Post-Pruning Decision Tree Algorithms Based on Bayes Minimum Risk and Loss Minimization to Combat Over-Fitting Problem

Ahmed Mohamed Ahmed Mohamed

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Applied Mathematics and Computer Science

Eastern Mediterranean University
May 2018
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Assoc. Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree ofDoctor of PhilosophyinApplied Mathematics and Computer Science.

Prof. Dr. Nazim Mahmudov
Chair, Department of Mathematics

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree ofDoctor of PhilosophyinApplied Mathematics and Computer Science.

Assoc. Prof. Dr. Ali Hakan Ulusoy Co-Supervisor	Assoc. Prof. Dr. Ahmet Rizaner Supervisor

Examining Committee

1.Prof. Dr. Rashad Aliyev

2.Prof. Dr. Hamza Erol

3.Prof. Dr. Benedek Norbert Nagy

4.Prof. Dr. Efendi Nasiboğlu

5.Assoc. Prof. Dr. Ahmet Rizaner

ABSTRACT

Pruning is applied in order to combat over-fitting problem where the tree is pruned with the goal of identifying decision tree with the lowest error rate on previously unobserved instances. Post-pruning approach is one of the most powerful and effective pruning methods in decision trees. This method traverses the tree in bottom-up fashion removing the unproductive branches based on the misclassification error rate of each parent node and its leaves. In this thesis, two post-pruning approaches named as Pruning with Bayes Minimum Risk (PBMR) and Pruning based on Zero-One Loss Function (ZOLFP) are proposed. In PBMR approach, the tree is pruned based on the estimating risk-rate and a parent node of a subtree is converted to a leaf node if the estimated risk-rate of the parent node for that subtree is less than the sum of the risk-rates of its leaves. ZOLFP approach minimizes the expected loss of the tree by employing Zero-One loss function method. A subtree is pruned when expected loss of the node is less than or equal to the sum of the loss of its leaves. The two proposed methods are evaluated in terms of performance accuracy, tree complexity, precision/recall scores, True Positive (TP) and False Positive (FP) rates and area under ROC considering attribute selection. The experimental results show that the two proposed methods produce better classification accuracy compared to un-pruned decision tree, Reduced-Error Pruning (REP), and Minimum-Error Pruning (MEP) with complexity not much different than the complexities of REP and MEP. The experiments also demonstrate that the proposed methods show satisfactory performance in terms of precision/recall score, TP rate, FP rate and area under ROC.

Keywords: decision tree, pruning, post-pruning, Bayes minimum risk, Zero-One loss

ÖZ

Budama, en düşük hata oranı ile karar ağacını tanımlama sırasında meydana gelen aşırı-uydurma sorunu ile mücadele etmek amacıyla uygulanır. Sonradan-budama yaklaşımları karar ağaçlarında en etkili budama yöntemlerinden olup her bir ana düğümün ve yapraklarının yanlış sınıflandırma hata oranına dayanarak verimsiz dallarını çıkartarak ağacı aşağıdan yukarıya doğru inceler. Bu tezde, Bayes Minimum Risk (PBMR) ve Sıfır-Bir Kayıp Fonksiyonu (ZOLFP) temel alınarak geliştirilen iki sonradan-budama yaklaşımı önerilmiştir. PBMR yaklaşımında, bir alt ağacın ana düğümü, bu alt ağacın ana düğümünün tahmini risk oranının yapraklarının toplam risk oranlarından daha az olması durumunda bir yaprak düğümüne dönüştürülür. ZOLFP yaklaşımı, Sıfır-bir kayıp fonksiyonuna bağlı bir yöntem kullanarak ağacın beklenen kaybını en aza indirir. Önerilen iki yöntem performans doğruluğu, ağaç karmaşıklığı, hassasiyet/geri-çağırmaskoru, Gerçek Pozitif (TP) ve Yanlış Pozitif (FP) oranları, öznitelik seçimi veKarar Vericinin Etkinliği (ROC) altındaki alan dikkate alınarak değerlendirilmiştir. Deneysel sonuçlar, önerilen her iki yöntemin, Azaltılmış Hata Budama (REP) ve Minimum Hata Budama (MEP) yöntemlerinin karmaşıklığına yakınbir karmaşıklık ile budamayı gerçekleştirdiğini vebu yöntemler ile karşılaştırıldığında daha iyi sınıflandırma doğruluğu ürettiğini göstermektedir. Deneyler, önerilen yöntemlerin, hassasiyet skoru, geri-çağırma skoru, TP oranı, FP oranı ve ROC altındaki alan açısından da tatmin edici bir performans sergilediğini göstermektedir.

Anahtar Kelimeler: karar ağacı, budama, sonradan-budama, Bayes minimum risk, Sıfır-bir kayıp

DEDICATION

To my parents, wife, sons and my daughter

ACKNOWLEDGMENT

I wish to record my deepest thanks and gratitude to my supervisor Assoc. Prof. Dr. Ahmet Rizaner, and my co-supervisor Assoc. Prof. Dr. Ali Hakan Ulusoy for their advices, support, and guidance during all stages of this thesis. I highly appreciate their valuable suggestion and cooperation.

My great thanks and appreciations go to my father and mother for their support and all sacrifices they made for me to reach this stage, I never forget their every Friday calls.

My great thanks to my brother and sister for their continuous support and encourage.

My special gratitude goes to my friend Samer Khalil who is in the jail because of his political opinions. My deep thanks for his support, encouragement, and always help for my family during my absence. Words cannot express how I am grateful. My special thanks to my friend Nazim Khalil for all kinds of help and supports he provides for me.

I would like to record my deep thanks and love to my two sons and daughter whom I owe my life. I hope this attempt light up their road in life.

I would like to express my deepest thanks to my wife for her support, encourage, and taking care of our family during my absence, I highly appreciate her sacrifices, and I would like to dedicate this thesis to my family wife, sons, and daughter.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CART	Classification and Regression Trees
DM	Data Mining
DT	Decision Tree
FN	False Negative
FP	False Positive
ID3	Iterative Dichotomize
InfoGainAttributeEval	Information Gain Evaluator
KDD	Knowledge Discovery in Database
MDL	Minimum Description Length
MEP	Minimum-Error Pruning
MLP	Multi-Layer Perception
NB	Naïve Bayes
OneR	One Rule
PBMR	Pruning with Bayes Minimum Risk
REP	Reduced-Error Pruning
RL	Reinforcement Learning
ROC	Receiver Operating Characteristic
ROCCH	Receiver Operating Characteristic Convex Hull

SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TCSDT	Test Cost-Sensitive Decision Tree
TN	True Negative
TP	True Positive
UCI	University of California-Irvine
UDT-C4.5	Un-pruned C4.5 DT Algorithm
ZOLFP	Zero-One Loss Function Pruning

Chapter 1

INTRODUCTION

1.1 Introduction

Nowadays the data size becomes very large due to the rapid growth in information technology. Data accumulated from various fields such as engineering, banking and medicine. During this explosive growth of data, the importance of finding valuable information become necessary, therefore different data analysis techniques have been introduced to extract useful information from this huge data [1]. In the past, data analysis techniques that based on statistics were being used to discover patterns and build predictive models. The high progress in databases and information technologies reflects the need for new approaches that capable to store and manipulate this huge chunk of data. So, data mining has been developed independently of statistics though "mining data" to predict future behaviour. Data mining is the process of extracting useful information from row data for further decision making. It is also known as knowledge discovery process, knowledge mining from data, or data analysis.

Various data mining algorithms and techniques are used for analyzing and mining databases to uncover hidden pattern such as classification, clustering, and regression [2]. Among these techniques classification is the most widely applied data mining technique in commercial and scientific domains. Classification is the process of generalizing a set of pre-classified examples to develop a new model that is capable to classify the population of records at large. This process involves two stages learning and classification. During learning stage, the training data is analyzed by using one of the classification methods to build a learning model that is represented in the form of classification rule, while in classification stage data is tested to estimate the accuracy of the classification rules. For example, fraud detection program that might classify a user log in as “authorized” or as “non-authorized”. Moreover, data classification algorithms try to investigate the relationships between the attributes and discover which one is capable to predict the output based on given input. The algorithm is feed by new dataset not used before with the same attribute set, excluded the prediction attribute that is not known yet. The algorithm analyses the input and produces a prediction. The prediction accuracy is used as measurement to justify how efficient the algorithm is and how data classification algorithms can perform the classification of instances with high accuracy to their correct feature (attribute) space. We consider here supervised classification methods, which provide the learning algorithms with known quantities to support future data processing. In the literature, the researchers introduced several classification methods such as decision tree, neural network, and support virtual machine. Each classifier characterizes objects by means of a set of features or parameters that are related to the task. Decision trees that are a flow like structure is one of the effective data mining techniques that is widely used due to their high accuracy [3]. Induction is performed in top down method and some attribute selection measures are used to select attributes like information gain, gain ratio, gain index etc. During the induced process, a decision tree may generate unwanted and meaningless tree that is large in size with more complexity because of over-fitting problem [4]. Therefore, pruning methods are introduced to combat over-fitting problems that affect the performance accuracy of the dataset. Two types of pruning are used: post-pruning and pre-pruning. In post-pruning, initially the tree is grown and then unsuitable branches are removed. Hence, post-pruning decision tree produces the (complete) tree and then adjusts it in order to improve the classification accuracy on unseen instances. While in pre-pruning the over-fitting is checked during the tree building process. There are two techniques for pre-pruning: minimum number of object pruning and chi-square pruning.

Several post-pruning methods have been introduced in the literature such as Reduced-Error Pruning (REP), error complexity pruning, Minimum-Error Pruning (MEP), and cost-based pruning. Among these techniques, REP produces smaller trees with better accuracy. REP traverses the decision tree in post or derby checking all internal nodes and replacing subtrees with their leaf nodes when the former misclassification error rate is no smaller than the later. Furthermore, in this method as in some other post-pruning methods, the dataset is divided into three sets as training set that is used for training, the validation set that is used for pruning the tree, and test set that is used for an unbiased estimation over future unseen instances [4].One of the advantages of REP method is the linearity of its computational complexity [5], but it tends to over pruning when the test set is too smaller than the training set.

The aim of most pruning algorithms is to minimize the expected error rate of the decision tree. Recently, some post-pruning methods have been introduced to prune the tree with respect to the loss matrix. So, in such cases, it may be desirable to prune the tree by estimating the expected loss instead of estimating the misclassification error rate. Pruning for loss minimization is another point of view in pruning that may result in different pruning behavior than does pruning for error minimization. This type of pruning relays on the probability distribution to adjust the prediction that leads to minimize the expected loss [6].On the other hand, some researchers adopted the idea of incorporating the general loss matrix into C4.5 decision tree algorithm to reduce the misclassification cost or to minimize the misclassification loss. These algorithms developed heuristic splitting methods that perform dual tasks. The first task is to select the splitting that minimizes the misclassification cost and the second task is to select proper split for subsequent splits.

Another technique to avoid over-fitting problem is appropriate selection of features. Here, the main idea is how to select the best features at each node and removing the redundant features that may increase the complexity. Generally, feature selection process has two steps. Firstly, an appropriate algorithm is used to select a subset of features and proceed to find an optimal split point, and secondly, another algorithm is run for evaluating features to find which one provides better performance by applying some goodness measures. Consequently, stopping criterion is needed to terminate the feature selection process from running exhaustively or forever. Hence, feature selection methods select the most appropriate feature among all the competing candidate subsets of features by employing some evaluation criteria. This process is like exhaustive search such as trying to find the best feature only even if the set of features is not too large.

1.2 Motivation

The advance progresses in information technologies result in a large amount of data that need to be analyzed and managed to gain useful information knowledge to predict future behavior. As a result, several data analysis techniques have been developed to process this accumulated data that gathered in digital environment so as to acquire meaningful knowledge for future benefits. Among these introduced techniques, data mining is one of the vital techniques that provide meaningful results when processing data, accordingly it draws the attention of several researchers. The process of analyzing this raw data is known as data mining, machine learning, and Knowledge Discovery in Database (KDD). KDD consists of overlapped sub-process that is used to extract large databases, while data mining is a sub-process in KDD that is utilized to analysis the data by using machine learning algorithms.

Decision tree is one of the most widely popular machine learning methods that is used for data analysis to acquire valuable meaning from row data. Decision tree is powerful, easy to understand and interpret, and produces efficient results, so that until nowadays being researched actively. Decision process in decision tree is similar to human decision process so that they are easier to understand and interpret. Further, decision trees are nonparametric in building classification models, so the problem of finding the optimal decision tree is NP-complete problem. Despite decision trees are more accurate and efficient compared to other learning methods, it happens that they may provide very large trees in size that makes the tree un-understandable. This is called over-fitting problem that tends to reduce decision tree efficiency. To overcome this problem, pruning methods are introduced. Removing the anomalies and meaningless branches optimize the computational efficiency of the tree and as well as the accuracy.

1.3 Research Issue

A decision tree is a flowchart-like tree structure, in which internal nodes denote test nodes, branches denote an outcome of test nodes, and leaf nodes hold class labels. Decision trees suffer from over-fitting problem that appears during data classification process. Decision tress sometimes produces a tree that is large in size with unwanted branches, so that the tree becomes difficult to understand with low predictive efficiency. Pruning methods are introduced to combat this problem by removing the non-productive and meaningless branches to avoid the un-necessary tree complexity. The main aim of this thesis is to introduce post-pruning methods that are capable to combat the over-fitting problem.

