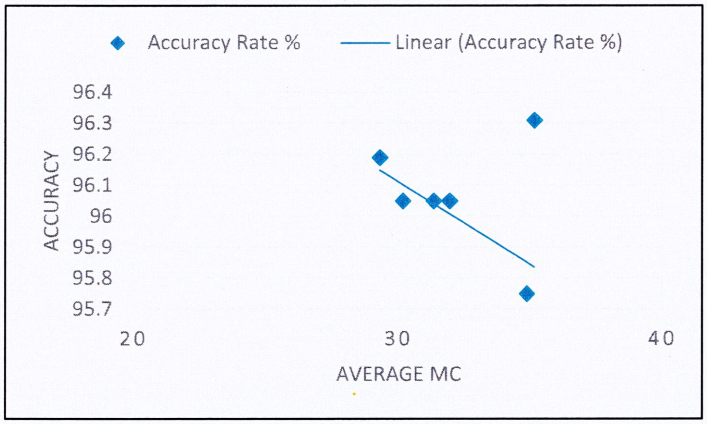
Figure 4.32: Tree Size vs. Accuracy in the Breast Cancer Dataset

The relationship between the accuracy and the average total MC was almost proportional inverse and the relationship between the average TC

and accuracy was monotonically decreasing for the same reason mentioned in section 4.3.1.1. The two relationships are shown in Figures 4.33 and 4.34, respectively.

Average MC	Accuracy Rate %
35.14	96.31
30.16	96.05
29.28	96.19
31.33	96.05
31.92	96.05
34.85	95.75
34.85	95.75
34.85	95.75
34.85	95.75
34.85	95.75

(a) Experimental Results

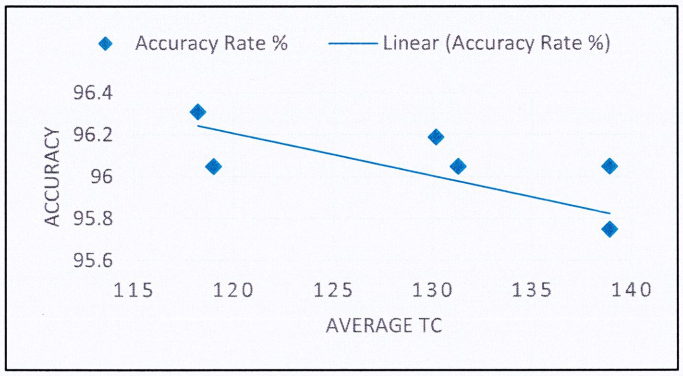


(b) Plotted Results

Figure 4.33: The Average MC Vs Accuracy in the Breast Cancer Dataset

Average TC	Accuracy Rate %
118.19	96.31
118.97	96.05
130.13	96.19
131.23	96.05
138.86	96.05
138.86	95.75
138.86	95.75
138.86	95.75
138.86	95.75
138.86	95.75

(a) Experimental Results



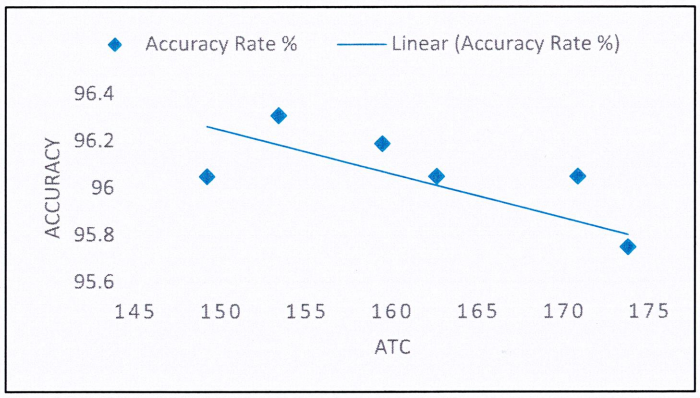
(b) Plotted Results

Figure 4.34: Average TC vs. Accuracy in Breast the Cancer Dataset

The resulting relationship between the ATC and the accuracy, when both classification cost types were considered, was almost proportional inverse as Figure 4.35 shows. This is acceptable since the relationship between the average total TC/accuracy was decreasing monotically, the relationship between the average total MC/accuracy was proportional inverse, considering that the ATC of classification is the sum of the two.

ATC	Accuracy Rate %
153.33	96.31
149.13	96.05
159.41	96.19
162.56	96.05
170.78	96.05
173.71	95.75
173.71	95.75
173.71	95.75
173.71	95.75
173.71	95.75

(a) Experimental Results



(b) Plotted Results

Figure 4.35: The ATC vs. Accuracy in the Breast Cancer Dataset

4.3.3.2 Setting the Minimum Number of Objects Parameter

When the MinNumObj parameter was tested for the breast cancer dataset, a batch of DT was constructed. Figure 4.36 shows the tree resulting from setting the MinNumObj parameter to 5.

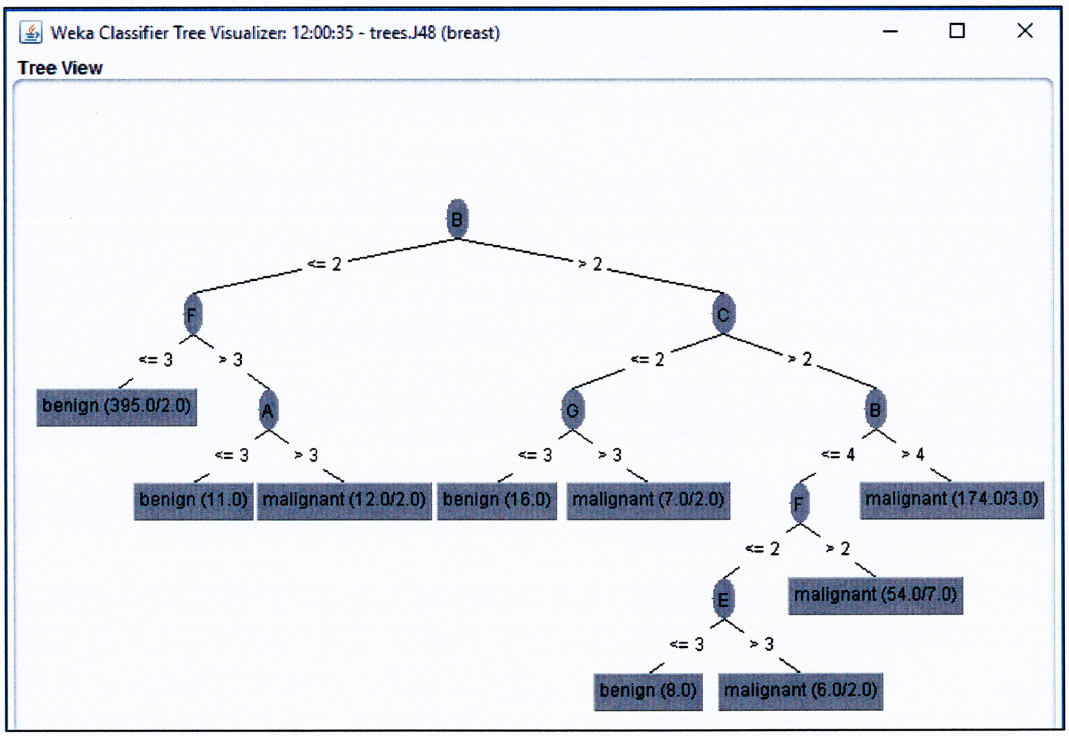
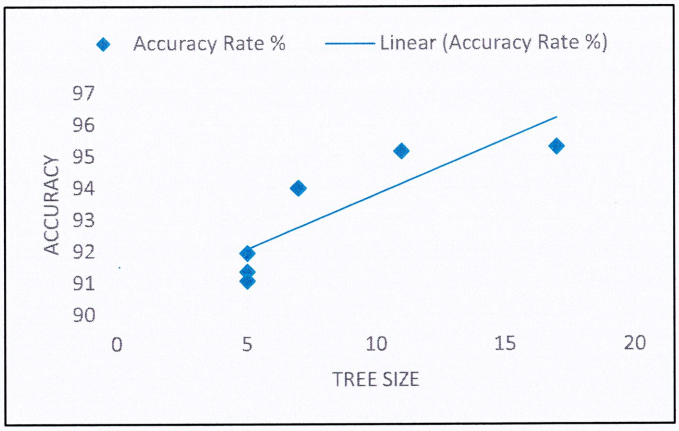


Figure 4.36: The DT Constructed using the MinNumObj Pruning in the Breast Cancer Dataset

The experimental results showed that the tree size was decreasing as MinNumObj grows. The results also showed that, the classifier's accuracy was decreasing as the tree size decreased, i.e. there is an almost proportional relationship between the two as shown in Figure 4.37.

MinNumObj	Tree Size	Accuracy Rate %
5	17	95.31
10	11	95.17
15	7	94
20	7	94
25	5	91.36
30	5	91.36
35	5	91.95
40	5	91.07

(a) Experimental Results



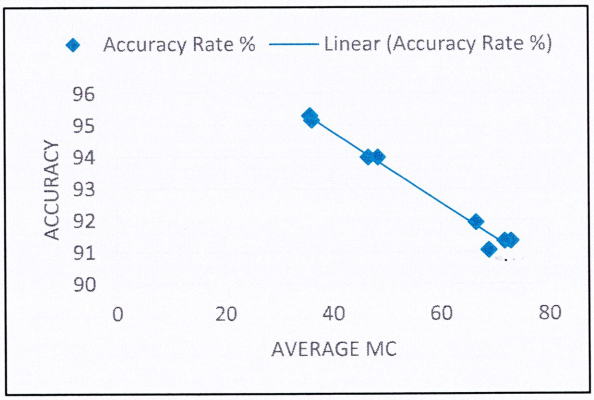
(b) Plotted Results

Figure 4.37: Tree Size vs. Accuracy in the Breast Cancer Dataset

In addition, the relationship between the average total MC and the accuracy was almost proportional inverse and the relationship between the average total TC and the accuracy was almost proportional as expected. Both relations are shown in Figure 4.38 and 4.39, respectively.

Average MC	Accuracy Rate %
35.72	95.31
36.02	95.17
46.56	94
48.32	94
72.91	91.36
71.74	91.36
66.47	91.95
68.81	91.07

(a) Experimental Results

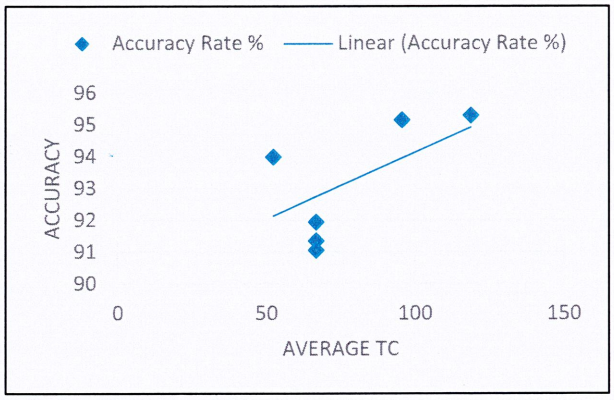


(b) Plotted Results

Figure 4.38: The Average MC vs Accuracy in the Breast Cancer Dataset

Average TC	Accuracy Rate %
118.61	95.31
95.63	95.17
52.35	94
52.35	94
66.88	91.36
66.88	91.36
66.88	91.95
66.88	91.07

(a) Experimental Results



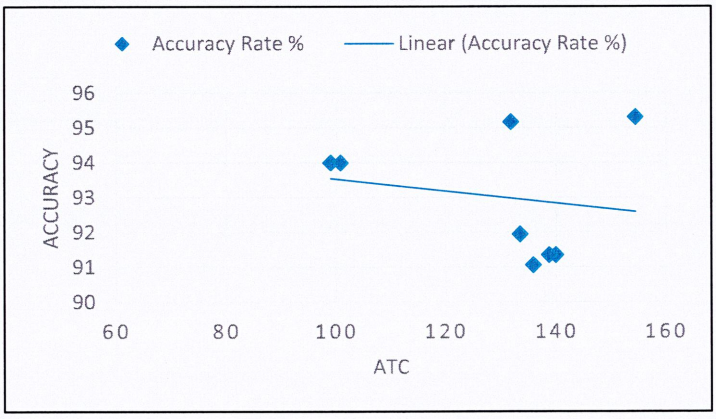
(b) Plotted Results

Figure 4.39: The Average TC vs Accuracy in the Breast Cancer Dataset

The resulting relationship between the ATC and the classification accuracy was almost proportional inverse as illistrated in Figure 4.40.