1.4 Organization

The thesis is organized as follow. In Chapter 1, the introduction, motivation and research issues are figured out, whereas in Chapter 2, the literature review and background studies are carried out to explain the previous different techniques that have been introduce in the state of art. In Chapter 3, the concept of data mining is explained followed by clarification of the notation of KDD and machine learning. Further in Chapter 3, a case study of the performance analysis of some data mining techniques is presented. In Chapter 4, decision tree concept and characteristics are discussed in detail in terms of decision tree attribute selection methods, decision tree induction, and advantages and disadvantages of decision tree methods. Meanwhile decision tree over-fitting problem and generalization error estimation are explained and clarified, followed by a brief definition for decision tree pruning methods and types. In Chapter 5, decision tree post-pruning approach is investigated and studied in detail. In this chapter, the notations of new proposed methods are explained and experimental results about the proposed methods are presented. Finally, the conclusion of the thesis and future work are explained in Chapter 6.

Chapter 2

LITERATURE REVIEW

2.1 Related Works

During the past decades, several researches have been conducted in data classification to solve problems in various fields such as finance, medical, the stock market, manufacturing, telecommunication, health care and customer relationship. The Decision Tree (DT) is mostly used for the purpose of data classification that supports human decision making in an understandable and perfect form [7, 8]. The DT is an oriented graph that consists of a finite number of nodes which can be subdivided to root nodes, inner nodes, and leaves. DTs are built by using descending strategy that operates recursively starting with the full dataset from the root node. By employing some certain criteria, the dataset is split into smaller subsets, thus applying split criteria recursively all subsets become pure [9]. Since DTs are flexible and obvious in presenting classification procedure, their results are acceptable in knowledge discovery [10].

Most of classification algorithms like the DT suffer from over-fitting, because most of the real-world data sets contain noisy data and irrelevant correlated attribute that affect the classification algorithms and lead to poor predication accuracy on unseen data. Selecting the best feature for the DT induction at nodes consequently results in reducing over-fitting. Feature selection is performed in sequential manner by adding and removing features one by one. For instance, in [11] researchers developed new

DT induction method named as linear discriminant trees that trained the linear model with the whole feature space by assigning weights to each feature and then removed all the irrelevant variables. While in [12], authors developed new splitting criteria for the DTs so called correlation based multi branch support vector. The concept of correlation was employed in order to select the best splitting variable, and then the Support Vector Machine (SVM) specified the related variable. Since the tree here returned more thresholds, at each node the tree was divided to one subset or more. The results showed that the DT outperformed compared algorithms in term of shorter height, and less number of useful features.

Following that in [13], a new feature selection method was proposed for the SVMs, which were computationally suitable for high dimensional datasets. The proposed method was trained on toy and real datasets and the experiments reflected promising results. Hence the SVMs' speed for critical time applications increased. The experiments also showed that the SVMs sometimes suffered from high dimensional spaces that contained more irrelevant features where the proposed method followed a new way to avoid this naturally occurring complex problem.

In particular, many post-pruning algorithms for the DTs have been introduced in the literature to overcome over-fitting problem such as REP, pessimistic pruning, error-based pruning, cost-complexity pruning and minimum-error pruning [14-16]. Each of these algorithms attempted to produce simple tree structure with high accuracy, by estimating the misclassification errors at each decision node and propagated this error up the tree. The authors in [17] conducted a research which compared several pruning methods for error minimization. However, researchers in [18] deduced that when error minimization was the evaluation criterion, most pruning algorithms resulted in trees that were larger than necessary. Although the research in [19] performed an empirical comparison for five pruning methods, the experiment results showed that the methods such as critical-value pruning, error complexity pruning and REP outperformed the pessimistic-error pruning and minimum-error pruning in terms of the tree size and accuracy. Consequently, authors in [20] studied REP in different variants that were adding a new perspective to its algorithmic properties, analyzing the algorithm with less assumption compared to previous analyses methods, and emptying subtrees in the analyses process.

Furthermore, the study about error-based pruning algorithm clarified that varying the certainty factors resulted in a smaller tree [21]. Therefore, error-based pruning produced applicable tree size with good accuracy compared to REP. REP method in the DT was also analyzed in [22]. This study investigated the influence of pruning on the accuracy and tree size. The results showed that the produced tree was with small size and high accuracy. Post-pruning DT algorithm that was based on C5.0 DT algorithm and Bayesian posterior theory was introduced in [23]. The proposed method outperformed the original C5.0 DT algorithm and revealed that using Bayesian posterior theory as an enhancer for C5.0 classifier resulted in less memory and less classification time to search and build the rules. In [24], the researchers performed a comparative study to investigate and analyze six well-known post-pruning techniques namely, REP, pessimistic error pruning, MEP, critical value pruning, cost complexity pruning, and error based pruning respectively. The researchers highlighted the theoretical weakness and strengths of each method. Their results showed that REP produced the smallest subtree with the lowest error rate with respect to the pruning set. However, in [25] the researchers introduced a new DT algorithm based on J48 and REP. The new method was compared to original J48 DT and the results showed that their method produced a smaller tree and produced better performance accuracy. In [26], the researchers implemented a DT induction algorithm with REP technique to improve the performance accuracy. Their proposed method generated an optimal DT with less complexity and better performance accuracy.

Moreover, the research in [27] investigated multi-label classification problem. In case of problem one instance can belong to more than one class at a time. The researchers introduced a new pruning technique known as PruDent. In terms of performance accuracy, PruDent was more accurate compared to other state-of-the-art approaches and its computational costs were linear. On the other hand, one of the drawbacks of this method was the reliance on confidence scores. Further, authors in [28] adopted ability, stability and scale as a new classification standard for classification evaluation. Results showed that the improved method solved problems better than single standard evaluation methods and reflected more advantages. So, this improved method produced a more balanced classification performance and less model complexity. The researchers in [29] introduced a new DT algorithm that was named as Competition Cost-sensitive C4.5 for numeric data based on C4.5. They designed a heuristic function that was based on the test cost and the information gain ratio. The results showed that the proposed post-pruning algorithm was more effective and stable, so it produced a DT with less cost. In [30], the researchers developed a post-pruning DT algorithm that was based on Bayesian theory, where the branches that were generated by C4.5 algorithm validated by the Bayesian theorem. So that, this method produced a smaller tree since the branches out of the condition range were removed. The results showed that the proposed method produced a simpler DT compared to the original C4.5 algorithm. In [31], various measurement techniques were utilized to evaluate the performance of post-pruning methods like accuracy, stability, and simplicity. A multi-objective evaluation was proposed to select the best sub-tree during the post-pruning process. Moreover, the researchers developed a procedure for obtaining the optimal sub-tree based on user provided preference and value function information.

The researchers in [32] conducted a comparison study for REP method in DT. The performance of REP was analyzed, and they deduced that if the algorithm produced simple tree with low accuracy it meant the algorithm computed high misclassification during instance learning process. The results showed that J48 and REP produced a tree with high accuracy of classification with less complexity. Nevertheless, research in [33] introduced a new pruning method that aimed at improving both classification accuracy and tree size. The newly introduced method was first applied to a DT by implementing pre-pruning at the inducing phase of the tree, and post-pruning after the inducing process. In [34], researchers introduced a new DT pruning method based on backpropagation neural networks, called soft-pruning. Firstly, C4.5method was employed for the DT induction and then obtained trees were pruned by using soft pruning method. The experiment results indicated that soft pruning method performed better than original C4.5pruned and un-pruned trees. Additionally, in [35], authors proposed a new C5.0 classifier method that employed feature selection, cross validation, REP and model complexity for the original C5.0 method to reduce the tree size and improve the accuracy. The experimental results showed that when feature selection was applied, the attribute space was reduced for feature set while applying cross validation technique provided a more reliable estimate of predictive. On the other hand, when the model complexity was increased, the accuracy of the classification was also increased, and whenever

REP method was applied, the over-fitting problem of the DT was solved. The experiment results showed that the accuracy of the proposed method was improved compared to the original C5.0 method.

A new pruning method called Multi-Level Pruned Classifier that integrated the pruning phase into the building phase was developed in [36]. To evaluate the effectiveness of their proposed method Credit Card Database was utilized. The experiments were conducted on the dataset with pruning and without pruning in terms of complexity and classification accuracy. Instead of classifying the transactions to either fraud or non-fraud it desired to be classified into four risk levels, such that the proposed method showed a promising progress. While researchers in [37] proposed a DT method that adopted Rough Set Theory for pruning, their proposed method introduced depth-fitting ratio which involved both the depth and the explicit degrees of the sub-trees under evaluation. The experimental results demonstrated that new proposed method was feasible and effective for pruning. The constructed DT sizes were quite reduced, while the prediction accuracy was well improved. Furthermore, in [38], a multi-strategy pruning algorithm was introduced to trim the tree. During the pruning process, three groups of strategies were implemented to get the optimal solution, which were namely, simple size, degree of matching, and scale of the tree. The experimental results illustrated that the multi-strategy pruning algorithms for the DT pruning improved the efficiency and accuracy of intrusion detection system. In [39], the researchers introduced a DT pruning method based on genetic algorithms. The experimental results indicated that applying genetic algorithms in the DT pruning was effective and feasible; the pruning operation was converted to procedures that enhance the edge weight. The novel approach was compared with some DT pruning techniques including cost-complexity pruning, pessimistic error pruning and REP. The results showed that the new method had better or equal effect with other pruning methods. A unifying framework based on the four-tuple (space, operators, evaluation function, and search strategy) was introduced in [40]. Six well-known pruning methods were studied based on the proposed framework in order to specify the aspects, strengths and weaknesses of these pruning methods, these pruning methods are namely, REP, pessimistic error pruning, cost complexity pruning, MEB, critical value pruning, and error based pruning respectively.. The results demonstrated the fact that pruning methods did not reduce the productivity, but they could enhance the final tree accuracy.

Most of classification algorithms aim to minimize the misclassification error rate of datasets, if the cost of misclassification errors is equal. Indeed, the real-world applications demonstrated that this assumption is not correct. Different misclassification errors can have quite large different costs. Meanwhile, cost sensitive learning considers misclassification costs instead of misclassification errors. The main aim of this type of learning is to produce a high accuracy of classifying examples into a set of known classes [41]. Though researchers in [42] investigated how misclassification cost is vital in many real-world problems. They provided a solution to the problem of the cost that became different for different examples. The introduced method known as direct cost sensitive was compared to MetaCost method. This method estimated the class probability distribution for each example and then using cost matrix to compute the optimal class label for each test example. It was proven that the new method was preferable compared to MetaCost method.

In [43], ROC Convex Hull (ROCCH) method was introduced to solve the problem of comparing multiple classifiers in imprecise environment. The introduced method reduced the management of classifier performance data by selecting the optimal classifiers. The proposed ROCCH method was tested under different conditions capable to handle many real-world realistic problems. Moreover, authors in [44] presented three methods to tackle over-fitting problem for Test Cost-Sensitive Decision Tree (TCSDT) method which are namely feature selection, smoothing and threshold pruning. The feature selection method was applied first to pre-process the dataset and then smoothing and pruning process were conducted before calculating the class probability estimate for each DT leaf, followed by applying TCSDT algorithm to perform the classifications process. The experiments proved that the introduced TCSDT method performed better than original TCSDT and other competing algorithms on the selected datasets. Recent researches provided successfully new algorithms that produced robust learner which capable to minimize both testing costs and misclassification costs. Accordingly, authors in [45] developed a framework that based on DT qualified to generated accurate decisions within a strict bound on testing cost. The experimental results showed that the proposed framework produced trees with lower misclassification costs along a wide range of testing cost bounds. So far, an experimental study for cost complexity pruning and C4.5’s error-based pruning that concentrated on pruning with loss minimization and probability estimation instead of error minimization was conducted in [46]. The study revealed that when the probability was estimated by Laplace correction at leaves level, all pruning methods were improved.

Chapter 3

MINING AND DATA ANALYSIS TOOLS

3.1 Overview of Data Mining Tools

In recent years, the volume of digitalized data has grown rapidly from various fields due to the advance progress in information technology, engineering, banking, and medicine. It has been reported that the volume of this data increased exponentially, and it becomes impossible to be analyzed manually in order to serve human needs. As a result, different data analysis techniques have been developed to handle the problem of how to acquire meaningful knowledge from this accumulated raw data for future benefits. Among the introduced techniques data mining is one of the vital techniques that provides noteworthy results when processing data. So, it draws the attention of several researchers. The process of analyzing this raw data is known as data mining, machine learning, and KDD.

3.1.1 Knowledge Discovery in Databases

KDD is referred to the process of obtaining valuable knowledge from large data, and it consists of several processes within each other. Hence, it consists of various research domains such as soft computing, data mining, pattern recognition, and databases [47]. KDD application area is very interesting and expands. It includes stock markets, security attacks, industrials, financial issued, telecommunications, and internet agents [47]. On the other hand, KDD also processes digital data that is gathered in social site and internet environment. The main objective of KDD is to provide users with simple understandable knowledge.

3.1.2 Data Mining

Nowadays, Data Mining (DM) has attracted a lot attention in data analysis area, and it became recognizable as a new tool for data analysis that can be used to extract valuable and meaningful knowledge from data. DM provides modern techniques to unveil hidden patterns within huge databases, where these hidden patterns can be very useful for predicting future decisions. DM techniques are used to fetch previous unknown patterns in large chunk data. When these patterns are extracted it can be used further to develop businesses. Nonetheless mining process involves three steps:

- Exploration
- Pattern identification
- Deployment

Exploration: This step is also known as data pre-processing. Data is cleaned and prepared to fit the learning algorithms' criteria, thus it contains the process of removing missing values, transforming data to another form and handling noise or outlier data.

Pattern identification: In the second step, the suitable DM approach is chosen according to the mining goal for example, classification, clustering, or regression.

Deployment: In step three, the appropriate search method is selected for patterns. After that the mined patterns are evaluated and interpreted with respect to the mining goal.

Classification is the most widely used DM technique to extract knowledge from huge data. The classification process involves two major steps: learning and classification.

In learning step, classifier analyzes the training data, while during classification step the classifier is applied on test data with the aim of estimating the performance accuracy of the classification rule. Then, these rules can be applied to the new data if its accuracy is acceptable [48]. Several classification algorithms have been introduced in the past years such as DT induction Bayesian classification, neural networks and SVM.

3.1.3 Machine Learning

During the past decades machine learning becomes one of the main tools in information technology that employed to extract meaningful information from large data sets. Machine learning detects valuable meaning from patterns by automated systems, such as search engines that are used to bring website best results for users, anti-spam software that filters e-mail messages to detect spam messages, and credit card transactions that are equipped with secured software in order to detect frauds. Machine learning field was derived from the field of AI, however machine learning growth from the question trigged that days: Can a machine think and learn like human beings? After the paper of Alan Turing's: "Computing Machinery and Intelligence" [49], the answer of this question leads to the birth of machine learning field. Hence machine learning is known as the science that teaches a machine to learn by itself from the available sources such as data and algorithms. The process of learning is performed by training one model to learn from the available data, then the produced model can be utilized to predict future learning processes, thus learning is acquired from experience. Most of nowadays intelligent systems contain smart programs that have the ability to learn and simulate human activities rather than following specific instruction. The smart software is designed by using machine learning algorithms. Lots of machine learning algorithms have been developed.

These algorithms are grouped in some levels based on specific characteristics. There are four types of groups: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning.

3.1.3.1 Supervised Learning

Supervised learning takes the possible outputs that are already known as an input during the learning process. The correct classification is already assigned to the source data before the learning algorithm is trained on the data. Thus, when new input with unknown target is given to the learning model, the model produces the correct target. Supervised learning aims to build models that represent the training data in proper and straightforward way. It contains such algorithms as DTs, artificial neural networks, naïve Bayes, SVMs, random forests and nearest neighbor algorithms. For example, after classification algorithm being trained on animal image dataset, it will be capable to detect animals since the dataset is labeled with some characteristics of these animals.

3.1.3.2 Unsupervised Learning

Unsupervised learning is used to extract hidden patterns from unlabeled input data. So, it is capable to assign inputs to correct targets without providing with previous knowledge. Hence, the unsupervised learning algorithm learns the data by itself without guidance. Many unsupervised learning algorithms are introduced but the most important algorithms are: clustering, expectation minimization algorithm and principal component analysis. Furthermore, unsupervised machine learning is similar to the so called true AI that based on the idea that is capable to learn how to rime complicated processes without human aids or guidance.

3.1.3.3 Semi-Supervised Learning

Semi-supervised learning is a group of supervised learning algorithms that has the ability to handle unlabeled data of unsupervised learning. Thus, semi-supervised learning algorithms manage both unlabeled data and a small amount of labeled data. This combination of learning methods increases the learning process overall accuracy, and outperforms some other supervised approaches when exposed to some certain conditions [50].

3.1.3.4 Reinforcement Learning

Reinforcement Learning (RL) is a kind of machine learning that contains supervised learning and dynamic programming. This combination generates powerful learning method [50]. RL has lots of applications in the fields of machine intelligence, such as robots, smart games and handy intelligent devices that learn from opponents.

3.2 Difference between Knowledge Discovery in Databases, Data Mining and Machine Learning

KDD is the process of extracting valuable knowledge from large databases, and it consists of various sub-processes which overlap in each other, and DM is a part of these sub-processes of KDD. While KDD is concerned with the whole extracting process, DM pays attention just to the analysis part. Further, KDD is considered to be multidisciplinary activity that includes DM as a vital data analysis part of the whole process. The difference between machine learning and DM is more overlapped because both of them handle similar problems. In fact, DM exploits machine learning algorithms to analyze raw datasets, while machine learning is concerned with the process of developing algorithms that enable computer systems to learn by itself without human aid. In particular, machine learning provides DM with the algorithms that are used to process chunk data to reveal interesting patterns. As mentioned above DM is a sub-process of KDD processes that extract raw data to find valuable knowledge.

3.3 Case Study: Performance Analysis of Selected Data Mining Techniques to Predict Instructor Performance

This study focuses on predicting the instructor performance and investigates the factors that affect students' achievements to improve the education system quality based on research in [51]. Turkey Student Evaluation records dataset is considered and run on different data classifier such as J48 DT, Multi-Layer Perception (MLP), Naïve Bayes (NB), and Sequential Minimal Optimization (SMO). The dataset is collected from University of California-Irvine (UCI) Machine Learning Repository and contains a total of 5,820 evaluation scores provided by students from Gazi University in Ankara, Turkey. There are 28 course specific questions and additional 5 attributes. The data set is tested and analyzed using above mentioned data. The experiment is conducted by applying Weka software, and then a comparison of accuracy is performed for all algorithms after the prediction process. It is found that using the attribute evaluation method on the dataset is helpful in order to predict the instructor performance. The most important attributes in the dataset are selected by utilizing One Rule (OneR) attribute evaluator of Weak, and then the above-mentioned algorithms are run on the dataset. The best 24 attributes are considered after removing the worst ten attributes with lower impacts on the dataset. Table 3.1 shows the accuracy of the prediction when mentioned algorithms are applied on the dataset after attribute evaluation is done. This means that these attributes are more significant in predicting the instructors' performances and accurately describe their experiences. On the other hand, Table 3.1 shows that SMO performs better than other algorithms with an accuracy level of 85.8%. Furthermore, from Table 3.1 it is also observed that J48 DT algorithm outperforms other algorithms when applied on all the dataset with an accuracy level of 84.8%. Table 3.2 shows the number of courses that are taught by each instructor, and the number of students who evaluated each instructor.

Table 3.1: Accuracy results after attribute evaluation process and when algorithms run on all dataset

Algorithm	Accuracy after attribute evaluation process for attributes with highest impacts	Accuracy when algorithms run on all data set for all attributes
J48 DT	85.1%	84.8%
NB	84.3%	83.3%
SMO	85.8%	84.5%
MLP	84.6%	82.5%

Another interesting issue is observed from the results which show that the performance of an instructor is mainly affected by the number of courses that is taught. Table 3.4 shows that all classification algorithms obtained lower prediction accuracy when running on instructor 3 dataset file, compared to the prediction accuracy obtained by those algorithms when run on instructor 1 and 2 dataset file. By comparing of all classifiers, SMO and MLP algorithms performed best among all classifiers with accuracies as 87.0%, 86.2% respectively for instructor 1 dataset as shown in Table 3.3. While the accuracy of SMO degraded, MLP continued to give the best performance with an accuracy of 87.2% for instructor 2 dataset as shown in Table 3.3.

Table 3.2: Instructors, courses, and numbers of students evaluated for each instructor

Instructor	Course Code	Total Number of Students
1	2, 7, 10	776
2	1, 6, 11, 13	1,444
3	3, 4, 5, 8, 9, 12,13	3,601

Table 3.3: Performance accuracy of each instructor individually

Algorithm	Performance accuracy for instructor 1	Performance accuracy for instructor 2	Performance accuracy for instructor 3
J48 DT	85.4%	85.7%	82.8%
NB	85.5%	86.8%	82.0%
MLP	86.2%	87.4%	82.8%
SMO	87.0%	85.4%	83.0%

On the other hand, the results show that the performance prediction accuracy increased when the worst ranked attributes are removed compared to the case when the algorithms run on the dataset with all attributes. Table 3.4 indicates the performance prediction accuracy of instructors for attributes that have the highest impact on dataset after removing the worst attributes. It can be noticed from the results that the performance prediction accuracies of all algorithms in Table 3.4 are better than the accuracies obtained by these algorithms with all attributes that are presented in Table 3.3 except for MLP and SMO that perform well on instructor 1 dataset in Table 3.3 whereas their performance prediction accuracy degrades when run on instructor 1 dataset after removing worst attributes as show in Table 3.4.

Table 3.4: Performance accuracy of instructors for attributes that have the highest impact on dataset

Algorithm	Performance accuracy for instructor 1	Performance accuracy for instructor 2	Performance accuracy for instructor 3
J48 DT	85.6%	86.4%	83.0%
NB	85.9%	87.3%	82.8%
MLP	85.6%	87.8%	83.5%
SMO	85.2%	86.4%	83.8%

Chapter 4

DECISION TREE ALGORITHM

4.1 Overview of Decision Trees

The DT is powerful, easy to understand and interpret, and produces efficient results, so that until nowadays being researched actively. The main idea of the DT is to generalize known structure to predict unknown input instances by learning simple decision rules from previous trained instances [52]. The DT consists of root node, internal nodes, and leaves or terminal nodes, where internal nodes denote to test nodes, while the branches are the outcome of test nodes, and leaf nodes hold the class labels.

The DT algorithms adopted greedy strategy to build the tree by producing a series of optimal decisions regarding which attribute to be selected for data partitioning. The training data is partitioned into successive small subsets in recursive way to grow the DT. The classification is performed by navigating the tree in top down order starting from the tree root down to the tree leaf based on the outcome of the tests along the path. The DT algorithm takes the training data as input that is known as data partition, the attribute list that contains the candidate attributes, and the attribute selection method to determine the optimal attribute to split the node of the tree.

At each recursive step, an attribute test condition is selected to partition the tree to small subsets during the tree growing process. Further a method for determining a test condition for the various attribute type is assigned by the DT algorithm followed by a specific measurement that employed to evaluate the goodness of each test condition. To terminate the tree growing process stopping condition is required. One of the most common strategies is to continue building the tree testing nodes until either all examples assigned to the same class, or all examples received identical attribute values. Hence, the attribute selection method partitioned the data into two or more sub-spaces according to some specific criteria [53]. Then the attribute with the highest score value is chosen as a root node for the tree followed by the next selected child nodes.

4.2 Attribute Selection Measures

Choosing the best attribute selection method is challenging and depends on the attribute value type. During the attribute selection process, the irrelevant and redundant attributes are removed. Thus, only the attributes that provide significant contribution are selected to participate in classification process. Before splitting starts, the measurement method that is used to select the best split attributes is defined in terms of reduction of impurity for parent and child nodes [54]. The better splitting attribute is the attribute with the larger reduction of impurity. In the literature, several attribute selection methods have been introduced such as entropy, gain ratio, gini index, etc. [55]. Let $p(i|t)$ be the relative frequency of class i at node t. The details of these impurity measures are as following:

Entropy

Entropy is used to calculate the homogeneity of a sample as in [54]:

$$Entropy(t) = -\sum_{i} p(i|t) log_2 p(i|t) \tag{4.1}$$

Information Gain

In information gain, the examples are partitioned based on a target attribute so as to reduce the entropy, and is measured as in [54]:

$$Gain_{split} = Entropy(p) - \sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \tag{4.2}$$

where parent node p with n records is split into k partitions and n_i is the number of records in partition i.

Gain Ratio

The gain ratio normalizes the information gain as follows in [54]:

$$Gain_{ratio} = \frac{Gain_{split}}{SplitInfo} \tag{4.3}$$

where,

$$SplitInfo = -\sum_{i=1}^{k} \frac{n_i}{n} \log_2 \frac{n_i}{n} \tag{4.4}$$

Gini Index

Gini index is an impurity-based criterion that measures the divergences between the probability distributions of the target attribute's values, and is defined at a node t as in [54]:

$$Gini(t) = 1 - \sum_{i} p(i|t)^2 \tag{4.5}$$

Classification Error

Classification error at a node t is defined as in [54]:

$$Error(t) = 1 - \max\{p(i \mid t)\} \quad (4.6)$$

4.3 Characteristics of Decision Trees

The DTs have several powerful characteristics. Decision process in the DT is similar to human decision process so that they are easier to understand and relatively easier to interpret. Further, the DTs are nonparametric to build classification models, and the problem of finding the optimal DT is NP-complete problem. The DTs are capable to manage large volume data and can construct models quickly even if the size of the data is very huge. Techniques used to construct the DTs are computationally inexpensive so after the tree is constructed by the training data, testing process is performed very fast with a worst-case complexity of $O(w)$ where w is the maximum depth of the tree. Whereas, the DTs can handle both discrete and continuous features which are difficult to be handling by most of DM techniques, the DT is affected by specific type of Boolean attributes that contain odd/even number whose value is 0 or 1. Moreover, the DTs are robust to handle redundant attributes. If two redundant attributes encountered during induction process, only one of the two redundant attributes would be used for the tree splitting. Feature selection method can be used to eliminate irrelevant attributes if the dataset includes some irrelevant attributes. The DT algorithms are induced by employing top-down approach that constructs the tree from a given dataset by partitioning the dataset recursively starting from root node to leaf node where the number of instances becomes smaller. The DTs are adversely affected by noisy data so that some methods for avoiding over-fitting are applied to prune the unwanted meaningless branches that are created during the tree grown process.