ATC	Accuracy Rate %
154.33	95.31
131.65	95.17
98.91	94
100.67	94
139.79	91.36
138.62	91.36
133.35	91.95
135.69	91.07

(a) Experimental Results



(b) Plotted Results

Figure 4.40: The ATC vs Accuracy in the Breast Cancer Dataset

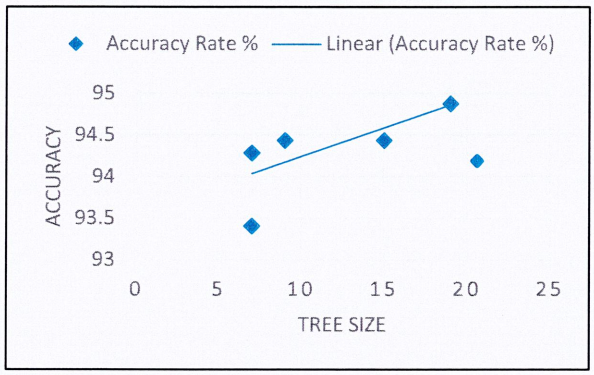
4.3.3.3 Setting the Number of Reduced Error Pruning's Folds

The experimental results on the breast cancer dataset when the NumFold pruning parameter was tuned, showed an almost proportional relationship between the tree size and the accuracy and an almost proportional inverse relationship between the average total MC and the classification accuracy

(like the other two datasets), as shown in Figures 4.41 and 4.42, respectively.

NumFolds	Tree Size	Accuracy Rate %
3	19	94.88
5	15	94.44
7	7	94.29
9	21	94.14
11	7	93.41
13	9	94.44
15	9	94.44

(a) Experimental Results

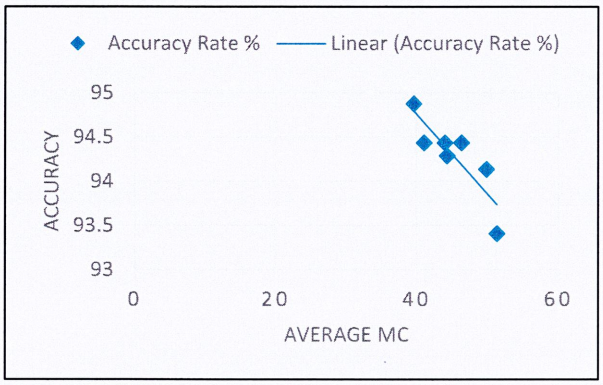


(b) Plotted Results

Figure 4.41: Tree Size vs. Accuracy in the Breast Cancer Dataset

Average MC	Accuracy Rate %
39.53	94.88
46.27	94.44
44.22	94.29
49.78	94.14
51.24	93.41
43.92	94.44
41	94.44

(a) Experimental Results



(b) Plotted Results

Figure 4.42: Tree Size vs. Accuracy in the Breast Cancer Dataset

The DT resulting from having the NumFold paramter equals to 3 (the default setting) is shown in Figure 4.43.

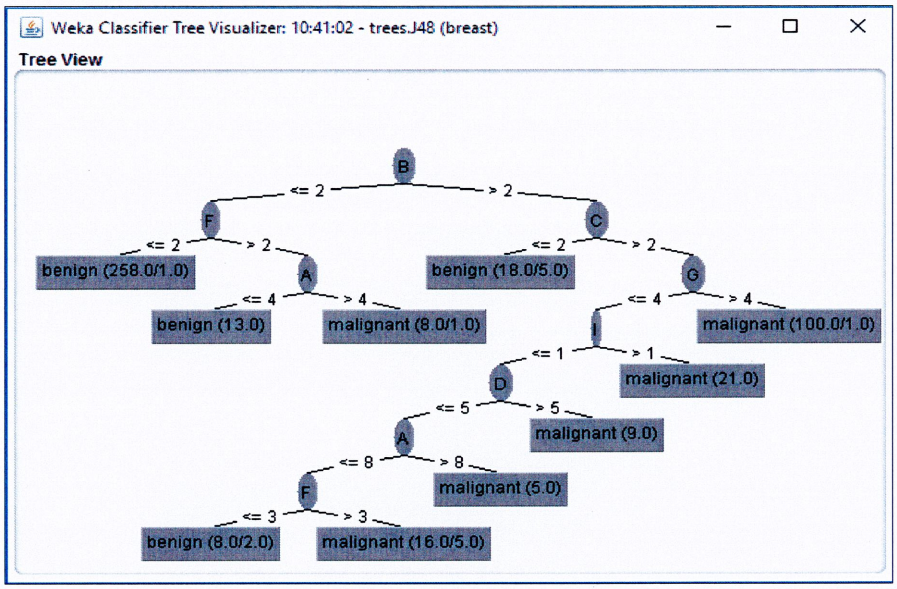
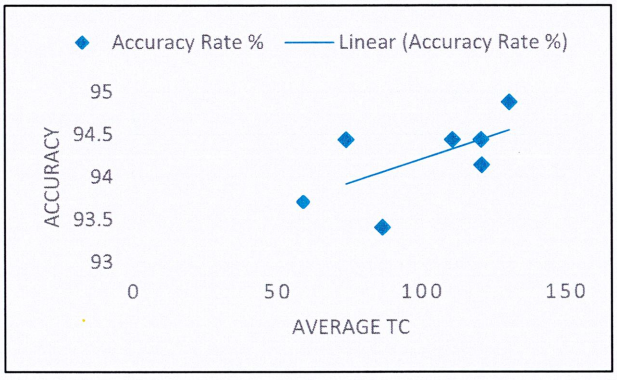


Figure 4.43: The DT Constructed using the NumFold Pruning in the Breast Cancer Dataset

In most of the cases, the relationship between the accuracy and average total TC was almost proportional and the resulting relationship between the ATC and the classification accuracy was also almost proportional for the same reason mentioned in Section 4.3.1.3.

Average TC	Accuracy Rate %
130.08	94.88
120.23	94.44
52.2	94.29
120.41	94.14
86.14	93.41
73.68	94.44
110.32	94.44

(a) Experimental Results



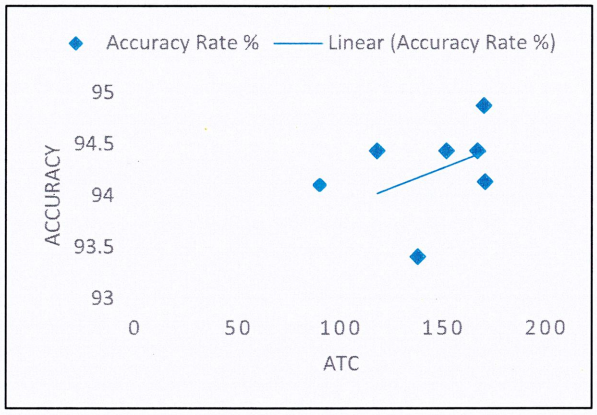
(b) Plotted Results

Figure 4.44: The Average TC vs Accuracy in the Breast Cancer Dataset

Scattered diagrams of the relationships are shown in Figures 4.44 and 4.45.

ATC	Accuracy Rate %
169.61	94.88
166.5	94.44
96.42	94.29
170.19	94.14
137.38	93.41
117.6	94.44
151.32	94.44

(a) Experimental Results



(b) Plotted Results

Figure 4.45: The ATC vs Accuracy in the Breast Cancer Dataset

4.3.4 Datasets Comparision

This section compares the results obtained from using different DT pruning methods, including pre-pruning and post pruning methods. The resulting relationships obtained using the HD, thyroid disease, and breast cancer datasets are summerized in Table 4.3.

Table 4.3: The Resulting Relations using All Datasets

Relationships	Pruning Parameters	HD Dataset	Thyroid Dataset	Breast Cancer Dataset
Tree Size vs. Accuracy	CF	Almost proportional inverse	Almost constant	Almost proportional inverse
	MinNumObj	Almost proportional	Almost proportional	Almost proportional
	NumFold	Almost proportional	Almost constant	Almost proportional
MC vs. Accuracy	CF	Monotonically decreasing	Proportional inverse	Almost proportional inverse
	MinNumObj	Almost proportional inverse	Proportional inverse	Almost proportional inverse
	NumFold	Proportional inverse	Proportional inverse	Almost proportional inverse
TC vs. Accuracy	CF	Monotonically decreasing	Almost constant	Monotonically decreasing
	MinNumObj	Almost proportional	Almost proportional	Almost proportional
	NumFold	Almost proportional	Almost constant	Almost proportional
Total Classification Cost vs. Accuracy	CF	Monotonically decreasing	Proportional inverse	Almost proportional inverse
	MinNumObj	Monotonically decreasing	Almost proportional inverse	Almost proportional inverse
	NumFold	Almost proportional	Almost proportional inverse	Almost proportional

From the above table it is clear that the association between the ATC and accuracy depends on the relationship between the average total MC with the classifier's accuracy and the average total TC with the accuracy as the ATC is the total of the average total MC and the average total TC.

The relationship between the average total MC and the classifier's accuracy was almost proportional inverse in all datasets regardless their size, number of classes, number of dimensions, or the classes distribution. The same relation was obtained regardless the pruning method used.

On the other hand, the relationship between the average total TC with accuracy was monotonically decreasing in HD and breast cancer datasets when the CF parameter was tuned as shown in Tables 4.3. The reason behind having a monotonically decreasing relationship between the two is that increasing the CF means increasing the percentage of allowed error during the DT pruning. In addition, increasing the CF increases the tree size, thus the relationship between the tree size and accuracy becomes almost proportional inverse and as the average total TC depends heavily on the tree size, its relationship with accuracy becomes monotonically decreasing.

Additionally, as tuning the CF slightly changes the tree size and as the classifier's accuracy was very high when thyroid disease dataset was used, the resulting relationship between the average total TC and accuracy was almost constant.

Moreover, when the MinNumObj and NumFold parameters were tuned, the relationship between the average total TC and accuracy was almost proportional in all datasets as expected and stated in Tables 4.3 except for the thyroid disease dataset, it was almost constant for the same reason mentioned in the previous paragraph.

The resulting relationship between the ATC and the accuracy was proportional inverse or monotonically decreasing in all datasets when the CF or MinNumObj were considered. However, when the NumFold was considered, it was almost proportional in HD and breast cancer datasets and almost proportional inverse in thyroid disease dataset.

This indicates that the dataset characteristics may affect the overall relationship between the ATC and accuracy. However, other factors can

affect the relationship too including the TCs assigned to each dataset attribute and the cost of misclassifications.

4.4 Results Validation

In addition to representing associations graphically using scattered plots, this section conducts a correlation analysis to express the relationship between the ATC and accuracy numerically. The Pearson correlation coefficient was computed using SPSS to determine whether the association between the classification accuracy and ATC is negative (inverse) or positive (proportional), and how significant the association is. It also indicates whether a weak, moderate, or strong association exists.

Absolute Pearson correlation values between zero and 0.3 indicate weak association, and when the values are between 0.31 and 0.6 indicate a moderate association, whereas when Pearson correlation values are above 0.6, it indicate a strong association (Gerber & Finn, 2005).

Table 4.4 summarizes all correlation matrices resulting from the correlation analysis.