4.4 Decision Trees Induction

The DT induction is an inference technique that is most widely used in DM world [56]. Generally, the DT induction is based on Hunt's concept learning system. Quinlan developed this concept to produced ID3 method [57]. The DT belongs to a top down induction family where the algorithms build the tree by starting from the root and continue to the leaf. During this process, the internal nodes of the tree are selected recursively based on attribute selection approach.The tree growth is terminated when some stopping criteria is encountered. This way of forming the DT is known as greedy approach. The most implemented DT inducers are ID3, C4.5, C5.0, and CART. Here are some details about these methods:

4.4.1 ID3

ID3 stands for Iterative Dichotomize and was introduced by Quinlan 1986. It is considered as the very simple DT algorithm [57]. On the other hand, ID3 method used information gain approach as the splitting criterion and is neither capable to handle missing values and numeric data nor it applies any pruning algorithm. The ID3 stops tree growing when all training instances belong to the same class so there is no need for more division, or if the best information gain does not exceed zero.

4.4.2 C4.5

C4.5 is a developed ID3 introduced by Quinlan in 1993 [58]. Gain ratio is used as splitting criteria instead of information gain. C4.5 is capable to handle continuous and numeric attributes and can deal with training data that includes missing values by applying some corrections. Post-pruning approach is performed in C4.5 by means error-based pruning approach. The tree growing stops when the following conditions are satisfied:

- All training instances belong to the same class.

- No better information gain is committed by any one of remaining attributes.
- No more training examples is available at the node.

4.4.3 CART

Classification and Regression Trees (CART) method was introduced by Breiman in 1984 [59]. The tree is grown in CART recursively from root at the top to the leaves down by adopting greedy algorithm. The split process in CART is based on twoing criteria [60] rather than gini index that is used in the first version. Further, this method induced the tree estimating of the misclassification costs so that the tree is pruned by cost–complexity pruning. CART is capable to generate regression trees that predict real number at their leaves, so in regression case CART tends to split that minimizes the prediction squared error, while the prediction in each leaf depends on the weighted mean of the node.

4.4.4 C5.0

C5.0 is the algorithm developed as successor version of C4.5 presented by Quinlan in 2004 [61]. C5.0 builds the DTs as C4.5 method in atop down recursive way by following greedy algorithm manner. In C5.0, the induction is performed by using gain ratio and information gain splitting criteria approaches. Hence, the attribute with the highest information gain is selected to make the decision, while post-pruning methods are employed to overcome over-fitting problem. C5.0 is much faster than C4.5 and more efficient in consider of computational aspect and memory usage [62].Here are some aspects of C5.0 algorithm:

- It views large DTs as a set of rules, so it becomes easier to understand.
- It handles noisy and missing value data.
- It has the ability to specify the attribute as relevant or not during classification.

4.5 Advantages and Disadvantages of Decision Trees

The DTs as a classification method have many advantages that have been figured out in the literature such as:

- The DTs are self-explanatory and easy to understand even if the tree is complex. Moreover, the DT is considered to be comprehensible and trees can be converted to a set of rules.
- The DT has the ability to handle both nominal and numeric attributes.
- The DTs are capable to handle datasets with missing values.
- The DTs are capable to handle datasets that contain errors.
- The DTs are nonparametric methods. They do not have any assumptions regarding the space distribution and the classifier structure.
- The DTs are capable handle heterogeneous and high dimensional data.

The DTs also have some disadvantages which are as follows:

- Most of the DT algorithms prefer target attributes with discrete values only.
- The DTs are constructed by divide and conquer manner. Some subtrees can be replicated many times during induction. The same subtrees cannot be represented as one in the DT because every path is mutually exclusive, so the trees will be replicated.
- The greedy characteristics of the DTs cause sensitivity to training data with noise and irrelevant attributes.

4.6 Over-fitting Models in Decision Tree

During the classification process of the DT, it happens that some branches of the DT may contain noise or outliers in the training data so that the classifier produces a complex tree with unwanted meaningless branches that are difficult to understand. This phenomenon is known as over-fitting problem that occurs due to the overmatch of the training data. On the other hand, the DT model may produce a tree with very small size while the error rate of the training and testing sets is too large. This phenomenon is so called under-fitting problem. Over-fitting problem increases the tree structure complexity, reduces the precision of the unknown data instance, reduces the performance accuracy, and reduces the computational efficiency as well. Several methods have been introduced to overcome over-fitting problem, but one of the most effective and realizable approach is the pruning method. Some factors that cause over-fitting problem are discussed below:

4.6.1 Over-fitting Due to Noisy Data

In general, real world datasets are vulnerable to noise due to human error, faulty measuring devices, and the like. Thus, since the performance of predicting models depends on the data quality, a classifier that builds from this noisy data mostly produces a lower predictive performance. So, handling noise in machine learning models is a vast task in order to obtain effective models. Noisy data caused the so-called over-fitting problem that leads classifiers to generate large trees with high complexity and degrade performance. One of the eventual ways to cope with noise is to avoid over-fitting problem that helps classifiers to generate less complex trees [63]. Removing noisy instances from the dataset according to certain evaluation mechanisms is another approach that can tackle the over-fitting problem.

4.6.2 Over-fitting Due to Lake Representation of Sample

Over-fitting may sometimes occur due to lake representative [64] of samples in the training data. The DT algorithm mostly produces incorrect prediction when the training data contains small records without enough representatives of samples. So, insufficient of training records in the dataset leads to predicting test records by using another training record which is irrelevant to the classification purpose.

4.7 Estimating Generalization Errors

In general, it is not sufficient to consider the prediction error rate that is obtained from the training set since the learning algorithm has not yet been trained on records that it has never seen before. Therefore, the best alternative is to estimate the generalization error of the tree in order to reduce the complexity. Indeed, such models with less complexity generate lower generalization errors. For more details some methods for generalization error estimation are given below:

4.7.1 Re-substitution Estimate

Re-substitution error estimate method is based on the assumption that the training dataset is sufficient to represent the whole data comprehensively. Accordingly, the error obtained during the DT induction is called re-substitution error. Thus, this estimated error considers being an optimistic generalization error, although it is usually known that the error obtained from training data is insufficient to represent generalization error.

4.7.2 Pessimistic Error Estimation

In this approach, the generalization error is computed by calculating the sum of training error and the penalty of complexity for each node, and the estimated generalization error is considered to be pessimistic error of the DT.

4.7.3 Minimum Description Length

The Minimum Description Length (MDL) is based on the idea of Occam’s razor that states one should prefer the simplest hypotheses among various compatible hypotheses. Therefore, MDL method selects the model with shortest description of the data when the model complexity is computed. The MDL principle assumes that the evaluation of a model should be based on the data instead of the model closeness to the true model. The MDL principle provides immune methods that are capable to avoid over-fitting problem.

4.8 Over-fitting Handling Methods in Decision Tree

Pruning methods usually applied to reduce the tree complexity by removing the non-productive branches from the tree, hence this improves the performance accuracy and reduces the tree complexity. Pruning task includes two methods: post-pruning method, and pre-pruning method. Each of the two methods has advantages and drawbacks, but the main difference is the time of implementation, and the methods of performing pruning process. Pruning is performed during the DT induction for pre-pruning method, or after induction process terminated for post-pruning according some certain criteria. These criteria involve the estimation of error rate committed by the DT classifier before and after each possible prune.

Post-pruning is implemented after the DT induced and the full tree is grown. In post-pruning, the data is divided into two sets as training set and test set. The training set is used to grow the tree while the test set is used to prune the tree by testing the classifier on unseen data. Then, the DT traverses the tree in bottom up fashion comparing the error rate committed before and after pruning in order to decide whether the subtree should be pruned or retained. The error rate for each child node is estimated and compared to the total error of the parent node, and then the subtree is pruned if the comparison dictates that pruning is useful at the given node. Thus, when the parent node is pruned this node is converted to a leaf node.

Pre-pruning or online-pruning is implemented during the tree building process and the DT algorithm tries to stop the process when over-fitting is encountered by applying some stopping threshold conditions. In pre-pruning, the tree is navigated in top-down order, so it differs from post-pruning. Pre-pruning involves terminating condition that determines it as desirable to stop pruning process during the tree generation.

Furthermore, another way to combat over-fitting problem is to apply feature selection methods. Over-fitting can be reduced if at each node the best feature subspace is selected and the features with small contribution to the performance accuracy are removed. Hence, avoiding these features may lead to the increase of generalization ability and reduce node complexity.

4.9 Decision Tree Performance Evaluation Measurements

Evaluation of the DT performance is a vital task that provides answer to the question of which model is the best. So, when the DT is produced by multiple models for the same dataset, performance measures are exploited to select the preferable model. The process of predicting performance of different machine learning methods on a given dataset is not easy task as it sounds. Classification algorithms are evaluated to prefer one over other for a given dataset by using various performance notations such as accuracy, cost, speed, scalability, sensitivity, specificity, etc.

The aim of this section is to discuss the most common and effective measures that are used to evaluate the DT performance. In fact, the DT metrics that are used to measure the performance have various significances. Mostly, the performance is measured with respect to the number of correctly classified instances, speed, and sometimes by the tree size. On the hand, it is sufficient to measure the performance of a model in term of error rate. The classifier is firstly trained with the training data to predict the class of each example. If the prediction is correct, this is considered as success otherwise it is considered as error or failure. Hence, the rate of the error is estimated over the whole set of instances after the classifier is trained on an unseen data called test data in order to measure the classifier overall performance. Error rate on the training set is not sufficient to predict the future performance because the classifier has already been learned from this same data. The performance metrics that are used to evaluate the performance and their definitions are given below.

4.9.1 Accuracy

Accuracy is considered to be one of the best preferable DT performance evaluation methods. Accuracy is estimated in order to test and verify if the produced model is suitable for future predictions. Accuracy is the number of correctly classified instances for specific class divided by the total number of instances in that class. Accuracy measurement is based on the confusion matrix where the columns represent the actual classes and the rows represent the predicted classes. Confusion matrix indicates the number of instances of specific class that are misclassified belonging to another class. The diagonal of this matrix indicates the number of instances that are correctly classified.

True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN) rates are used to compare the results of the classifier under test with respect to specific judgment. While the terms positive and negative are used to represent the prediction values of the classifier, the terms true and false are used to measure how the prediction correspond to that specific judgment. These notations can be discussed in detail as below:

a. **Accuracy:** Accuracy is measured as:

   Accuracy = (TP+TN) / (TP+TN+FP+FN)

b. **True Positives Rate:** TP rate that is also known as recall represents the proportion of relevant instances (positive) that have been retrieved over the total amount of relevant instances.

   TP Rate = TP / (TP + FN)

c. **True Negative Rate:** TN rate represents the correctly negative instances that are labeled by the classifier.

   TN Rate = TN / (TN + FP)

d. **False Positive Rate:** FP rate represents the negative instances that are incorrectly classified as positive by the classifier.

   FP Rate = FP / (FP + TN)

e. **False Negative Rate:** FN rate represents the positive instances that are misclassified as negative by the classifier.

   FN Rate = FN / (FN + TP)

f. **Precision:** Precision is used to measure how many instances that are classified as positive class are truly positive. It is the ratio of instances that are correctly predicted as positive.

   Precision = TP / (TP + FP)

g. **Area under Receiver Operating Characteristic (ROC):** ROC curve is generated by plotting TP rate against FP rate at some various threshold settings.

Consequently, the rest of this section discusses some other estimating methods that are widely used. These methods are based on accuracy metrics and error rates and are also important for selecting the best classifier with respect to the performance accuracy.

4.9.2 Holdout

In holdout method, the dataset is partitioned randomly into two disjoint subsets, called training set and testing set. The training set is used to train the classifier in order to build the model, while the test set is used for testing the classifier to evaluate the performance accuracy. In such cases, one-third of the data is holding out for testing and the remaining two-thirds are used for training. Hold-out method can be represented in various format based on the percentages of data that held-out for the test, such as 10% holdout validation, 20% hold-out validation, 50% holdout validation and so on. However, in some cases small hold out percentage like10% holdout validation may expose over-fitting problem [65].

The main idea of this method is to split the dataset into two parts assuming that they are from the same type. So, this method suffers from some limitation. Firstly, the test set data might not fully be similar to the training data so that the produced model may not be sufficient as desired. Secondly, small training set size may lead to inconsistent model. On the other hand, larger training set may lead to less efficient model because small test set is used to estimate the performance accuracy. Thirdly, since the training set and test set are derived from the original data, some classes may overrepresented in one set and underrepresented in the other set. Finally, random sampling is proven to be non-sufficient for building reliable models. Stratification approach is used to ensure that the random sampling is done properly, and the number of classes is distributed well in both datasets. However, in general stratification provides only a safety safeguard for the sampling distribution but does not fully remedy the issue [66].

4.9.3 Cross Validation

Cross validation method is used when the size of the data is limited where it is risky to partition the data into two sets only. In this approach, the data is partitioned into two equal size subsets. Each example in the data set is used for training for same certain number of time and once for testing [67]. For instance, if the data is partitioned into two equal size sets, and one of these sets is selected for training and the other set for testing, then the role of the sets are swapped by choosing the previous training as test set and that test set as training set.

The total error is estimated by summing up the error of both sets. This approach is called two-fold cross validation. While in *k*-fold cross validation the data is equally split into *k* equal partitions, then the classifier is trained and tested on these *k*-fold partitions. In each iteration, one of these partitions is used for testing and the rest for training. This process is repeated *k* times. Hence, in this approach each partition is used exactly once for testing and *k*-1 for training. The accuracy rate of each iteration is computed, and then total accuracy is calculated by summing up these accuracies and averaged to get the total model accuracy.

Moreover, such as in holdout method, *k*-fold cross validation is utilized as a stratified approach to balance the distribution of folds since the cross-validation accuracy depends on the distribution of the folds [68]. Several studies have been made by applying various learning methods on different dataset to specify the suitable number of fold to be used in *k*-fold cross validation method. It was found that 10-fold cross validation produced better results compared to those cross validations with different assigned fold number [69]. One of the *k*-fold cross validation disadvantages is the increase of training set and decrease of test set when the chosen *k*- fold is great.

4.9.4 Leave-one-out Method

Leave-one-out cross-validation is simply special case of *k*-fold cross validation and has the same principal of *k*-fold cross validation, but the only difference is that, each test set contains only one example, and the *k* value is equal to the total number of examples in the dataset. So, during the execution, each example is left out once in each turn and the classifier is trained on the remaining examples. Thus, every example is used once for testing during the model building [70].

One of the main drawbacks of this method is that it is computationally very costly to repeat the procedure *N* times, and also the variance of the estimated performance is very high because each test set contains only one example. On the other hand, one of the important advantages of the leave-one-out approach is that it offers a chance to utilize as much as possible data for training.

Chapter 5

DECISION TREE POST-PRUNING

5.1 Overview of Post-Pruning Technique

The DT is one of the most powerful and efficient techniques in DT which has been widely used by researchers [71, 72]. Compared to the other classification techniques the DT is faster and provides better accuracy. During the data classification process, some branches of the DT may contain noise or outliers in the training data and this result in a complex tree which is difficult to understand. Therefore, pruning techniques are applied in order to remove those unwanted branches in order to improve the prediction accuracy.

Pruning is based on the notation of Ockham's razor rule that is stated as “The more assumption you have to select is the one with more unlikely explanation among various assumptions” [73]. This assumption is very effective for DT over-fitting. During DT induction, the learning algorithm produces a model that is too bias on noisy data. The error obtained from the training data is not sufficient to measure the accuracy so that the model is tested on unseen data before in order to prune the tree with the aim of maximizing the accuracy. Two main pruning strategies have been developed to avoid over-fitting namely pre-pruning and post-pruning as mentioned in the previous chapters. The aim of this chapter and the rest of the thesis is to investigate post-pruning methods and attempt to develop some post-pruning algorithms to combat the over-fitting problem.

Post-pruning approach consists of two phases as growing phase and pruning phase. In the first phase, the tree is fully grown to its maximum size by using the training dataset, while in the second phase, the tree is post pruned by using the test data set. In post-pruning method, the pruning is performed by traversing the tree in bottom-up order replacing the subtree with new leaf node or pruning the subtree and raising another subtree that produces better accuracy according to specific condition based on the post-pruning algorithm utilize [74, 75].In most of post-pruning methods, the pruning process is achieved by comparing the error rate yielded when the DT learning method is trained by the training set and the errors of the testing set.

In the past decades, several post-pruning algorithms have been introduced such as REP, error complexity pruning, MEP, and cost-based pruning. Amongst these methods, REP is the best method that produces the smallest tree with better accuracy. Most of pruning methods such as REP and MEP traverse the DT in bottom-up order estimating the misclassification errors for each node so as to reduce the tree size and to avoid the over-fitting problem. The following subsection discuses some most used popular post-pruning methods.

5.1.1 Cost–Complexity Pruning

Cost–complexity pruning was developed by Breiman in 1984 [76], to be used in CART DT. Cost-complexity pruning is a two-stage pruning method. In the first stage, a sequence of trees T_0, T_1,...,T_k are generated from the training dataset by the learning algorithm, where T_0 represents the original tree before pruning and T_k denotes to the tree root. The subtrees with the smallest error rate increase from the training data are pruned. This small increase amount of error rate in the training data is measured by α metric, so the subtrees that obtain the smaller value of $\alpha$is pruned. In the second stage, the pruning is performed based on the generalization error estimation for each one of these trees. The generalization error is estimated by utilizing cross-validation approach or hold out approach, and then the tree with the least generalization error is chosen as the optimal tree among the sequence of trees.

5.1.2 Reduced Error Pruning

REP is a simple method that was suggested by Quinlan in 1987 [77].It traverses the fully produced tree model in the bottom-up fashion. In this method, each internal node that produces large error rate than its parent node is pruned and replaced by its following leaf node, otherwise, it is kept when the error rate of parent is greater than the internal node. The process stops after the whole tree is traversed from the most left bottom nodes to the root node. The studies have shown that this method produced smaller tree compared to other post-pruning methods.

5.1.3 Minimum Error Pruning

MEP was developed by Niblett and Bratko in 1987 [78]. MEP traverses the tree in bottom-up order similarly to REP. During this process, the error rate is computed for each expected internal node before the subtree is pruned or either if it is retained. A subtree is replaced by a node if the expected error rate of the former is no smaller than the later. The pruning process continues recursively until the full tree is pruned. The disadvantage of this algorithm is the un-reality of the practical part since it considers all classes equal likely.

5.1.4 Pessimistic Pruning

Pessimistic pruning is another pruning method that was introduced by Quinlan in 1987 [79]. The main characteristic of this algorithm is that it does not require a separate test for pruning. It just depends on the error that is estimated from the training data. Further, pessimistic pruning method assumes that half of examples that are gathered at leaf nodes of a subtree mostly be incorrectly classified, and then this fraction is divided by the total number of examples in that leaf. Thus, this fraction is obtained by employing the continuity correction in binomial distribution [80]. Pessimistic approach traverses the tree in top down order instead of bottom-up order. It navigates the internal nodes recursively and prunes an internal node when its estimated misclassification error rate + ½ is not exceed one standard error rate of previous estimated misclassification error rate for a subtree. The advantage of pessimistic method is that it does not need to divide the dataset into two sets for training and testing. Only the training data is considered. It is also faster than other post-pruning methods such as cost-complexity pruning and REP since only one tree is produced instead of producing a sequence of trees to select one of them.

5.2 Proposed Methods

In this thesis, we adopt post-pruning approach to combat the over-fitting problem that rises during data classification process. To do so, two new post-pruning techniques are developed which are namely Pruning with Bayes Minimum Risk (PBMR) and Zero-One Loss Function Pruning (ZOLFP). Our aim is to achieve small tree with high prediction accuracy. In the remaining of the section, the notation of these two proposed post-pruning methods will be discussed with running examples and both of them will be trained on real world datasets to examine their efficiency compared to some selected well-known post-pruning methods in the art.

5.2.1 Pruning with Bayes Minimum Risk

5.2.1.1 Bayes Minimum Risk Notation

As defined in [81, 82], Bayes minimum risk classifier is a decision model that performs quantifying trade-offs among different decisions by utilizing probabilities and the cost that are related to such decisions. The method suggested in this thesis considers a post-pruning approach that estimates the risk-rate for the parent node of the subtree and its leaves. The risk associated for each node denoted as R_k is computed as:

$$R_k(a_i|x) = \sum_{j=1, j\neq i}^{T_c} L_k(a_i|C_j) p_k(C_j|x) \tag{5.1}$$

Where $L_k(a_i|C_j)$ and $p_k(C_j|x)$ are the loss function when an example a_i is predicted in class C_j while true class is C_i and the estimated probability of an example belonging to C_j , respectively and T_c is the total number of classes. The total risk of the leaves can be calculated as:

$$R_l = \sum_{m=1}^{T_l} R_m(a_i|x) \tag{5.2}$$

Where T_l is the total number of leaves under the subtree.

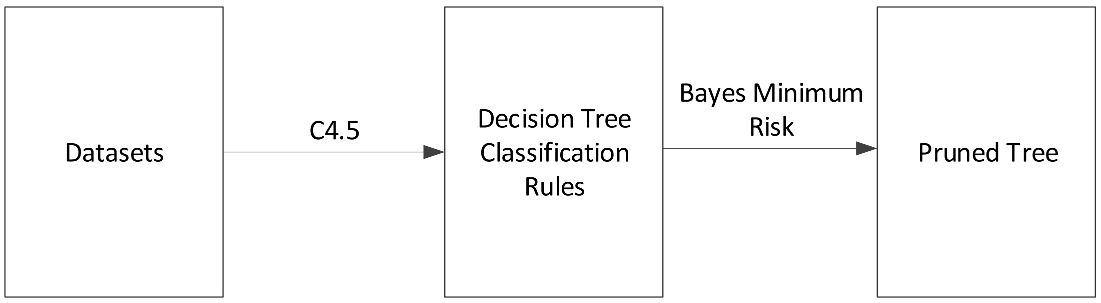


Figure 5.1: The principle of PBMR.

5.2.1.2 Pruning with Bayes Minimum Risk Algorithm

This section discusses the idea of PBMR algorithm published in [83]. In this proposed algorithm, a DT algorithm is used to build and initiate a tree model. Then, linear regression method is applied to build models on leaves level of the tree. The proposed modified DT algorithm is implemented recursively with the following sequence until the tree is formed. Proposed algorithm adopts a post-pruning bottom-up method for C4.5 DT algorithm using Bayes minimum error method that estimates risk rates instead of estimating the misclassification error as illustrated in Fig. 5.1. Moreover, the flowchart in Fig.5.2 indicates the structure of the proposed algorithm and way follow to proceed.

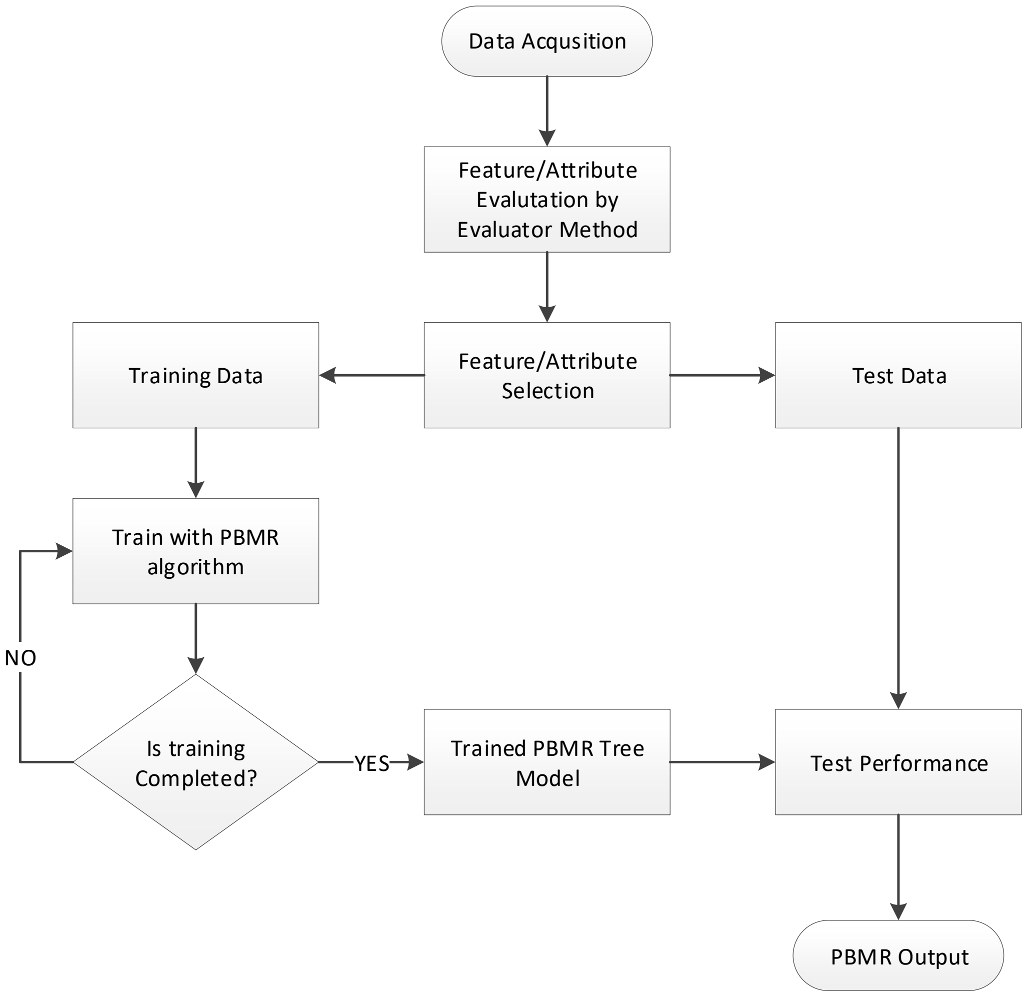


Figure 5.2: The flowchart of the PBMR training and testing.