Table 4.4: Correlation Matrix for all Datasets

Dataset	Correlations	CF	MinNumObj	NumFold
HD Disease	Pearson Correlation	-.929**	-.630	.078
	Sig. (2-tailed)	.000	.094	.868
	N	10	8	7
Thyroid Disease	Pearson Correlation	-.992**	-.844**	-.719
	Sig. (2-tailed)	.000	.008	.069
	N	10	8	7
Breast Cancer Disease	Pearson Correlation	-.818**	-.187	.227
	Sig. (2-tailed)	.007	.658	.624
	N	10	8	7

According to Table 4.4, the relationship between the ATC and the accuracy in the HD dataset is negative (or inverse) when the CF pruning parameter was used. This asserts the results obtained from the scattered graph in Figure 4.6. It is very strong from both sides as its absolute value is very close to one (0.929).

The correlation matrix also shows a significant relationship indicated by the sig (2-tailed)* row. The sig value (.000), which implies that its value is below 0.0005, is considered significant, as the correlation is significant at 0.01 or below. This means that the ATC is significantly related to the classification accuracy.

---

*Sig. (2- tailed) indicates the level of statistical significance in both directions (Pallant, 2010).

The Pearson correlation coefficient results shown in Table 4.4 also indicates a negative, and a strong relationship between the ATC and the accuracy resulting when tuning the MinNumObj parameter. The same result was shown in the scattered plot in Figure 4.11. However, the Sig. (2-tailed) row indicates that the relationship is insignificant.

The correlation matrix also indicates that the correlation between the ATC and the accuracy resulting from tuning the Numfold pruning parameter is positive, weak, and insignificant using the HD dataset. The graphical display in Figure 4.16 also showed a positive relation.

In thyroid disease dataset, the correlation matrix shown in Table 4.4 shows a negative, strong relationship between the ATC and accuracy when CF pruning parameter was used. It also shows that the two are significantly related to each other as the significance is below 0.01. Thus, the result obtained in Figure 4.20 is verified.

As what the graph of Figure 4.25 illustrated, the correlation matrix shown in Table 4.4 reflects a negative, strong, and significant relationship between the ATC and the accuracy when tuning the MinNumObj parameter in the thyroid disease dataset.

When the NumFold pruning parameter is used with different settings, the association between the ATC and the accuracy was negative, strong, but insignificant. This verifies the result obtained in Figure 4.30.

Like the HD and the thyroid datasets, the Pearson correlation coefficient value (-0.818) in Table 4.4 obtained when the breast cancer dataset and the CF pruning method were used indicated an inverse, strong, and significant relationship between the ATC and accuracy. This asserts the result obtained in Figure 4.35.

The computed Pearson correlation coefficient results show a negative, weak, and insignificant association between the ATC and accuracy when the MinNumObj pruning parameter was used with breast cancer dataset. Figure 4.40 also showed a negative relationship between the two.

Moreover, the correlation matrix shown in Table 4.4 shows a positive, weak, and insignificant association between the ATC and accuracy when the NumFold pruning parameter was adjusted in breast cancer dataset. A positive relationship was also obtained in Figure 4.45.

Table 4.5 summarizes all associations between the ATC and the accuracy in all datasets using different pruning methods. All matrices show a negative association between the ATC and the accuracy except the two highlighted cases in Table 4.5. However, the two positive associations were very weak and insignificant.

Thus, the relationship between the ATC and the accuracy can be expressed as a negative relationship in the selected datasets.

Table 4.5: Summery of all Correlation Matrices

Dataset	Pruning Parameter	Association type	Pearson Correlation	Significance
HD Dataset	CF	Negative	Strong	Significant
	MinNumObj	Negative	Strong	Insignificant
	NumFold	**Positive**	Weak	Insignificant
Thyroid Dataset	CF	Negative	Strong	Significant
	MinNumObj	Negative	Strong	Significant
	NumFold	Negative	Strong	Insignificant
Breast Cancer Dataset	CF	Negative	Strong	Significant
	MinNumObj	Negative	Weak	Insignificant
	NumFold	**Positive**	Weak	Insignificant

4.5 The Impact of Dataset Characteristics on Accuracy/cost of a Classifier

This section studies the impact of the dataset properties on the ATC and the accuracy of a DT classifier.

When all experiments were completed, a dataset of 75 rows and number of columns was obtained. The number of instances, number of classes, number of attributes, class distribution, ATC, and accuracy columns were used to determine the dataset characteristics impacting the classifier's accuracy and ATC using correlation and stepwize multiple regression analysis.

As the class distribution is a categorical variable and the correlation and multiple regression analysis requires all variable to be numerical. The class distribution has been coded into numerical in a process called "dummy coding". Value 0 indicates a balanced dataset and value 1 indicates an imbalanced dataset.

Before conducting the correlation and multiple regression analysis, the Min-Max normalization was also applied to normalize all variables as different scales were used for different datasets. SPSS was used for implementation.

The Min-Max normalization method is an efficient and commonly used for having a consistent data scale. As per the Min-Max normalization method, the normalized data (A') computed as follows (Patro & sahu, 2015):

$$A' = \frac{(A - Min\ value\ of\ A)}{Max\ value\ of\ A - Min\ value\ of\ A} \times (D - C) + C \qquad 4.1$$

where A is the data value to be normalized, C and D are the predefined scale, i.e. [0,1] in our case.

Table 4.6 shows the result of the Pearson correlarion analysis. Based on the Pearson values illustrated Table 4.6, all dataset characteristics affect the classifier accuracy and the ATC significantly. In other words, the number of instances, number of classes, number of attributes, and class distribution in a dataset makes a significant difference to the classifier's accuracy and ATC.

Among all dataset properties, the class distribution has the strongest association with the classifier's accuracy (Pearson value = 0.959) and ATC (Pearson value = 0.922).

The positive correlation values between the aforementioned characteristics and the classifier's accuracy indicates a positive (proportional) relationship with the classification accuracy, i.e. the higher number of instances, classes, and attributes, the higher classification accuracy. However, a negative relationship exists between the dataset characteristics and the ATC of a classifier.

Table 4.6: The Correlation Matrix of Dataset Properties and the Classifier's Accuracy/Cost

		Accuracy	ATC
No. Instances	Pearson Correlation	.713**	-.802**
	Sig. (2-tailed)	.000	.000
	N	75	75
No. Classes	Pearson Correlation	.642**	-.742**
	Sig. (2-tailed)	.000	.000
	N	75	75
No. Attribute	Pearson Correlation	.338**	-.467**
	Sig. (2-tailed)	.003	.000
	N	75	75
Class Distribution	Pearson Correlation	.959**	-.922**
	Sig. (2-tailed)	.000	.000
	N	75	75
**. Correlation is significant at the 0.01 level (2-tailed).			

A multiple regression analysis was also run on the experimental results to determine the percentage in which the dataset properties can affect the classifier's accuracy and the ATC. The datasets properties were used as the independent vaiables and the ATC and accuracy were used as the dependent variables.

The results illistrated in Table 4.7 show that 95.6% of variance in accuracy (denoted by the R square column) caused by the number of attributes and the class distribution properties. This also means that the number of attributes and the class distribution properties are the best subset of properties in explaining the dependent variable (i.e. accuracy).

Table 4.7: The Model Summery in Regression Analysis Using Accuracy

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Sig.
1	.959[a]	.920	.919	2.7763	.000[a]
2	.978[b]	.956	.955	2.0845	.000[b]
a. Predictors: (Constant), Class Distribution					
b. Predictors: (Constant), Class Distribution, No. Attributes					

In addition, the first row of Table 4.7 shows that the class distribution makes the largest contribution (R Square =0.920) in determining the accuracy of a classifier (i.e. 92%). The number of instrances makes less contribution in predicting the classifier's accuracy (R Square =0.956-0.920= 0.036, i.e. 3.6%).

When the stepwize multiple regression analysis was run on the experimental results to determine the impact of the dataset properties on the classifier's ATC, Table 4.8 was obtained.

Table 4.8 indicates that 95.6% of variance in ATC (R square =0.956) caused by the class distribution and the number of attributes properties.

The class distribution impact the ATC by 85.1% (R square= 0.851) while the impact of the number of attributes is just 10.5% (95.6%-85.1%).

Table 4.8: The Model Summery in Regression Analysis Using ATC

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate	Sig.
1	.922[a]	.851	.849	57.5922	.000[a]
2	.978[b]	.956	.955	31.5302	.000[b]

a. Predictors: (Constant), Class Distribution

b. Predictors: (Constant), Class Distribution, No. Attributes

The Sig. column in Table 4.7 and 4.8 indicates the significance of the set of independent variables in predicting the dependent variable. In this case, the class distribution and the number of attributes together are significantly related to the classifier's accuracy and ATC (0.000 implies that the significance level is below 0.0005).

By subtracting 95.6% from the 100%, we found that 4.4% of variance in the accuracy and the ATC is due to other factors including the number of instances, the number of classes, and maybe some other factors.

4.6 Use of Pareto Dominance Approach to Resolve the Accuracy/cost Trade-off

In this research, two (conflicting) objectives need to be optimized, the classifier's accuracy and the ATC of a DT classifier. The classifier's ATC must be minimized while the accuracy needs to be maximized.

The set of all solutions obtained in section 4.3 were used to approach the optimal balance between the classification accuracy and the ATC. The bi-objective optimization problem in this research can be stated using Pareto dominance approach as (Kamila et al., 2016):

$$find\ x = (x_1, x_2) \qquad 4.2$$

that minimizes $f_1(x)$, and maximizes $f_2(x)$

where x is a two dimentional vector of decision variables, $f_1(x)$ and $f_2(x)$ are the objective functions, i.e. the ATC and the accuracy functions.

Based on the definition of Pareto dominance in section 3.6.2, solution x^* dominates solution x *iff:* x^* outperforms x in one of the two objectives and performs the same or better than x in the other objective. Therefore, in Pareto dominance approach, often there is a set of non-dominated solutions also called "Pareto front" where no solution considered better than others.

Rarely, the multi-objective algorithm based on Pareto dominance approach returns a single solution that optimizes all objectives (i.e. dominates all other solutions), which is called "Pareto optimal solution". In this case, x^* is a Pareto optimal solution, if there is no x such that: $f_1(x) < f_1(x^*)$ or $f_2(x) > f_2(x^*)$, where f_1 and f_2 are the ATC and the accuracy functions, respectively.

To determine the set of non-dominated solutions from a set of solutions for multi-objective minimization problems, the "faster_pareto2" function in Matlab was employed. The set of solution vectors was passed to the function as a matrix, where each row in the matrix is one solution as shown in Figure 4.46.

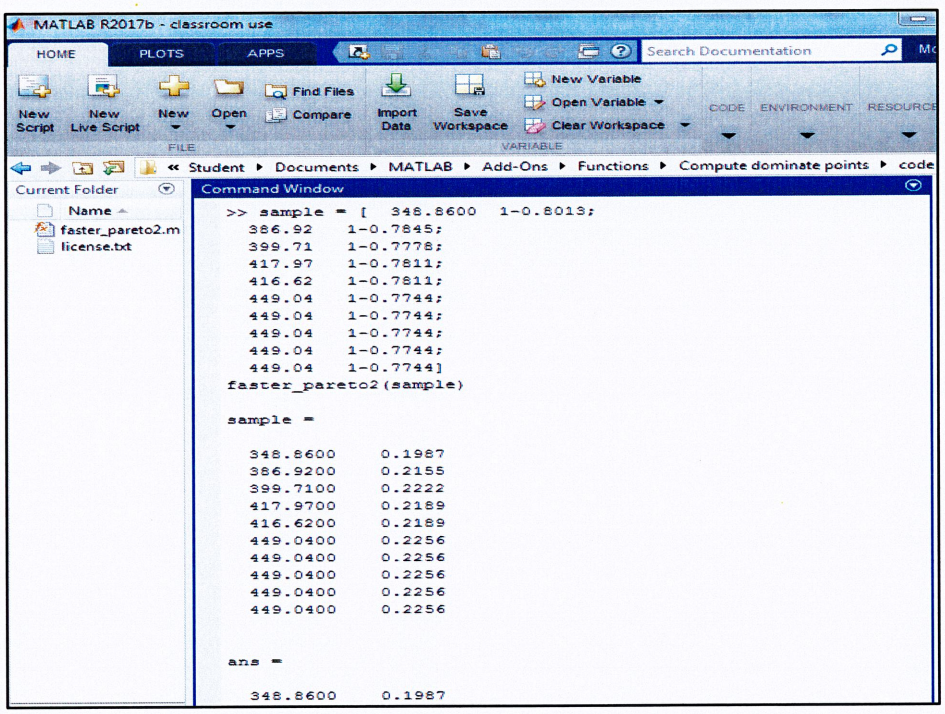


Figure 4.46: Use of faster_pareto2 Function in Matlab

Since our goal is to minimize the ATC and maximize the classifier's accuracy, the solution vectors were entered to the matrix in the following form (*total cost*, 1-*accuracy rate*), where (1-*accuracy rate*) is equivelent to the error rate. This will convert our problem into a minimization problem, i.e. to minimize both objectives, so that the "faster_pareto2" function can be used, as no function in Matlab found to minimize some objectives and maximize others. After getting the Pareto optimal solutions, it was converted back to (*total cost*, *accuracy*) form.