When the DT learning algorithm finishes building the tree model, the proposed algorithm given in Fig. 5.3 traverses the tree model in bottom-up fashion to compute the risk rates of the parent node of the subtree (R_p) and the leaf nodes (R_l) as in (5.1) and (5.2), respectively. The parent node is converted to a leaf node if the risk-rate of the parent is less than the risk-rate of its leaves ($R_p < R_l$), otherwise, the parent node is retained. The process is repeated for all parents of leaves until the tree is optimized. To clarify the notation, the new method is illustrated through a simple example. A simple DT example is given in Table 5.1.

```
Input: Dataset
Output: Post-pruned DT with Bayes minimum risk function
1.  Rank attributes
    Rank attributes at the dataset according to their significance level and select
    the most important attributes improving the performance accuracy
2.  Generate initial DT based on C4.5
3.  For each node
4.      If node is a parent node then
5.      Compute the parent risk rate ($R_p$) and the risk-rate of its leaves ($R_l$)
6.          If $R_p < R_l$ then
7.              Convert parent node to a leaf
8.          Else
9.              Retain the parent node
10.         End if
11.     End if
12. End for
13. Return the final tree
```

Figure 5.3: The algorithm of PBMR.

Fig.5.4 shows how the newly introduced method is applied to perform pruning operation on the given DT. In this figure an un-pruned DT with two classes is presented, where each node is labeled by the node number and the risk-rate associated with that node.

The proposed method traverses the tree in a bottom-up fashion converting a node to a leaf if the risk-rate of the leaves is greater than the risk-rate of the node. To perform this task, the pruning method traverses the tree from left to right in bottom-up order. So that, in the first step the pruning method starts from the most left branch which is node 3 in our case as shown in Fig.5.4 (a).

Table 5.1: Example of a simple DT

a	***b***	***c***	***class***
1	0	1	Yes
1	0	0	No
1	1	0	No
1	1	1	Yes
0	1	0	Yes
0	0	0	Yes
0	1	1	No
0	0	1	No

Because the risk-rate of subtrees of node 3 (2) exceeds node 3 risk-rate (1), these subtrees are removed and node 3 becomes a leaf node given as in Fig.5.4 (b). In the second step, the pruning method traverses nodes 6 converting it to a leaf since the risk-rate of its subtrees (1) is greater than the risk-rate of node 6 itself (0), as shown in Fig.5.4 (c). Then, in step three, node2 is traversed after both of its successors are removed since the subtrees of node 2has lower risk-rate (1) than node 2 itself (2), the subtrees are retained. In the last step, the risk rates of the subtrees attached to node 0 (1) is less than the risk-rate of node 9 itself (2), so that the subtrees of node 9 are also retained for the same reason.

5.2.2 Experiments of Proposed Methods

5.2.2.1 Experiment Data

The experiment datasets are collected from UCI machine learning repository [84]. In this thesis, six different datasets have been used to evaluate the proposed methods. Table 5.2 presents the number of instances, the number of classes, and the number of attributes for the datasets.

5.2.2.2 Experiment Process

C4.5 DT algorithm is exploited to create the tree and then the pruning process is performed by using the proposed PBMR methods. The proposed PBMR method is compared with two other post-pruning methods namely, REP and MEP.

5.2.2.3 Experiment Results and Discussion

Experiments are conducted by using java eclipse combined with Weka. It is known that attribute evaluator techniques can be applied to select the attributes that have the greatest impact on the dataset. Removing the worst ranked attributes that have lower importance on the dataset usually increases the accuracy of the algorithms. In this context, Weka's attribute evaluator techniques namely OneR and Information Gain (InfoGainAttributeEval) are employed to select the attributes with high impact on the datasets and remove the worst attributes that are shown in Table 5.3.

Table 5.2: Datasets description.

Datasets	Number of instances	Number of attributes	Number of classes
Car	1,728	7	4
Diabetes	768	8	2
Labour	57	16	2
Blogger	100	6	2
User Knowledge	258	6	4
Flare Solar	1,066	13	3

After the worst attributes are removed with the attribute evaluators, the accuracies obtained by PBMR with 10-fold cross-validation and by dividing the datasets into two sets as training and test are compared in Table 5.4. For 10-fold cross validation, datasets are partitioned into 10 subsets of equal size and each subset is employed for testing and the rest for training.

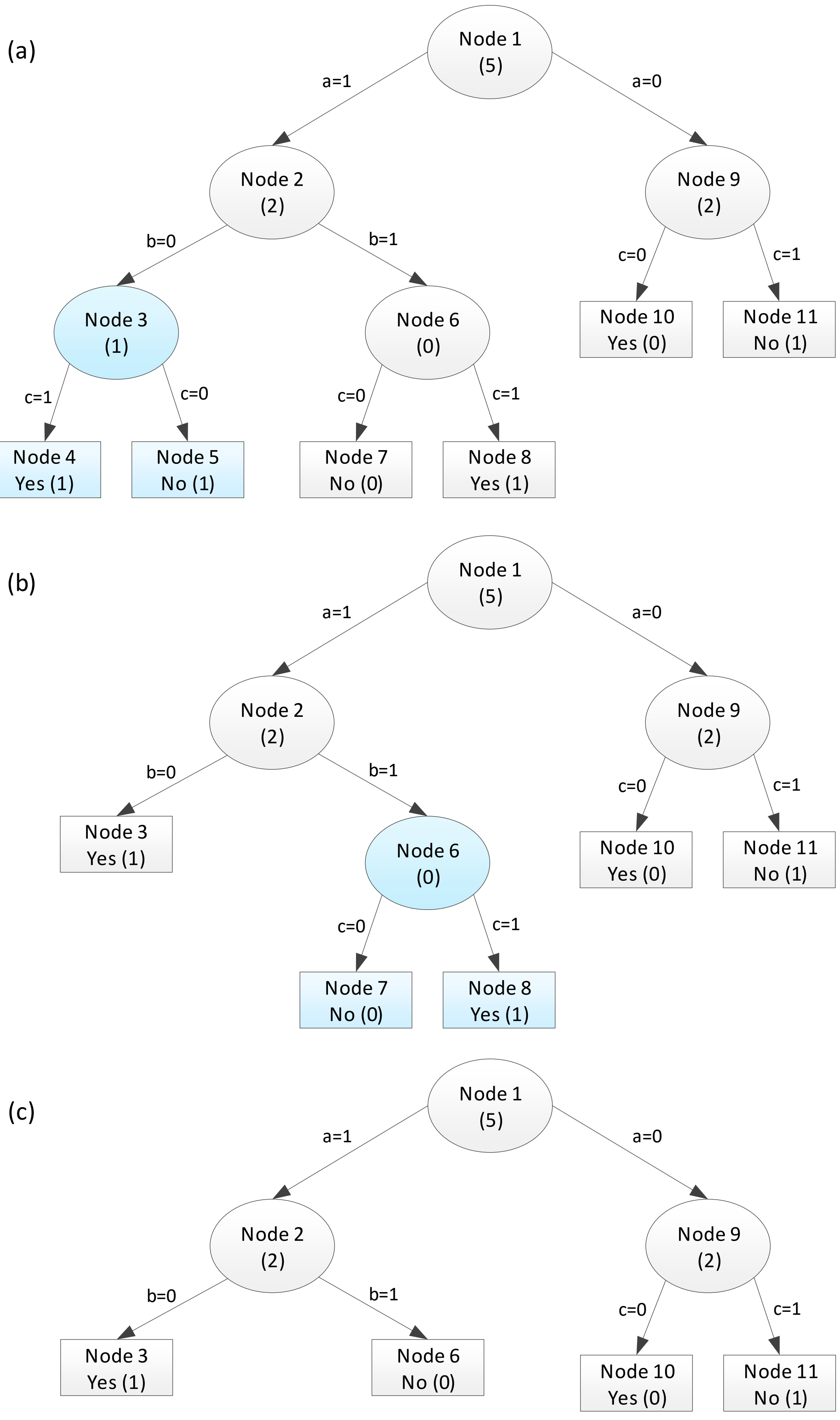


Figure 5.4: A simple DT example for PBMR method

Table 5.3: Attributes removed by OneR and InfoGainAttributeEval attribute evaluators

Datasets	OneR	InfoGainAttributeEval
Car	perspons, buying	lug-boot, doors
Diabetes	mass, pedi, skin	pres, pedi, skin
Labour	wage2.wage, shift_diff, dur, hours. hrs, wage3.wage	education-allowance, standby-pay, working-hours, duration
Blogger	lmt, lpss	lmt, lpss
User Knowledge	STG, SCG	STG, SCG
Flare Solar	Previous 24 hours activity code, Historically, M-class production, Area of the largest spot, Area, Did region become historically	Historically, C-class production, Evolution, Did region become historically, Area of the largest spot

Additionally, the same datasets are divided into two sets as training and test sets. 60% of each dataset is randomly selected as training set and 40% as testing set based on holdout approach with OneR attribute evaluator.

It is clear from the results that the case when datasets are divided into two as training and testing sets obtained better performance for PBMR compared to the case when 10-fold cross-validation is used. The results also show that PBMR with 10-fold OneR attribute evaluator achieves better accuracies compared to PBMR with 10-fold InfoGainAttributeEval evaluator with three scores to the former and one scores for the latter, whereas both methods have the same sores in two cases. Because hold out method suffers from some limitation, it is not sufficient to give a reliable model so that cross-validation methods are used. Since 10-fold cross validation method with OneR attribute evaluator shows better performance, OneR attribute evaluator with 10-fold cross validation method is employed for the rest of the experiments.

Table 5.4: Accuracy of PBMR with OneR and InfoGainAttributeEval attribute evaluators.

Datasets	Accuracy (%)		
	60% training, 40% testing, holdout, OneR	10-fold, OneR	10-fold, InfoGainAttribute Eval
Car	97	96	87
Diabetes	80	76	73
Labour	90	75	74
Blogger	80	76	76
User Knowledge	93	83	83
Flare Solar	99	98	99

Table 5.5 shows the accuracy and the tree complexity in terms of tree size as the total number of nodes and leaves for PBMR, REP and MEP approaches. The results are also compared with the original un-pruned C4.5 DT algorithm (UDT-C4.5) to illustrate the effect of pruning. For all the datasets, the proposed PBMR method produces better accuracies. In terms of complexity, PBMR produces greater tree than REP and MER in Blogger and User Knowledge datasets and produces the same tree size with REP in Car, Diabetes, and Flare Solar datasets, although its performance in terms of accuracy is higher than the other methods.

As seen from the results, besides of having better accuracy performance, all pruning-based approaches produce smaller tree sizes as compared to DT-C4.5 in all the cases.

The next experiment includes the weighted average of precision/recall scores evaluations of the proposed method PBMR, REP, and MEP in Table 5.6. The results show that PBMR produces better precision/recall scores than REP and MEP in all datasets.

Table 5.5: Accuracy and tree size for PBMR, REP and MEP

Datasets	Algorithms	Accuracy (%)	Number of nodes	Number of leaves
Car	UDT-C4.5	94	186	134
	PBMR	96	182	129
	REP	94	182	129
	MEP	88	113	81
Diabetes	UDT-C4.5	73	43	22
	PBMR	75	41	21
	REP	74	41	21
	MEP	74	15	8
Labour	UDT-C4.5	72	47	39
	PBMR	77	27	23
	REP	65	13	8
	MEP	75	36	33
Blogger	UDT-C4.5	73	43	28
	PBMR	76	24	17
	REP	72	15	11
	MEP	75	12	9
User Knowledge	UDT-C4.5	75	404	399
	PBMR	76	400	398
	REP	75	81	80
	MEP	74	81	80
Flare Solar	UDT-C4.5	95	25	21
	PBMR	98	19	16
	REP	97	19	16
	MEP	96	25	21

The weighted average of TP rate, FP rate and area under ROC curve are also considered to measure the performance of the pruning methods as in Table5.7. The proposed method produces the highest TP rate and the lowest FP rates for all datasets. Moreover, the proposed PBMR method produces highest scores in terms of the area under ROC for four datasets.

Table 5.6: precision/recall scores for PBMR, REP, and MEP

Datasets	Algorithms	Precision (%)	Recall (%)
Car	UDT-C4.5	94	94
	PBMR	95	96
	REP	94	94
	MEP	89	88
Diabetes	UDT-C4.5	72	73
	PBMR	75	75
	REP	73	74
	MEP	73	74
Labour	UDT-C4.5	72	72
	PBMR	77	77
	REP	65	65
	MEP	74	75
Blogger	UDT-C4.5	72	73
	PBMR	75	76
	REP	70	72
	MEP	74	75
User Knowledge	UDT-C4.5	76	75
	PBMR	77	76
	REP	75	75
	MEP	75	74
Flare Solar	UDT-C4.5	95	95
	PBMR	98	98
	REP	97	97
	MEP	96	96

5.2.2 Loss Minimization Method

In the past decade, many algorithms have been introduced to minimize the misclassification error rate, while some researches attempted to minimize the expected loss or the cost of misclassifying an example in the dataset. Although, recent researches continued to develop more accurate algorithms to enhance the misclassification error rate, applications in business, medicine, and science have shown that real problems require more subtle measures of performance [85, 86]. Furthermore, in general, various kinds of errors have different costs.

Table 5.7: TP rate, FP rate, and area under ROC for PBMR, REP, and MEP methods

Datasets	Algorithms	TP rate (%)	FP rate (%)	Area under ROC (%)
Car	UDT-C4.5	94	3	96
	PBMR	96	2	98
	REP	94	3	98
	MEP	88	6	96
Diabetes	UDT-C4.5	73	31	74
	PBMR	75	32	77
	REP	74	34	74
	MEP	74	36	73
Labour	UDT-C4.5	72	36	72
	PBMR	77	22	79
	REP	65	29	82
	MEP	75	31	79
Blogger	UDT-C4.5	73	39	73
	PBMR	76	29	85
	REP	72	36	81
	MEP	75	36	73
User Knowledge	UDT-C4.5	75	10	89
	PBMR	76	6	95
	REP	75	7	94
	MEP	74	11	91
Flare Solar	UDT-C4.5	95	9	54
	PBMR	98	8	82
	REP	97	8	82
	MEP	96	9	31

The performance of classifiers is evaluated by estimating the error rate obtained from the dataset which represents the incorrect classified examples. It is sometimes suitable to evaluate classifiers by considering the cost of the misclassification errors based on loss matrix.

Several researchers have adopted the idea of incorporating the general loss matrix into C4.5 DT algorithm to reduce the misclassification cost or to minimize the misclassification loss [87, 88]. These algorithms introduced heuristic splitting methods that perform dual tasks. Initially, they specify the splitting that minimizes the loss and secondly, they select the proper split for subsequent splits. Normally, the loss matrices are T_c by T_c matrices where T_c is the number of classes, while the rows represent the predicted classes by the classifier algorithm and the columns represent the correct classes. The loss function $L(i|j)$ gives the "loss" of predicting class i when the true class is j, and diagonal elements, $L(i|i)$, are always zero [89]. In general, loss functions state exactly how each action costs, for instance, the loss function $L(a_i|C_j)$ indicates the loss obtained for taking action a_i when the class is C_j.

In this thesis, a new DT post-pruning method that is based on loss minimization is introduced. This method focuses on employing the Zero-One loss function in C4.5 DT to avoid the over-fitting problem and improve the performance. The expected loss or so-called the conditional risk of Zero-One loss function is defined by:

$$R_{0/1}(a_i|x) = \sum_{\substack{j=1 \\ j\neq i}}^{T_c} P(C_j|x) = 1 - P(C_i|x) \tag{5.3}$$

Where $P(C_j|x)$ is the probability that an example *x* belongs to a specific class *C_j*.

In this thesis, Laplace method is utilized to estimate the probability that an example belongs to class *i* in the specific node. If there are N_i examples of class *i* at a node and T_c classes, then the probability P_{lc} that an example at this node belongs to class *i* is estimated by [90]:

$$P_{lc} = (N_i + 1)/(T_c + N_T) \tag{5.4}$$

Where N_T is the total number of examples at that node.

So, when Laplace method is employed in (5.3) to estimate the probability that an example belongs to specific class in a node, the Zero-One loss is computed as follow:

$$R_{0/1} = 1 - P_{lc} \tag{5.5}$$

Proposed pruning method prunes the DT in the bottom-up fashion by adopting REP approach. However, instead of estimating the misclassification error rate, the expected loss is estimated by applying Zero-One loss function method. For that reason, the proposed method is named as Zero-One Loss Function Pruning. The Zero-One loss for each node is computed by applying (5.5). Then, the expected loss, L_{ex}, for each node is computed by multiplying the Zero-One loss by the loss function related to that node as:

$$L_{ex} = R_{0/1}L(i|j) = (1 - P_{lc})L(i|j) \tag{5.6}$$

The pruning process is performed by dividing the dataset into two sets as training set and pruning set. The training set is used to create the DT and then it is tested by using the pruning set. During the pruning, the expected loss for each tested node is computed. If the expected loss for the tested node is less than or equal to the sum of the expected loss for the whole subtree under that tested node, these tested nodes are converted to leaf nodes, and leaf nodes are labelled by the least expected loss class that is equivalent to label of the class with the majority class label in DT nodes. So that, the converted test nodes to leaf nodes are given the class label of the least expected loss class.

```
Input: Dataset
Output: Post-pruned DT with Zero-One Loss Function
1.  Rank attributes,
    Rank attributes at the dataset according to their significance level and
    select the most important attributes improving the performance accuracy
2.  Divide the dataset into two set 70% for training set and 30% for pruning set
3.  Use the training set to create a DT based on C4.5
4.  Use the pruning set to test and prune the DT
5.  For each node
6.  If node is a test (parent) node then
7.          Compute the expected loss for this test (parent) node ($L_{ex}^{p}$)
            and the expected loss of its subtree ($L_{ex}^{t}$)
8.          If ($L_{ex}^{p} \leq L_{ex}^{t}$) then
9.              Convert the test (parent) node to a leaf
10.         else
11.             Retain the test (parent) node
12.         End if
13. End if
14. End for
15. Return the final tree
```

Figure 5.5: The algorithm of ZOLFP.

5.2.2.1 Proposed Zero-One Loss Function Pruning Algorithm

The algorithm in Fig. 5.5 shows the steps followed for the proposed ZOLFP algorithm.

5.2.2.2 A Running Example

In this example, there are two classes for un-pruned DT, a person is classified as healthy or sick accordingly. The proposed method ZOLFP is examined to prune this tree based on loss (cost) approach. ZOLFP method computes the expected loss for each node by applying (5.6).The loss function for each node is indicated where *h* denotes healthy class and *s* denote sick class.

In Fig.5.6 (a), the subtree should not be pruned when ZOLFP method is applied because the parent loss is greater than the sum of losses of the children (10.4 versus 5.9). Fig.5.6 (b) shows the reverse situation. Here, ZOLFP method prunes the tree because the parent has lower loss than the sum of losses of the children (9.2 versus 14.2).Thus, the subtree is pruned.

5.2.2.3 Proposed ZOLFP Method Experiment

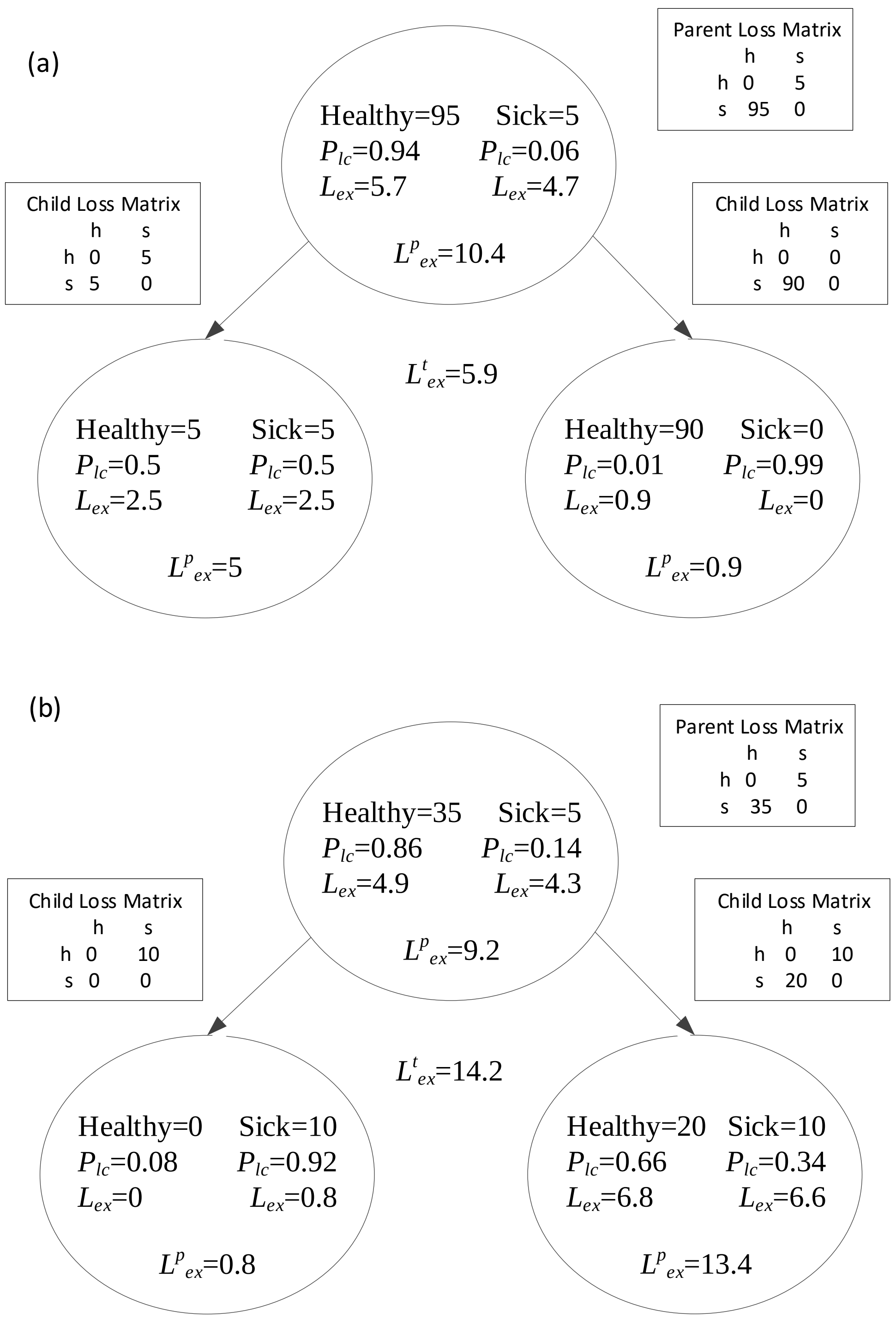
5.2.2.3.1 Experiment Process

The tree is created based on C4.5 DT algorithm and then, the pruning process is performed by using the newly introduced ZOLFP method as shown in Section 5.2.2.1. The experiment process is carried out by using java eclipse combined with Weka. The datasets of Table 5.2 are trained and tested with 10-fold cross validation by the proposed ZOLFP method, REP and MEP after the worst attributes are removed based on Table 5.3 results. OneR attribute evaluator with 10-fold cross validation method is employed in this experiment.

5.2.2.3.2 Experiment Result Analysis

Table 5.8 shows the accuracy and the tree complexity in terms of tree size as the total number of nodes and leaves for ZOLFP, REP and MEP approaches. The results are also compared with UDT-C4.5 algorithm to illustrate the effect of pruning. For all the datasets, the proposed ZOLFP method produces better accuracies compared to

REP and MEP methods. In terms of complexity, the proposed method ZOLFP produces smaller tree size than REP and MEP for three datasets.



(a)
Parent Loss Matrix h s h 0 5 s 95 0
Healthy=95 Sick=5
P_{lc}=0.94 P_{lc}=0.06
L_{ex}=5.7 L_{ex}=4.7
L^p_{ex}=10.4
Child Loss Matrix h s h 0 5 s 5 0
Child Loss Matrix h s h 0 0 s 90 0
L^t_{ex}=5.9
Healthy=5 Sick=5
P_{lc}=0.5 P_{lc}=0.5
L_{ex}=2.5 L_{ex}=2.5
L^p_{ex}=5
Healthy=90 Sick=0
P_{lc}=0.01 P_{lc}=0.99
L_{ex}=0.9 L_{ex}=0
L^p_{ex}=0.9
(b)
Parent Loss Matrix h s h 0 5 s 35 0
Healthy=35 Sick=5
P_{lc}=0.86 P_{lc}=0.14
L_{ex}=4.9 L_{ex}=4.3
L^p_{ex}=9.2
Child Loss Matrix h s h 0 10 s 0 0
Child Loss Matrix h s h 0 10 s 20 0
L^t_{ex}=14.2
Healthy=0 Sick=10
P_{lc}=0.08 P_{lc}=0.92
L_{ex}=0 L_{ex}=0.8
L^p_{ex}=0.8
Healthy=20 Sick=10
P_{lc}=0.66 P_{lc}=0.34
L_{ex}=6.8 L_{ex}=6.6
L^p_{ex}=13.4

Figure 5.6: A running example of ZOLFP method

Table 5.8: Performance and tree size for ZOLFP, REP and MEP methods

Datasets	Algorithms	Accuracy (%)	Number of Nodes	Number of Leaves
Car	UDT-C4.5	94	186	134
	ZOLFP	95	88	87
	REP	94	182	129
	MEP	88	113	81
Diabetes	UDT-C4.5	73	43	22
	ZOLFP	76	31	16
	REP	74	41	21
	MEP	74	15	8
Labor4	UDT-C4.5	72	47	39
	ZOLFP	77	36	30
	REP	65	13	8
	MEP	75	18	17
Blogger	UDT-C4.5	73	43	28
	ZOLFP	77	33	23
	REP	72	15	13
	MEP	75	12	9
User Knowledge	UDT-C4.5	75	404	399
	ZOLFP	77	402	399
	REP	75	81	80
	MEP	74	81	80
Flare Solar	UDT-C4.5	95	25	21
	ZOLFP	99	7	6
	REP	97	19	16
	MEP	96	25	21

For the dataset which the ZOLFP complexity is higher, the accuracy of ZOLFP is better than the accuracies of REP and MEP. On the other side, it is obvious that the proposed method ZOLFP produces smaller tree with better accuracy compared to UDT-C4.5 for all datasets.