The set of non-dominated solutions (from Pareto sense) obtained from using different pruning parameters in all datasets are shown in Table 4.9. Note that the first number in the solutions sets represent the ATC and the second is the accuracy rate.

Table 4.9: Non-dominated Solutions

Pruning Parameters	HD Dataset	Thyroid Dataset	Breast Cancer Dataset
CF	(348.86,80.13)	(31.32, 99.18)	(153.33, 96.31) (149.13, 96.05)
MinNumObj	(313.57, 83.5)	(30.81, 99.07)	(154.33, 95.31) (131.65, 95.17) (98.91, 94)
NumFold	(348.25,78.11) (338.93, 77.44) (320.30, 75.42) (315.02, 74.75)	(29.67, 99.18)	(169.61, 94.88) (96.42, 94.29) (117.6, 94.44)

Table 4.9 shows that in some cases only one solution was found optimizing both objectives, obviously that would be the optimal solution. In some other cases, two or more non-dominated solutions were obtained. As stated in the literature, selecting the final solution from the set of Pareto non-dominated solutions depends on the decision maker and the domain it applied to since all non-dominated solutions are considered Pareto optimal solutions and it is difficult to find a single solution that gives the extreme value for all objective functions.

Since we are dealing with medical datasets, the accuracy rate objective will be given more privilage as it concerns the life of indivisuals, unless the reduction in accuracy is very slight (below 0.5%) compared to the reduction in ATC. Thus, among the Pareto optimal solutions illistrated in Table 4.9 (in case of having more than one optimal solution), the solution that was nominated in each dataset for each pruning method is displayed in Table 4.10. Note that only the top two solutions, in terms of acuuracy rate, were compared in Table 4.10.

Table 4.10: Selected Solutions from Pareto Front

Top Two Pareto Optimal Solutions (in terms of accuracy)	Percentage of Cost Reduction	Percentage of Accuracy Reduction	Selected Solution
(153.33, 96.31) (149.13, 96.05)	$\frac{153.33 - 149.13}{153.33} \times 100 = 2.74\%$	$\frac{96.31 - 96.05}{96.31} \times 100 = 0.27\%$	(149.13, 96.05)
(154.33, 95.31) (131.65, 95.17)	$\frac{154.33 - 131.65}{154.33} \times 100 = 14.7\%$	$\frac{95.31 - 95.17}{95.31} \times 100 = 0.14\ \%$	(131.65, 95.17)
(169.61, 94.88) (117.6, 94.44)	$\frac{169.61 - 117.6}{169.61} \times 100 = 30.66\%$	$\frac{94.88 - 94.44}{94.88} \times 100 = 0.46\%$	(117.6, 94.44)
(348.25,78.11) (338.93, 77.44)	$\frac{348.25 - 338.93}{348.25} \times 100 = 2.68\%$	$\frac{78.11 - 77.44}{78.11} \times 100 = 0.86\%$	(348.25,78.11)

Now, the Pareto optimal solutions for all datasets with the corresponding pruning parameters value are displayed in Table 4.11.

Table 4.11 also shows the percentage of reduction in ATC in Pareto optimal solutions compared to the highest obtained values for the ATCs. According to Table 4.11, the reduction in cost reached 30.9% with only 0.46% reduction in accuracy (from Table 4.10), compared to the highest accuracy obtained (94.88%).

Table 4.11: Percentage of ATC Reduction Compared to Highest ATC

Dataset	Pruning Parameters values	Pareto Optimal Solutions	Percentage of Cost Reduction Compared to Highest ATC
HD Dataset	CF = 0.1	(348.86, 80.13)	$\frac{449.04 - 348.86}{449.04} \times 100 = 22.31\%$
	MinNumObj = 15	(313.57, 83.5)	$\frac{352.92 - 313.57}{352.92} \times 100 = 11.15\%$
	NumFold = 7	(348.25, 78.11)	$\frac{387.47 - 348.25}{387.47} \times 100 = 10.12\%$
Thyroid Dataset	CF = 0.2	(31.32, 99.18)	$\frac{31.64 - 31.32}{31.64} \times 100 = 1.01\%$
	MinNumObj = 5	(30.81, 99.07)	$\frac{33.57 - 30.81}{33.57} \times 100 = 8.22\%$
	NumFold = 5	(29.67, 99.18)	$\frac{31.91 - 29.67}{31.91} \times 100 = 7.02\%$
Breast Cancer Dataset	CF = 0.2	(149.13, 96.05)	$\frac{173.71 - 149.13}{173.71} \times 100 = 14.15\%$
	MinNumObj = 10	(131.65, 95.17)	$\frac{154.33 - 131.65}{154.33} \times 100 = 14.7\%$
	NumFold = 13	(117.6, 94.44)	$\frac{170.19 - 117.6}{170.19} \times 100 = 30.90\%$

From Table 4.11, we can observe that using the MinNumObj pruning parameter in HD dataset gave the best classifier's performance in terms of the classification accuracy and ATC compared to the CF and NumFold pruning parameter. However, it also shows that the NumFold gave the best results with the thyroid disease dataset. With the breast cancer dataset, the CF parameter gave the best accuracy rate and the NumFold gave the lowest ATC. Thus, we can conclude that no single pruning method is applicable to all datasets or can always outperform other pruning methods.

Finally, Table 4.12 shows the percentage of reduction in the ATC and the increase/decrease in accuracy compared to the standard C4.5 algorithm with its default settings. The reduction in ATC reached 21.14% (obtained by subtracting the ATC of the selected solution from the ATC of the standard C4.5 and divinding the result by the standard ATC) with only 1.6% reduction in the classification accuracy.

According to Table 4.12, the reduction in ATC in HD dataset reached 19.79% with an increase in accuracy by 6.85% compared to DTs built using C4.5 algorithm with the deafult settings. Table 4.12 also shows that in many other cases, our results decreased the ATC and increased (or keep the same) the classification accuracy concurrently compared to the standarad C4.5 algorithm.

Table 4.12 also shows that in thyroid disease dataset, the reduction in ATC reached 5.27% with the same classification accuracy as in C4.5 with its default pruning parameters settings.

Table 4.12: Percentage of ATC and Accuracy Reduction Compared to the Standard C4.5

Dataset/ Default Accuracy and ATC in C4.5	Pruning Parameters Values	Pareto Optimal Solutions	Percentage of Cost Reduction	Percentage of Accuracy Increase/ Reduction
HD Dataset (390.96, 77.87)	CF	(348.86, 80.13)	$\frac{390.96 - 348.86}{390.96} \times 100 = 10.77\%$	$\frac{80.13 - 77.78}{80.13} \times 100 = 2.93\%$ **Increase**
	MinNumObj	(313.57, 83.5)	$\frac{390.96 - 313.57}{390.96} \times 100 = 19.79\%$	$\frac{83.5 - 77.78}{83.5} \times 100 = 6.85\%$ **Increase**
	NumFold	(348.25, 78.11)	$\frac{390.96 - 348.25}{390.96} \times 100 = 10.92\%$	$\frac{78.11 - 77.78}{78.11} \times 100 = 0.0.31\%$ **Increase**
Thyroid Dataset (31.32, 99.18)	CF= 0.2	(31.32, 99.18)	$\frac{31.32 - 31.32}{31.32} \times 100 = 0\ \%$	$\frac{99.18 - 99.18}{99.18} \times 100 = 0\ \%$
	MinNumObj	(30.81, 99.07)	$\frac{31.32 - 30.81}{31.32} \times 100 = 1.63\%$	$\frac{99.18 - 99.07}{99.18} \times 100 = 0.11\ \%$ **Decrease**
	NumFold	(29.67, 99.18)	$\frac{31.32 - 29.67}{31.32} \times 100 = 5.27\%$	$\frac{99.18 - 99.18}{99.18} \times 100 = 0\ \%$
Breast Cancer Dataset (149.13, 96.05)	CF	(149.13, 96.05)	$\frac{149.13 - 149.13}{149.13} \times 100 = 0\%$	$\frac{96.05 - 96.05}{96.05} \times 100 = 0\ \%$
	MinNumObj	(131.65, 95.17)	$\frac{149.13 - 131.65}{149.13} \times 100 = 11.72\%$	$\frac{96.05 - 95.17}{96.05} \times 100 = 0.92\ \%$ **Decrease**
	NumFold	(117.6, 94.44)	$\frac{149.13 - 117.6}{149.13} \times 100 = 21.14\%$	$\frac{96.05 - 94.44}{96.05} \times 100 = 1.6\ \%$ **Decrease**

The set of non-dominated (Pareto optmimal) solutions in HD dataset when the CF, MinNumObj, and NumFold pruning parameters were used are displayed in Figure 4.47, 4.48, and 4.49, respectively. The selected solutions are circled when more than one non-dominated solutions exist.

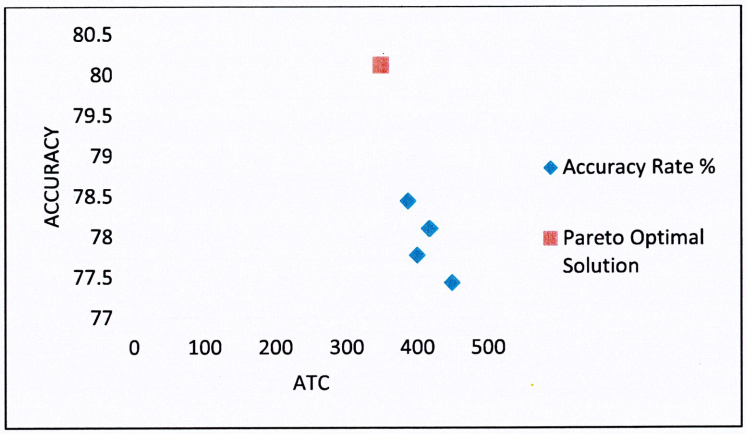


Figure 4.47: Non-dominated Solutions in HD when using the CF

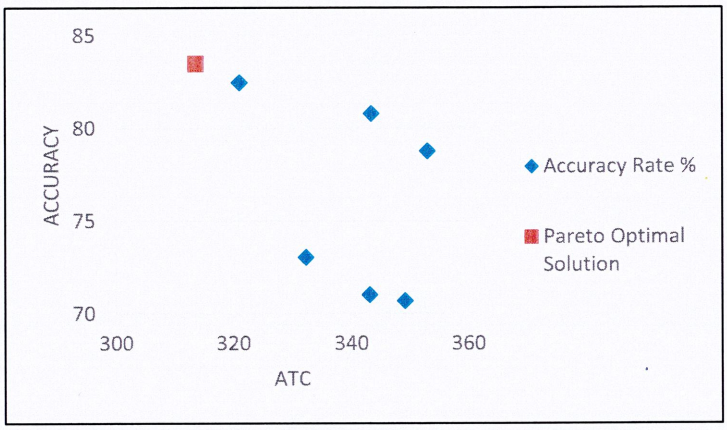


Figure 4.48: Non-dominated Solutions in HD when using the MinNumObj

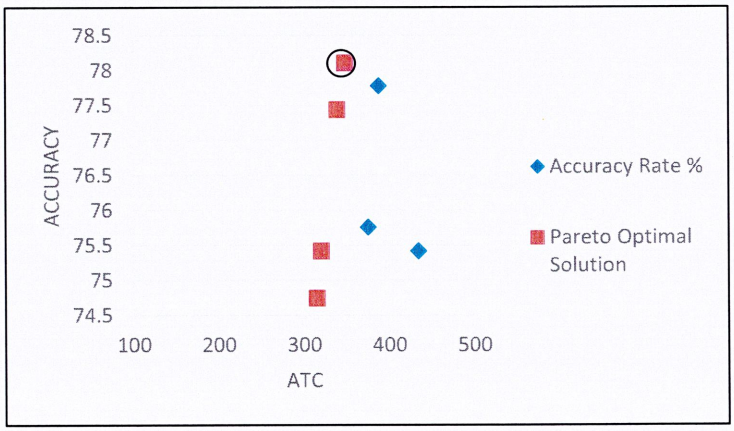


Figure 4.49: Non-dominated Solutions in HD when using the NumFold

The set of non-dominated solutions when thyroid diseaes dataset were used with the CF, MinNumObj, and NumFold pruning parameters are displayed in Figure 4.50, 4.51, and 4.52, respectively.

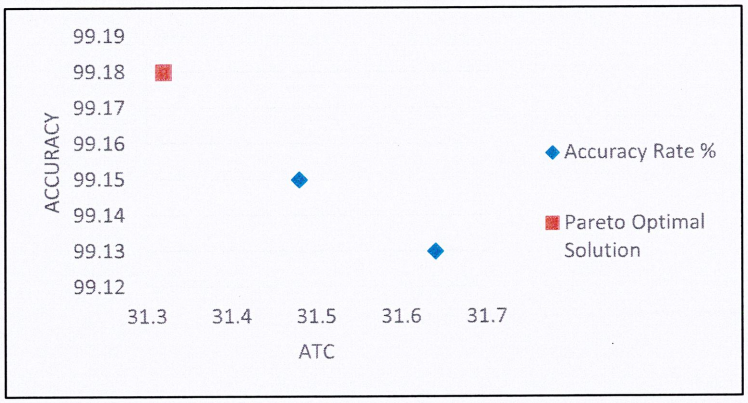


Figure 4.50: Non-dominated Solutions in Thyroid when using the CF

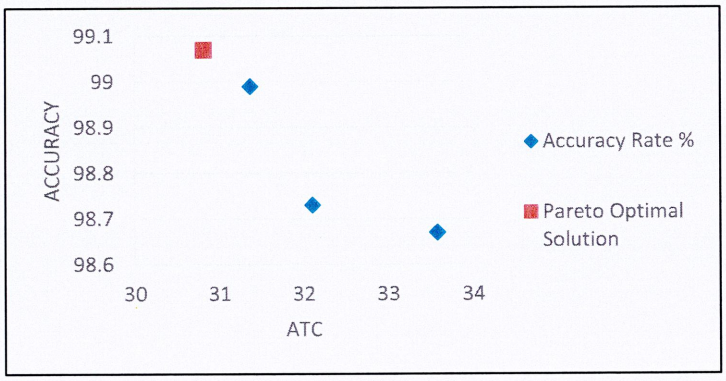


Figure 4.51: Non-dominated Solutions in Thyroid when using the MinNumObj

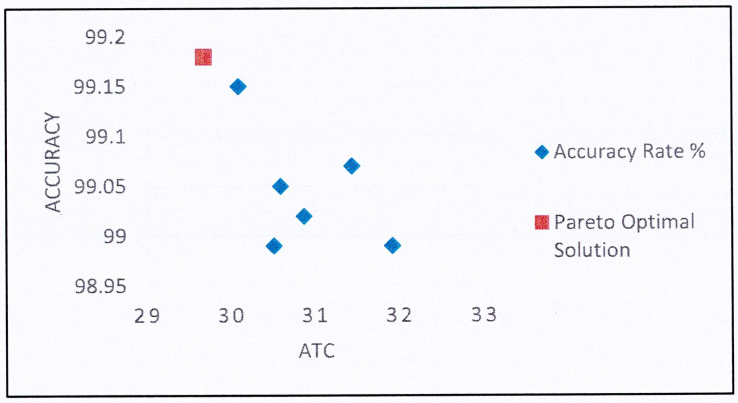


Figure 4.52: Non-dominated Solutions in Thyroid when using the NumFold

The set of non-dominated solutions in breast cancer dataset when the CF, MinNumObj, and NumFold pruning parameters were used are displayed in Figure 4.53, 4.54. and 4.55, respectively. Among the set of non-dominated solutions displayed, only the selected solutions are circled.

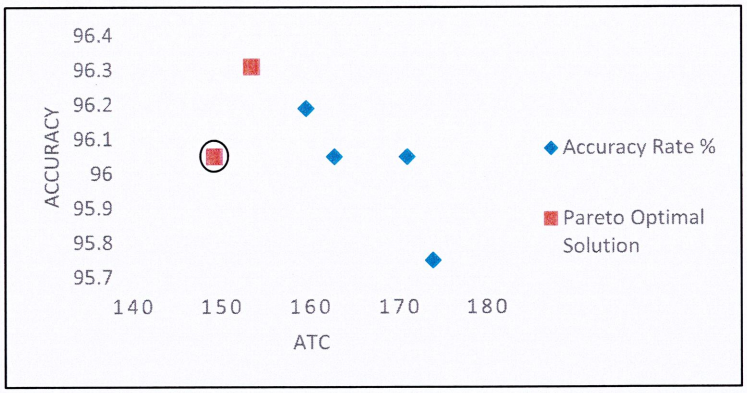


Figure 4.53: Non-dominated Solutions in Breast Cancer when using the CF

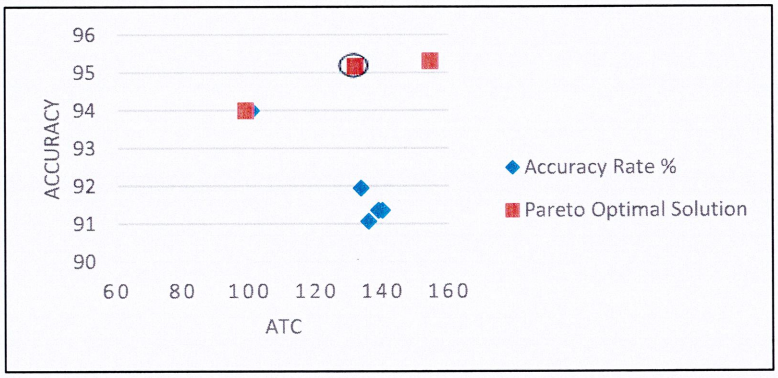


Figure 4.54: Non-dominated Solutions in Breast Cancer when using the MinNumObj

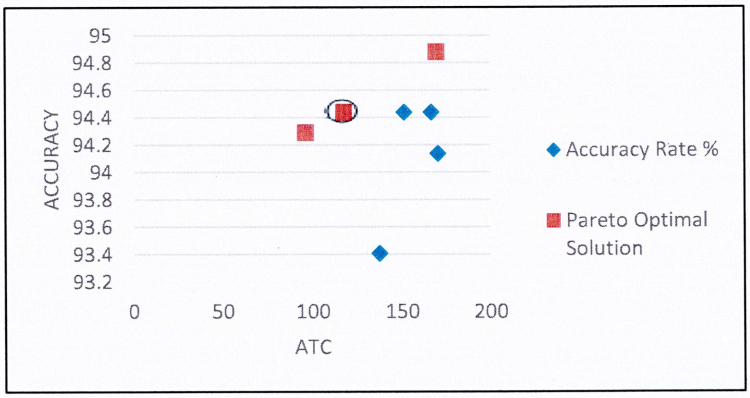


Figure 4.55: Non-dominated Solutions in Breast Cancer when using the NumFold

4.7 Results Comparison

This section compares the results of this research with other researcher's results considering the two main classification cost types: the MC and the TC.

From the literature, some researchers were successfully able to reduce the ATC more the result of this research (i.e. 21.14%) including Ling, et al. (2004), Deng and Song (2014), and Zhao et al. (2015).

The drop in the ATC reached 80% by Ling, et al. (2004) and reached 25% by Deng and Song (2014) compared to the standard C4.5 algorithm. In addition, Zhao et al. (2015) also successfully reduced the total classification cost by 42.6% and 55.6% compared to CS-C4.5 and CSGainRatio algorithms, respectively. However, the resulting accuracy rates were not mentioned in those researches.

Moreover, Lu, et al. (2017) also successfully reduced the MC up to 30%, but the drop in accuracy was 10.2% compared to Rotation Forest (RoF) algorithm. Note that they did not mention the percentage of reduction in total classification cost in their research.

Thus, among all reviewed research papers in the literature considering both classification cost types, our results outperform their results as the drop in the accuracy was only 1.6% and the drop in the cost reached 21.4% compared to the standard C4.5 algorithm. Table 4.13 presents a comparison between this research results and other researchers' results.

Table 4.13: Research Results Comparison

	Reduction in Total Cost	Reduction in Accuracy Rate	Dataset Used
Ling, et al. (2004)	Lowered the ATC up to 80% compared to C4.5 algorithm.	The accuracy rate was not measured.	Ecoli, Breast cancer, heart disease, thyroid disease, and Australia datasets.
Deng and Song (2014)	Lowered the total classification up to 25% compared to C4.5 algorithm.	The accuracy rate was not measured.	Credit dataset.
Zhao, et al. (2015)	Lowered the total classification cost by 9.9%, 42.6%, and 55.6% compared to post pruning, CS-C4.5, and CSGainRatio algorithms, respectively.	The accuracy rate was not measured or mensioned in their research.	Biodeg, Breast cancer, Clean, Credit, Diabetes, Eeg, Wdbc, and Wilt datasets.
Lu, et al. (2017)	The reduction in MC reached 30%. Although, they embedded both cost types in the new DT classifier, they only mentioned the percentage of reduction in average MC.	The drop in accuracy also reached 10.2% compared to Rotation Forest (RoF) algorithm.	Lung and ovarian datasets.
This research	The reduction in the ATC reached 21.14%, 19.79%, and 5.27% in breast cancer, HD, and thyroid disease datasets, respectively, when compared to C4.5 algorithm with its default settings.	The drop in accuracy rate was only 1.6% in breast cancer dataset. It was increased by 6.68% in HD dataset and remain the same in thyroid disease dataset when compared to C4.5 algorithm with its default pruning parameters' settings.	Heart disease, thyroid disease, and Breast cancer datasets.

4.8 Conclusion

This chapter started with the experimental framework followed in condutcting this research. Eight stages were involved. All experiments conducted on three different datasets, their setup, and their output from each stage were discussed in details.

The experimental results were also demonstrated, analyzed, and compared. The results showed that using the CF parameter for DT pruning results in having a negative relationship between the ATC and the classification accuracy in all datasets. The same results were obtained when the MinNumObj parameter was used for DT pruning. When reduced

error pruning method was used, the resulting relationship was almost proportional in HD and breast cancer datasets. However, it was almost proportional inverse in the thyroid disease dataset with a very slight difference in accuracy.

The resulting association between the ATC and the accuracy was verified in all datasets by calculating the Pearson correlation coefficients. The association found to be negative with all pruning parameters settings in all datasets, except for the HD and breast cancer datasets when tuning the NumFold pruning parameter (two cases out of nine), refer to Table 4.13. However, the two positive associations found was very weak and insignificant when the Pearson correlation coefficients were calculated.

In addition, the impact of the dataset characteristics on the DT's ATC and accuracy was studied, including the number of instances, number of classes, number of attributes, and the class distribution properties. A Pearson correlation analysis and a stepwize multiple regression analysis were used.

The results of the stepwize multiple regression analysis showed that 95.6% of variance in the ATC and the accuracy caused by the number of attributes and the class distribution properties. In fact, the class distribution makes the largest contribution in determining the accuracy and ATC values of a classifier based on the obtained R square values. The class distributing property affects the accuracy by 92% and affects the ATC by 85.1%, while the number of attributes property affects the accuracy and ATC by 3.6% and 10.5%, respectively.

The Pearson correlation analysis results showed that the number of instances and the number of classes properties are also impacting the classifier accuracy and the ATC.

The Pareto dominance approach was used to get the set of all non-dominated solutions from the whole set of solutions obtained from using different pruning methods on different datasets. Among the set of non-dominated solutions, one solution has been selected based on its contribution to minimize the ATC without sacrifying the classification accuracy since all datasets are medical. The percentage of ATC reduction reached around 30.9% compared to the highest ATC obtained, with only 0.46% reduction in accuracy.

When the ATC and the accuracy of selected Pareto optimal solutions were compared to that of DTs constructed using C4.5 algorithm (with its default pruning settings), the reduction in ATC in our Pareto optimal solutions reached 21.14%, 19.79%, and 5.27% in breast cancer, HD, and thyroid disease datasets, respectively. The reduction in the resulting classification accuracy was only 1.6% in breast cancer dataset. In HD dataset, the resulting classification accuracy was increased by 6.68% and the same accuracy rate was obtained in thyroid disease dataset compared to the standard C4.5 algorithm.

Finally, the results obtained from this research were compared with other researchers results. Although the percentage of ATC reduction by some other reserchers exceeded the percentage of ATC reduction obtained in this research, they only focused on ATC reduction and the resulting accuracy rate was not mentioned in their researches.

Other researchers almost got the same percentage of ATC reduction as of this research, but the reduction in the classification accuracy exceeded the one obtained in this research. Thus, the results of this research considered outperforming other researchers results.

The following chapter gives a conclusion and some recommendations to this research.

Chapter Five

Conclusion and Recommendations

5.0 Overview

This chapter will summarize the approach followed in conducting this research. Section 2 will discuss all the experimental results and will revise all objectives to ensure objectives achievement.

Section 3 will compare this research results with the results of other researchers. Section 4 and 5 will discuss the research contributions and some future directions.

5.1 Summary of the Approach Followed

This section summarizes the approach followed to conduct this research. Six stages were involved.

The first stage in this research is the literature review. During this stage, the problem of cost-sensitive learning was studied, different classification cost types were defined, the method for calculating the Average Total Cost (ATC) was studied. In addition, the relationships between some of the classification cost types and the classification accuracy were discussed and some of the researchers' attempts to minimize the Test Cost (TC), misclassification Cost (MC), or the ATC without reducing the classification accuracy were presented.

In the second stage, three preprocessed and cleaned datasets were selected from UCI machine learning repository. The selected datasets are the heart disease (HD), thyroid disease, and breast cancer datasets.

The next stage was to clean datasets from missing values by removing all records in datasets containing missing attribute values. In stage 4, the experiments were conducted. J48 algorithm was used for the DT construction.

To balance between the ATC and the accuracy, the behavior of DTs was altered by pruning. Three of the J48's (the Java implementation of C4.5 algorithm in Weka) pruning parameters, namely the Minimum Number of Objects (MinNumObj), the Confidence Factor (CF), and the Number of Folds (NumFold), have been adjusted on the three real medical datasets. The parameters have been utilized to control the level of the DT pruning. The ATC and the classifier's accuracy have been calculated and recorded at different pruning levels.

During the fifth stage, the experimental results were used to model the relationship between the ATC and the accuracy. The model then utilized to determine the required level of DT pruning that gives satisfactory accuracy/cost values using the Pareto dominance approach. The Pareto dominance approach was selected to optimize the two conflicting objectives, where the classifier's accuracy needs to be maximized and the ATC needs to be minimized. For this purpose, Matlab was employed. It returns the set of all non-dominated solutions for each pruning method in each dataset.

The final stage involves data analysis, comparison, and discussion. The relationship between the ATC and the accuracy was validated and the significance of the relationship was determined by calculating the Pearson correlation coefficients. The impact of the dataset properties on the classifier's accuracy and ATC was also determined by conducting Pearson

correlation and multiple regression analysis. Moreover, the optimized accuracy/cost values were compared to those of other researchers.

5.2 Summary of Results

This section summarizes the experimental results of this research.

One of this research objectives was to study the behavior of the DT using different pruning methods. The experimental results showed that when the MinNumObj and the NumFolds pruning parameters were used in all datasets, the accuracy was increasing as the tree size increases. However, when the CF parameter was used the opposite occurs. The reason is that increasing the CF will not only increase the tree size, but will also increase the percentage of allowed error rate inherent in the DT during the DT pruning.

The empirical results showed that a proportional inverse relationship exists between the average total MC and the accuracy in all datasets regardless the pruning method used. However, the relationship between the classifier's average total TC and the accuracy was proportional in all datasets as expected from literature, except when the CF parameter was used. The reason behind this unexpected result is that tuning the CF parameter makes the relationship between the tree size and the accuracy proportional inverse as mentioned in the previous paragraph. Moreover, as the tree size proportionally related to the average total TC (from literature), the resulting relationship between the average total TC and the accuracy becomes proportional inverse.

The resulting relationship between the ATC -which is the total of the average total MC and the average total TC- and the accuracy was modeled

graphically using scattered plot and numerically using correlation analysis to achieve the first objective of this research.

The graphical display illustrated a proportional inverse relationship between the ATC and the accuracy in all datasets when the CF and the MinNumObj pruning parameters were used. The average total MC dominates the average total TC and impacted the ATC of classification more. It shapes the relationship towards the inverse direction.

When the NumFold parameter was used with the HD and breast cancer datasets, the results showed an almost proportional relationship between the ATC and the accuracy. However, the Pearson correlation coefficient indicated that this positive correlation is very weak and insignificant (the Pearson correlation value was below 0.3).

The results were verified by computing the Pearson correlation coefficients. The Pearson correlation indicated an inverse relationship between the ATC and the accuracy in all datasets with all pruning methods except for the NumFold parameter as mentioned above.

Therefore, the overall relationship between the ATC and the accuracy rate can be modeled as illustrated in Figure 5.1. This result was expected as the MCs assigned in the classification cost matrix was high (FP= 600 and FN=1000) as it is supposed to be due to the sensitivity of data and as recommended by researchers from the Heart-Failure Research Group in the local medical school (Weiss et al., 2013). Thus, it affected the ATC more compared to the average total TC and makes its relationship with the accuracy rate becomes proportional inverse.

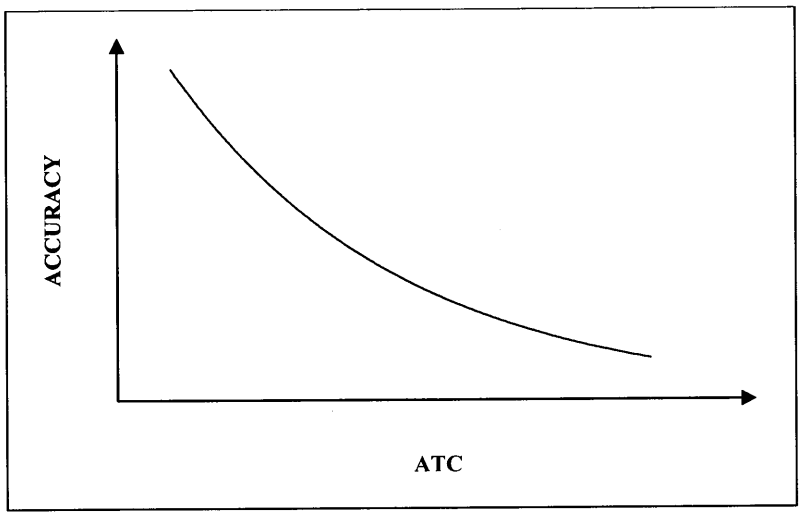


Figure 5.1: Modeling the Relationship between the ATC and the Accuracy

It is important to notice here that each dataset has its own characteristics, hence different datasets logically give different results. Additionally, the three selected datasets are not representing all possible datasets. However, the similarity in the experimental results on the three datasets made us conclude that the accuracy is decreasing (monotonically) as the ATC increases in the selected datasets and hence the threat of the representativeness of datasets is minimized.

Part of the first objective was to determine the required level of pruning that gives satisfactory accuracy/ATC values. To achieve this objective, one of the optimization approaches has been utilized, "the Pareto dominance approach". It was selected to optimize the two conflicting objectives, where the classifier's accuracy needs to be maximized and the ATC needs to be minimized. For this purpose, a function in Matlab was employed. It returns the set of all non-dominated solutions for each pruning method in each dataset.

Among the set of all non-dominated solutions, only one solution was nominated to be the optimal solution. The nomination was based on the

solution with the highest accuracy since all datasets are medical, unless the reduction in accuracy found very small (0.5% or below) compared to the reduction in the ATC.

The second part of the first objective was to determine the percentage of cost savings that gives satisfactory accuracy. The results showed that the reduction in the ATC reached 30.9% compared to the highest ATC obtained through experimentation with only 0.46% reduction in the classification accuracy.

When the selected Pareto optimal solutions were compared to the standard DTs constructed using the C4.5 algorithm (with default pruning settings), the reduction in ATC in our Pareto optimal solution reached 21.14%, 19.79%, and 5.27% in breast cancer, HD, and thyroid disease datasets, respectively. The reduction in the classification accuracy was only 1.6% in breast cancer dataset. In HD dataset, the resulting classification accuracy was increased by 6.68% and the same accuracy rate was obtained in thyroid disease dataset compared to the standard C4.5 algorithm.

The second research objective was to study the impact of the dataset characteristics on the accuracy and the ATC of a DT. To achieve this objective, a Pearson correlation analysis and a stepwise multiple regression analysis were conducted to determine the impact of the dataset size, the number of classes, the number of attributes, and the class distribution properties on the ATC and the accuracy of a DT classifier.

The results of the Pearson correlation analysis showed that all the aforementioned properties are impacting the ATC and the accuracy of the DT classifier significantly. Among all dataset properties, the class

distribution has the strongest association with the classifier's accuracy (Pearson value= 0.959) and the ATC (Pearson value= 0.922).

In addition, the results of the stepwise multiple regression analysis showed that 95.6% of the variance in the classifier's accuracy and the ATC caused by the number of attributes and the class distribution properties. In other words, they represent the best subset of the dataset properties for explaining the variance in the classifier's accuracy and the ATC. In fact, the class distribution contributes by 92% and 85.1% in explaining the accuracy and the ATC of a classifier, respectively. On the other hand, the number of attributes property explains 3.6% and 10.5% of the variability in the accuracy and the ATC of a classifier, respectively. Readers may refer to section 4.5 for more details.

It has been also noticed that although the HD dataset was the only balanced dataset, it was the hardest dataset to predict as it gives the lowest accuracy rate compared to the other two datasets. However, the overall relationship between the ATC and the accuracy was almost the same in all datasets.

Other factors such as the cost assigned to each attribute in the TC vector and the MCs used in the classification cost matrix were also found affecting the resulting relationship between the ATC and accuracy pairs.

According to the research results presented in this section, the answers to this research questions can be summarized as illustrated in Table 5.1.

Table 5.1: Answers to the Research Questions

Research Question	Answer
Can the relationship between the ATC and classification accuracy in DT classifier be modeled?	Yes, the scattered plot graphs and the computed Pearson correlation coefficients showed that the ATC is negatively related to the accuracy in a DT classifier on the selected datasets.
What is the optimum balance between the classification accuracy and the ATC?	Optimal balance relies on the dataset itself. For example, the optimum balance between the ATC and accuracy was obtained when the "Minimum number of object" pruning method was used in HD dataset and the minimum number of instances (i.e. MinNumObj parameter) was set to 15. While in the thyroid disease dataset, the optimum ATC/accuracy results was obtained when the reduced error pruning method was used and the NumFold parameter was set to 5. The reduced error pruning method also gave the best ATC/accuracy results in breast cancer dataset when the NumFold parameter was set to 13. Further explanation was given earlier in this section.
What is the percentage of saving in ATC that gives satisfactory accuracy?	There is no uniform saving in ATC that gives satisfactory accuracy for all datasets. For example, the percentage of ATC saving reached 19.79% in HD dataset with an increase in the classifier accuracy by 6.68% compared to the standard C4.5 algorithm with default the pruning parameters setting. In thyroid disease dataset the reduction in ATC reached 5.27% with the same accuracy rate as the standard C4.5 algorithm. The percentage of reduction in ATC reached 21.14% with only 1.6% reduction in accuracy compared to the standard C4.5 algorithm with the default pruning settings.
Using different pruning methods, are the accuracy and the cost of a classifier significantly affected by the dataset properties?	The dataset size, the number of classes, the number of attributes, and the class distribution of a dataset all are significantly affecting the ATC and the accuracy of a DT classifier based on the resulting correlation matrices. The class distribution has the strongest association with the classifier's accuracy and ATC. The multiple regression results showed that the class distribution affect the classifier's accuracy by 92% and the ATC by 85.1%.

	The number of attributes affect the classifier's accuracy by 3.6% and the ATC by 10.5%. The dataset size and the number of classes affect the ATC and accuracy by 4.4%.

5.3 Comparison with Other Researchers Results

This section briefly compares the results of this research and other researcher's results who aimed to minimize the total of MC and TC types.

Among the reviewed research papers in the literature considering both classification cost types, it has been concluded that this research results outperform other researcher's results as some researchers were successfully exceeded the percentage of cost reduction obtained in this research (21.14% compared to the standard C4.5 algorithm) including Ling, et al. (2004), Deng and Song (2014), and Zhao, et al. (2015). However, those researches did not mention the resulting accuracy rates.

In addition, Lu, et al. (2017) almost got almost the same percentage of cost reduction obtained in this research, but the reduction in accuracy exceeded the accuracy reduction obtained in this research.

Note that the decrease in accuracy was only 1.6% in this research when compared to the standard C4.5 algorithm with the default pruning parameters' settings and 0.46% when compared to C4.5 algorithm with adjusted pruning parameters' settings. On the other hand, reduction in accuracy reached 10.2% in Lu, et al. (2017) research compared to Rotation Forest (RoF) algorithm. For more details, readers may refer to section 4.7.

5.4 The Thesis Contribution

This section summarizes the main contributions of this research.

First of all, up to the researcher knowledge, this research is the first of its type to model the total of the MC and the TC against the classification accuracy. Other researchers have successfully modeled the MC or the TC (but not both) against the classification accuracy. However, they considered the same cost for all attributes and misclassifications in their models.

Furthermore, this research presented a comparison among different post-pruning and pre-pruning methods used in J48 classifier.

Additionally, the research results reveal that the optimal solution depends on the case, i.e. there is no single optimal setting for the CF, MinNumObj, and NumFolds post-pruning and pre-pruning parameters that can fit all datasets. In some datasets, pruning DTs using a particular setting for the CF gives the best accuracy/ATC results, while in others using the MinNumObj or NumFolds parameters may give better results. Therefore, data miners should determine the optimal pruning parameter using the Pareto dominance approach.

Moreover, the research was successful in balancing between the accuracy and the ATC of a classifier by reducing the ATC by 21.14% compared to the standard C4.5 algorithm, with only a reduction in the classification accuracy by 1.6%. Thus, it would be useful in many domains, especially in the medical diagnosis domain for performing diagnostic tasks, where both the classification accuracy and the ATC are of high importance.

Last but not least, unlike the previous researches who resolved the trade-off between the classification accuracy and the ATC through updating the attribute selection criterion of the DT algorithm, this research resolved the trade-off through adjusting the pruning parameters' settings and then selecting the optimal parameters' setting using the Pareto dominance approach. This way of resolving the problem reveals its effectiveness when compared to other researchers' results.

5.5 Future Directions

This section presents different aspects of this research that can be improved in the future.

First of all, researchers are recommended to consider group costs in calculating the total TC of a particular case. In this method, some attributes may share a common cost and performing them together or in a particular order leads to having a discount on the total TC of a case.

In addition, the computational time required to build the DT classifier can also be computed and considered in the selection of the final tree, and thus, different cost types can be involved in different cost scales.

This research also can be conducted on other classification algorithms (e.g. Random Forest or Naïve Bayes, etc.) to investigate the proper balance between the classification accuracy and the ATC.

Last but not least, other factors affecting the accuracy and ATC of a classifier, like the complexity of the classification algorithm, can be explored in future and the results can be verified using datasets from other domains rather than the medical domain.

References

Ahmadi, A., Dehzangi, O., & Jafari, R. (2012). Brain-computer Interface Signal Processing Algorithms: A Computational Cost vs. Accuracy Analysis for Wearable Computers. *9th International Conference on Wearable and Implantable Body Sensor Networks* (pp. 40-45). London: IEEE.

AlMana, A., & Aksoy, M. (2014). An Overview of Inductive Learning Algorithms. *International Journal of Computer Applications, 88*(4), 20-28.

Asif, R., Merceron, A., Ali, S., & Haider, N. (2017). Analyzing Undergraduate Students' Performance using Educational Data Mining. *Computers & Education, 113*, 177-194.

Bakhshinategh, B., Zaiane, O., ElAtia, S., & Ipperciel, D. (2017). Educational Data Mining Applications and Tasks: A Survey of the Last 10 Years. *Education and Information Technologies, 23*(1), 537-553.

Berry, M., & Browne, M. (2006). *Lecture Notes in Data Mining.* Hackensack: World Scientific.

Bhowmick, S., Toth, B., & Raghavan, P. (2009). Towards Low-cost, High-accuracy Classifiers for Linear Solver Selection. *International Conference on Computational Science, LNCS 5544*, (pp. 463-472).

Bohanec, M., & Bratko, I. (1994). Trading Accuracy for Simplicity in Decision Trees. *Machine Learning, 15*(3), 223-250.

Bradford, J., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. (1998). *Pruning Decision Trees with Misclassification Costs.* West Lafayebte: Purdue University.

Cestnik, B., & Bratko, I. (1991). On Estimating Probabilities in Tree Pruning. In Kodratoff, Y., *Machine Learning. EWSL 1991. Lecture Notes in Computer Science* (Vol. 482, pp. 138-150). Berlin, Heidelberg: Springer.

Chen, Y., Wu, C., & Tang, K. (2016). Time-constrained Cost-sensitive Decision Tree Induction. *Information Sciences, 354*, 140-152.

Cheng, Y., Chen, K., Sun, H., Zhang, Y., & Tao, F. (2018). Data and Knowledge Mining with Big Data Towards Smart Production. *Journal of Industrial Information Integration, 9*, 1-13.

Davis, J., Ha, J., Rossbach, C., Ramadan, H., & Witchel, E. (2006). *Cost-Sensitive Decision Tree Learning for Forensic Classification.* The University of Texas at Austin.

Deng, L., & Song, J. (2014). Decision Tree Classification Algorithm based on Cost and Benefit Dual-sensitive. *IEEE International Conference on Electro/Information Technology*, (pp. 320 - 323).

Dogan, N., & Tanrikulu, Z. (2012). A Comparative Analysis of Classification Algorithms in Data Mining for Accuracy, Speed and Robustness. *Information Technology and Management, 14*(2), 105-124.

Du, H. (2010). *Data Mining Techniques and Applications.* United Kingdom: Cengage Learning.

Du, J., Cai, Z., & Chen, Y. (2007). A Sorting Based Algorithm for Finding Non-dominated Set in Multi-objective Optimization. *3rd International Conference on Natural Computation (ICNC 2007)* (pp. 1-5). Haikou, China: IEEE.

Farid, D., Zhang, L., Rahman, C., Hossain, M., & Strachan, R. (2014). Hybrid Decision Tree and Naïve Bayes Classifiers for Multi-class Classification Tasks. *Expert Systems with Applications, 41*(4), 1937-1946.

Figueiredo, M., Esteves, L., Neves, J., & Vicente, H. (2016). A Data Mining Approach to Study the Impact of the Methodology Followed in Chemistry Lab Classes on the Weight Attributed by the Students to the Lab Work on Learning and Motivation. *Chemistry Education Research and Practice, 17*(1), 156-171.

Freitas, A. (2004). A Critical Review of Multi-objective Optimization in Data Mining: A Position Paper. *ACM SIGKDD Explorations Newsletter, 6*(2), 77-86.

Freitas, A., Costa-Pereira, A., & Brazdil, P. (2007). Cost-sensitive Decision Trees Applied to Medical Data. In Song, I., Eder, J., & Nguyen, T., *Data Warehousing and Knowledge Discovery. DaWaK*

2007. Lecture Notes in Computer Science (Vol. 4654, pp. 303-312). Berlin, Heidelberg: Springer.

Gerber, S., & Finn, K. (2005). *Using SPSS for Windows.* New York: Springer.

He, Q., Zhu, X., Li, D., Wang, S., Shen, J., & Yang, Y. (2017). Cost-effective Big Data Mining in the Cloud: A Case Study with K-means. *IEEE 10th International Conference on Cloud Computing (CLOUD)*, (pp. 74-81).

Hodge, V., & Austin, J. (2004). A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review, 22*(2), 85-126.

Hussain, S. (2017). Survey on Current Trends and Techniques of Data Mining Research. *London Journal of Research in Computer Science and Technology, 17*(1), 7-16.

Jensen, U., Kugler, P., Ring, M., & Eskofier, B. (2015). Approaching the Accuracy–cost Conflict in Embedded Classification System Design. *Pattern Analysis and Applications, 19*(3), 839-855.

Jiang, L., Li, C., & Wang, S. (2014). Cost-sensitive Bayesian Network Classifiers. *Pattern Recognition Letters, 45*, 211-216.

Jin, Y., & Sendhoff, B. (2008). Pareto-based Multiobjective Machine Learning: An Overview and Case Studies. *IEEE Transactions on Systems, Man, and Cybernetics, 38*(3), 397- 415.

Kamber, H. (2007). *Data Mining: Concepts and Techniques* (2nd ed.). Amsterdam: Elsevier.

Kamila, N., Jena, L., & Bhuyan, H. (2016). Pareto-based Multi-objective Optimization for Classification in Data Mining. *Cluster Computing, 19*(4), 1723-1745.

Kaur, G., & Chhabra, A. (2014). Improved J48 Classification Algorithm for the Prediction of Diabetes. *International Journal of Computer Applications, 98*, 13-17.

Kong, G., Jiang, L., & Li, C. (2016). Beyond Accuracy: Learning Selective Bayesian Classifiers with Minimal Test Cost. *Pattern Recognition Letters, 80*, 165-171.

Kulkarni, E., & Kulkarni, ,. R. (2016). WEKA Powerful Tool in Data Mining. *International Journal of Computer Applications, 2*, 10-15.

Li, F., Zhang, X., Zhang, X., Du, C., Xu, Y., & Tian, Y. (2018). Cost-sensitive and Hybrid-attribute Measure Multi-decision Tree over Imbalanced Datasets. *Information Sciences, 422*, 242-256.

Lim, T., & Loh, W. (2000). A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms. *Machine Learning, 40*(3), 203–228.

Ling, C., Sheng, V., & Yang, Q. (2006). Test Strategies for Cost-sensitive Decision Trees. *IEEE Transactions on Knowledge and Data Engineering, 18*(8), 1055- 1067.

Ling, C., Yang, Q., Wang, J., & Zhang, S. (2004). Decision Trees with Minimal Costs. *International Conference on Machine Learning*, (pp. 1-8). Canada.

Lu, H., Yang, L., Yan, K., Xue, Y., & Gao, Z. (2017). A Cost-sensitive Rotation Forest Algorithm for Gene Expression Data. *Neurocomputing, 228*, 270-276.

Maimon, O., & Rokach, L. (2005). *Data Mining and Knowledge Discovery Handbook.* New York: Springer.

Marbán, O., Menasalvas, E., & Fernández-Baizán, C. (2008). A Cost Model to Estimate the Effort of Data Mining Projects (DMCoMo). *Information Systems, 33*(1), 133-150.

Marler, R., & Arora, J. (2004). Survey of Multi-objective Optimization Methods for Engineering. *Structural and Multidisciplinary Optimization, 26*(6), 369-395.

Min, F., He, H., Qian, Y., & Zhu, W. (2011). Test-cost-sensitive Attribute Reduction. *Information Sciences, 181*(22), 4928-4942.

Neto, J., Souza, Z., Oliveira, S., Kolln, O., Ferreira, D., Carvalho, J., Braunbeck, O., Franco, H. (2017). Use of the Decision Tree Technique to Estimate Sugarcane Productivity Under Edaphoclimatic Conditions. *19*(6), 662-668.

Niha, S. (2017). Study of Data Mining Methods and its Applications. *International Research Journal of Engineering and Technology (IRJET), 4*(11), 1910-1913.

Ohta, S., Kurebayashi, R., & Kobayashi, K. (2008). Minimizing False Positives of a Decision Tree Classifier for Intrusion Detection on the Internet. *Journal of Network And Systems Management, 16*(4), 399-419.

Pallant, J. (2010). *SPSS Survival Manual (4th ed.).* Maidenhead: Open University Press.

Patel, N., & Upadhyay, S. (2012). Study of Various Decision Tree Pruning Methods with their Empirical Comparison in WEKA. *International Journal of Computer Applications, 60*(12), 20-25.

Patro, S., & sahu, K. (2015). Normalization: A Preprocessing Stage. *International Advanced Research Journal in Science, Engineering and Technology, 2*(3), 20-22.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing Misclassification Costs. *Proceedings of the 11th International Conference on Machine Learning*, (pp. 217-225).

Pipino, L. (2004). Data Mining, Dirty Data, and Costs. *Proceedings of the 9th International Conference on Information Quality (ICIQ-04)*, (pp. 164-169).

Qiu, C., Jiang, L., & Li, C. (2017). Randomly Selected Decision Tree for Test-cost Sensitive Learning. *Applied Soft Computing, 53*, 27-33.

Rajput, S., & Arora, A. (2013). Designing Spam Model-classification Analysis using Decision Trees. *International Journal of Computer Applications, 75*(10), 6-12.

Roiger, R., & Geatz, M. (2003). *Data Mining: A Tutorial-Based Primer.* Addison Wesley.

Rokach, L., & Maimon, O. (2005). *Data Mining and Knowledge Discovery Handbook.* Boston: Springer Science, Business Media, Inc.

Rokach, L., & Maimon, O. (2008). *Decision Trees : Theory and Applications.* Singapore: World Scientific.

Sahin, Y., Bulkan, S., & Duman, E. (2013). A cost-sensitive Decision Tree Approach for Fraud Detection. *Expert Systems with Applications, 40*, 5916-5923.

Sharma, G. (2016). A Study on Data mining Algorithms for Tourrism Industry. *International Journal of Latest Trends in Engineering and Technology, 7*(1), 580-587.

Sheng, S., Ling, C., & Yang, Q. (2005). Simple Test Strategies for Cost-sensitive Decision Trees. In Gama, J., Camacho, R., Brazdil, P., Jorge, A., & Torgo, L., *Machine Learning: ECML 2005. Lecture Notes in Computer Science* (Vol. 3720, pp. 365-376). Berlin, Heidelberg: Springer.

Sheng, V., & Ling, S. (2006). Feature Value Acquisition in Testing: A Sequential Batch Test Algorithm. *Proceedings of the 23rd International Conference on Machine Learning*, (pp. 809-816). Pittsburgh.

Sokolova, M., & Lapalme, G. (2009). A Systematic Analysis of Performance Measures for Classification Tasks. *Information Processing and Management, 45*(4), 427- 437.

Stiglic, G., Kocbek, S., Pernek, I., Kokol, & P. (2012). Comprehensive Decision Tree Models in Bioinformatics. *PLoS ONE, 7*(3), 1-13.

Tomar, D., & Agarwal, S. (2014). A Survey on Pre-processing and Post-processing Techniques in Data Mining. *International Journal of Database Theory and Application, 7*(4), 99-128.

Turney, P. (1995). Cost-sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm. *Journal of Artificial Intelligence Research, 2*, 369-409.

Turney, P. (2000). Types of Cost in Inductive Concept Learning. *17th International Conference on Machine Learning*, (pp. 15-21). California.

UCI Machine Learning Repository. (2018). Retrieved from University of California: https://archive.ics.uci.edu/ml/datasets.html

Urso, A., Fiannaca, A., Rosa, M., Ravì, V., & Rizzo, R. (2018). *Encyclopedia of Bioinformatics and Computational Biology*. Palermo: Elsevier.

Wang, T., Qin, Z., Jin, Z., & Zhang, S. (2010). Handling Over-fitting in Test Cost-sensitive Decision Tree Learning by Feature Selection, Smoothing and Pruning. *The Journal of Systems and Software, 83*(7), 1137-1147.

Weiss, G. (2005). Data Mining in Telecommunications. In Maimon, O., & Rokach, L., *Data Mining and Knowledge Discovery Handbook* (pp. 1189-1201). Boston: Springer.

Weiss, Y., Elovici, Y., & Rokach, L. (2013). The CASH Algorithm-cost-sensitive Attribute Selection using Histograms. *Information Sciences, 222*, 247-268.

Weka 3: Data Mining Software in Java. (n.d.). Retrieved from The University of Waikato: https://www.cs.waikato.ac.nz/ml/weka/

Witten, I., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques (2nd ed.)*. Amsterdam: Elsevier.

Witten, I., Frank, E., & Hall, M. (2011). *Data Mining: Practical Machine Learning Tools and Techniques (3rd ed.)*. Elsevier.

Wu, X., & Kumar, V. (2009). *The Top Ten Algorithms in Data Mining.* Boca Raton, USA: CRC Press.

Xu, Z., Min, F., & Zhu, W. (2013). Cost-sensitive Decision Trees with Post-pruning and Competition for Numeric Data. *Journal of Computational Information Systems, 9*(20), 7991- 8001.

Yang, Q., Ling, C., Chai, X., & Pan, R. (2006). Test-cost Sensitive Classification on Data with Missing Values. *IEEE Transactions on Knowledge and Data Engineering, 18*(5), 626- 638.

Zhang, G. (2013). Pareto Optimal Solution Analysis of Convex Multi-objective Programming Problem. *Journal of Networks, 8*(2), 437-444.

Zhao, H., Li, X., Xu, Z., & Zhu, W. (2015). Cost-sensitive Decision Tree with Probabilistic Pruning Mechanism. *International Conference on Machine Learning and Cybernetics* (pp. 81-87). Guangzhou: IEEE.

Zhou, Q., Zhou, H., & Li, T. (2016). Cost-sensitive Feature Selection using Random Forest: Selecting Low-cost Subsets of Informative Features. *Knowledge-Based Systems, 95*, 1-11.

Zuber, V., & Faltings, B. (2006). Overcoming Incomplete User Models in Recommendation Systems Via an Ontology. In Nasraoui, O., Zaïane, O., Spiliopoulou, M., Mobasher, B., Masand, B., & Yu, P., *Advances in Web Mining and Web Usage Analysis. WebKDD 2005. Lecture Notes in Computer Science* (Vol. 4198, pp. 39-57). Berlin, Heidelberg: Springer.

Appendices

Appendix (1): Pseudo Code for the Decision Tree Algorithm

```
TreeGrowing (S, A, y, SplitCriterion, StoppingCriterion)
Where:
S - Training Set
A - Input Feature Set
y - Target Feature
SplitCriterion - the method for evaluating a certain split
StoppingCriterion - the criteria to stop the growing process
Create a new tree T with a single root node.
IF StoppingCriterion(S) THEN
    Mark T as a leaf with the most
    common value of y in S as a label.
ELSE
    ∀a_i ∈ A find a that obtain the best SplitCriterion(a_i, S).
    Label t with a
    FOR each outcome v_i of a:
        Set Subtree_i= TreeGrowing (σ_{a=v_i} S, A, y).
        Connect the root node of t_T to Subtree_i with
                an edge that is labelled as v_i
    END FOR
END IF
RETURN TreePruning (S, T, y)
TreePruning (S, T, y)
Where:
S - Training Set
y - Target Feature
T - The tree to be pruned
DO
    Select a node t in T such that pruning it
        maximally improve some evaluation criteria
    IF t ≠ ∅ THEN T = pruned(T, t)
UNTIL t = ∅
RETURN T
```

الملخص

يتناول البحث دراسة استقصائية لجوانب عديدة تتعلق بمشجّرات اتخاذ القرار، فمن ذلك: المفاهيم الأولية لمشجّرات اتخاذ القرار، والخوارزميات القياسية لإنشائها، بالإضافة إلى التطبيقات المختلفة لهذه المشجّرات في التنقيب في البيانات وبخاصة في مجالات التسويق والهندسة والتشخيص الطبي.

كما تناول البحث بشكل تفصيلي التكاليف المرتبطة بمهمة التصنيف في مشجّرات اتخاذ القرار وعلاقتها بدقة التصنيف، حيث أن الموازنة بين الاثنين تمثل تحدّيًا كبيرًا هذه الأيام في العديد من المجالات كالتشخيص الطبي مع تزايد تطبيقات هذه المشجرات في اتخاذ القرار.

كما ويركز البحث بشكل خاص على إيجاد طريقة للموازنة بين تكلفة ودقة إنشاء مشجرات القرار بغرض الحصول على مصنف دقيق ذو تكلفة منخفضة وذلك عن طريق تمثيل العلاقة بين الاثنين واستخدامها لتحديد التوازن الأمثل بين كليهما.

خلال البحث تم استخدام طرق مختلفة لتقليم مشجرات القرار كما تم استعمال مقاييس مختلفة بعناية تامة إذ أنها يمكن أن تحدث اختلافاً ملحوظاً في أداء المصنف. وقد تم اختيار الإعدادات المثلى لتقليم المصنفات المنشئة باستخدام خوارزمية القياسية لبناء مشجرات القرار وذلك باستخدام نهج يسمى "هيمنة باريتو" وذلك لما يعرف عن فاعليته في حل المشاكل المرتبطة بالتنقيب عن البيانات.

تشير النتائج التجريبية إلى كفاءة استخدام أساليب التقليم المختلفة لتقليل متوسط إجمالي تكاليف التصنيف بشكل ملحوظ مع انخفاض طفيف في الدقة في بعض الأحيان. كما تشير النتائج إلى وجود علاقة عكسية بين متوسط إجمالي تكاليف التصنيف ودقته. عند مقارنة الحلول المقترحة مع الخوارزمية إنشاء المشجرات القياسية مع إعدادات التقليم الأساسية، أشارت النتائج إلى أن متوسط تكلفة التصنيف الإجمالية قد هبطت بمقدار 21.14٪، 19.79% و 5.27% في قاعدة بيانات سرطان الثدي وأمراض القلب والغدة الدرقية، على التوالي. كما أنه تم تخفيض معدل الدقة الناتج بنسبة 1.6 ٪ فقط في قاعدة بيانات سرطان الثدي. أما في قاعدة بيانات أمراض القلب فقد تم زيادة دقة المصنف الناتجة بنسبة 6.68٪ وتم الحصول على نفس معدل الدقة في قاعدة بيانات أمراض الغدة الدرقية.

جامعة البحرين

كلية تقنية المعلومات

"الموازنة بين الدقة والتكلفة في بناء مشجّرات اتخاذ القرار"

أطروحة مقدمة كجزء من متطلبات الحصول على درجة الماجستير في تقنية المعلومات

إعداد

منى منصور ميرزا علي الحمد 20021438

إشراف

د. أحمد محمد زكي

مملكة البحرين

أبريل 2019