Table 5.9: precision/recall scores for ZOLFP and REP methods

Datasets	Algorithms	Precision (%)	Recall (%)
Car	UDT-C4.5	94	94
	ZOLFP	95	95
	REP	94	94
	MEP	89	88
Diabetes	UDT-C4.5	72	73
	ZOLFP	76	76
	REP	75	74
	MEP	74	74
Labour	UDT-C4.5	72	72
	ZOLFP	76	77
	REP	65	65
	MEP	74	75
Blogger	UDT-C4.5	72	73
	ZOLFP	76	77
	REP	70	72
	MEP	74	75
User Knowledge	UDT-C4.5	76	75
	ZOLFP	77	77
	REP	75	75
	MEP	75	74
Flare Solar	UDT-C4.5	95	95
	ZOLFP	97	99
	REP	97	97
	MEP	96	96

Additionally, precision/recall scores are also investigated to evaluate the performance of ZOLFP, REP and MEP methods as shown in Table 5.9. For precision scores, ZOLFP produces better scores than REP and MEP in five datasets, and the same precision score with REP for Flare Solar dataset. While in terms of recall scores, ZOLFP produces better scores in all datasets.

Table 5.10: TP rate, FP rate, and area under ROC for ZOLFP, REP, and MEP methods

Datasets	Algorithms	TP rate (%)	FP rate (%)	Area under ROC (%)
Car	UDT-C4.5	94	3	96
	ZOLFP	95	2	98
	REP	94	3	98
	MEP	88	6	96
Diabetes	UDT-C4.5	73	31	74
	ZOLFP	76	32	83
	REP	74	29	81
	MEP	74	31	77
Labour	UDT-C4.5	72	36	72
	ZOLFP	77	27	84
	REP	65	29	82
	MEP	75	31	79
Blogger	UDT-C4.5	73	39	73
	ZOLFP	77	33	82
	REP	72	36	81
	MEP	75	36	73
User Knowledge	UDT-C4.5	76	10	89
	ZOLFP	77	6	95
	REP	75	7	94
	MEP	75	11	91
Flare Solar	UDT-C4.5	95	9	54
	ZOLFP	99	8	64
	REP	97	8	72
	MEP	96	9	31

Furthermore, the weighted averages of TP and FP rates and area under ROC curve are also taken into account in order to measure the performance of the pruning methods as in Table 5.10. The proposed method produces the highest TP rate in all datasets and the lowest FP rates for four datasets. Consequently, the proposed

ZOLFP method produces highest scores in terms of the area under ROC compared to REP and MEP with four scores.

5.2.2.4 Comparison of Proposed PBMR and ZOLFP Methods Experiment

On the other hand, the experiment also includes investigation of the performance of the two new proposed methods. A comparison is performed in order to evaluate the performance of these two new methods. The comparison is conducted between the two proposed methods PBMR and ZOLFP in term of accuracy and complexity in Table 5.11.

Table 5.11: Accuracy and tree size for PBMR and ZOLFP methods

Datasets	Algorithms	Accuracy (%)	Number of Nodes	Number of Leaves
Car	PBMR	94	182	129
	ZOLFP	95	88	87
Diabetes	PBMR	75	41	21
	ZOLFP	76	31	16
Labour	PBMR	77	27	23
	ZOLFP	77	36	30
Blogger	PBMR	76	24	17
	ZOLFP	77	33	23
User Knowledge	PBMR	76	400	398
	ZOLFP	77	402	399
Flare Solar	PBMR	98	19	16
	ZOLFP	99	7	6

Scores of accuracy and complexity of two methods are presented in Table 5.12 and Table 5.13. In this context, zero (0) score represents the worse algorithm and one (1) score represents a better algorithm, whereas equal sign (=) represents equality [91]. The comparison of performance accuracies in Table 5.12 shows that ZOLFP is better than PBMR in terms of accuracy with five scores for ZOLFP and one scores for

PBMR. While the comparison of complexity in Table 5.13 shows that PBMR and ZOLFP methods are equal.

Table 5.12: Score of accuracy for PBMR and ZOLFP methods

Algorithms	Scores						Total Wins
	Car	Diabetes	Labour	Blogger	User Knowledge	Flare Solar	
PBMR	0	0	1	0	0	0	1
ZOLFP	1	1	0	1	1	1	5

Table 5.13: Score of complexity for PBMR and ZOLFP methods

Algorithms	Scores						Total Wins
	Car	Diabetes	Labour	Blogger	User Knowledge	Flare Solar	
PBMR	0	0	1	1	1	0	3
ZOLFP	1	1	0	0	0	1	3

Chapter 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

This thesis targeted over-fitting problem in DT that caused several drawbacks such as increasing complexity of the produced tree model, reducing the performance accuracy, and degrading the computational efficiency as well. The research in the thesis adopts post-pruning approaches to combat this obstacle. So, in this context two post-pruning methods are developed which are PBMR and ZOLFP.

In PBMR, the risk-rate is estimated instead of estimating the misclassification errors as in most post-pruning methods like REP. Hence, after the DT is built, PBMR method traverses it in post order to prune. Firstly, it computes the risk-rate of the subtree corresponding to current non-leaf node, and then the risk-rate of its leaf node. If the risk-rate of the former is less than the risk-rate of the later, the leaf node is pruned, otherwise it is retained.

In ZOLFP, the expected loss is estimated for pruning the tree instead of estimating the misclassification error rate. Pruning for loss minimization is another point of view in pruning that may result in different pruning behavior than does pruning for error minimization. This type of pruning relays on the probability distribution to adjust the prediction to minimize the expected loss or to supply a confidence level associated with the prediction. Thus, ZOLFP prunes the tree by computing expected loss for each node, then if the expected loss for this node is less than or equal to the sum of the expected loss for the whole subtree under it, the subtree is pruned and this node is converted to a leaf node, otherwise the subtree is retained.

Six different datasets are employed to train and test the proposed methods in comparison with REP and MEP methods. Experiments are conducted by using java eclipse combined with Weka. Initially worst attributes are removed from the datasets by using OneR and Information Gain attribute evaluators from Weka with 10-fold cross validation. OneR attribute evaluator is utilized since it produces better performance compared to Information Gain evaluator.

Various evaluation methods are exploited to evaluate the efficiency of the proposed methods in terms of attribute selection, accuracy, complexity, precision score, recall score, TP rate, FP rate and area under ROC. The experimental results demonstrated that both of the proposed methods produce better classification accuracy when compared to REP and MEP in all test datasets. In terms of tree complexity, the proposed methods create trees that are not much different than the complexities of REP and MEP, whereas both proposed methods produce smaller tree with compared to UDT-C4.5 in all datasets. On the other hand, the results show that the proposed methods yield satisfactory performance in terms of precision score, recall score, TP rate, FP rate and area under ROC compared to both REP and MEP approaches.

6.2 Future Work

The proposed algorithms adopt post-pruning bottom-up method for C4.5 DT algorithm. As future works, the proposed methods PBMR and ZOLFP can be applied on C5.0 DT classifier and can be also modified for other DT base classifiers such as best first tree and random forest.

REFERENCES

[1] Mendes, R. R., de Voznika, F. B., Freitas, A. A., & Nievola, J. C. (2001, September). Discovering fuzzy classification rules with genetic programming and co-evolution. *In European Conference on Principles of Data Mining and Knowledge Discovery* (pp. 314-325).Springer, Berlin, Heidelberg.

[2] Freitas, A. A. (2013). *Data mining and knowledge discovery with evolutionary algorithms*. Springer Science & Business Media.

[3] Han, J., Pei, J., & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

[4] Osei-Bryson, K. M. (2007). Post-pruning in decision tree induction using multiple performance measures. *Computers & operations research,* 34(11), 3331-3345.

[5] Drazin, S., & Montag, M. (2012). Decision tree analysis using Weka. *Machine Learning-Project II, University of Miami*, 1-3.

[6] Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., &Brodley, C. E. (1998). Pruning decision trees with misclassification costs. I*n European Conference on Machine Learning*, 131-136, Springer Berlin Heidelberg.

[7] Badr, S., & Bargiela, A. (2011). Case study of inaccuracies in the granulation of decision trees. *Soft Computing*, 15(6), 1129-1136.

[8] Yang, H., & Fong, S. (2013). Incremental optimization mechanism for constructing a decision tree in data stream mining. *Mathematical Problems in Engineering*, 2013.

[9] Delen, D., Kuzey, C., & Uyar, A. (2013). Measuring firm performance using financial ratios: A decision tree approach. *Expert Systems with Applications*, 40(10), 3970-3983.

[10] Jiang, L., Li, C., & Cai, Z. (2009). Learning decision tree for ranking. *Knowledge and Information Systems*, 20(1), 123-135.

[11] Yildiz, O. T., & Alpaydin, E. (2005). Linear discriminant trees. *International Journal of Pattern Recognition and Artificial Intelligence*, 19(03), 323-353.

[12] Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVMs. *In Advances in neural information processing systems* (pp. 668-674).

[13] Yazdi, H. S., & Salehi-Moghaddami, N. (2012). Multi branch decision tree: a new splitting criterion. *International Journal of Advanced Science and Technology*, 45, 91-106.

[14] Han J., Pei J., & Kamber M. *Data mining: concepts and techniques*. Elsevier; 2011 Jun 9.

[15] Breiman L., Friedman J., Olshen R., & Stone C. Regression Trees. Wadsworth Int. *Group.*

[16] Buntine W. Learning classification trees. *Statistics and computing*. 1992 Jun 1; 2(2):63-73.

[17] Esposito F., Malerba D., & Semeraro G. Simplifying decision trees by pruning and grafting: New results. *Machine Learning*: ECML-95. 1995:287-90.

[18] D., & Oates, T. (1999). The Effects of Training Set Size on Decision Tree Complexity. In Proc. *14th Int. Conf. on Machine Learning* (pp. 254-262).

[19] Mingers J. An empirical comparison of pruning methods for decision tree induction. *Machine learning*. 1989 Nov 1; 4(2):227-43.

[20] Doni. An analysis of reduced error pruning. *Journal of Artificial Intelligence Research.* 2001; 15: 163-187.

[21] Hall L. O., Bowyer K. W., Banfield R. E., Eschrich S., & Collins R. Is error-based pruning redeemable? *International Journal on Artificial Intelligence* Tools. 2003 Sep; 12(03):249-64.

[22] Mohamed W. N., Salleh M. N., & Omar A. H. A comparative study of reduced error pruning method in decision tree algorithms. *In Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on 2012 Nov 23 (pp. 392-397). IEEE.*

[23] Mehta S., & Shukla D. Optimization of C5. 0 classifier using Bayesian theory. *In Computer, Communication and Control (IC4), 2015 International Conference on 2015 Sep 10 (pp. 1-6). IEEE.*

[24] Esposito, F., Malerba, D., Semeraro, G., & Kay, J. (1997). A comparative analysis of methods for pruning decision trees. *IEEE transactions on pattern analysis and machine intelligence, 19(5), 476-491.*

[25] Kapoor, P., Rani, R., & JMIT, R. (2015). Efficient Decision Tree Algorithm Using J48 and Reduced Error Pruning. *International Journal of Engineering Research and General Science*, 3(3).

[26] Patel, R. R., & Aluvalu, R. (2014). A reduced error pruning technique for improving accuracy of decision tree learning. *International Journal of Engineering and Advanced Technology (IJEAT)ISSN*, 2249-8958.

[27] Alali, A., & Kubat, M. (2015). PruDent: A Pruned and confident stacking approach for Multi-Label classification. *IEEE Transactions on Knowledge and Data Engineering*, 27(9), 2480-2493.

[28] Xie, H., & Shang, F. (2014). The study of methods for post-pruning decision trees based on comprehensive evaluation standard. *In Fuzzy Systems and Knowledge Discovery (FSKD), 14th International Conference on* (pp. 903-908). IEEE.

[29] Xu, Z., Min, F., & Zhu, W. (2012).Cost-sensitive C4.5 with post-pruning and competition. *arXiv preprint arXiv: 1211.4122*.

[30] Zhang, W., & Li, Y. (2013, June). A Post-Pruning Decision Tree Algorithm Based on Bayesian. In *Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on* (pp. 988-991). IEEE.

[31] Osei-Bryson, K. M. (2007). Post-pruning in decision tree induction using multiple performance measures. *Computers & operations research, 34(11), 3331-3345.*

[32] Mohamed, W. N. H. W., Salleh, M. N. M., & Omar, A. H. (2012, November). A comparative study of reduced error pruning method in decision tree algorithms. In *Control System, Computing and Engineering (ICCSCE), 2012 IEEE International Conference on* (pp. 392-397). IEEE.

[33] Mahmood, A. M., Mrithyumjaya, P. G. V. G. K., & Kuppa, R. A new pruning approach for better and compact decision trees. *International Journal on Computer Science and Engineering*, 1(2), 2551-2558.

[34] Kijsirikul, B., & Chong kasemwongse, K. (2001).Decision tree pruning using backpropagation neural networks. *In International Joint Conference on Neural Networks (IJCNN),* 3, 1876-1880.

[35] Galathiya, A. S., Ganatra, A. P., & Bhensdadia, C. K. (2012).Improved decision tree induction algorithm with Feature Selection, Cross Validation, model complexity and reduced error pruning. *International Journal of Computer Science and Information Technologies*, 3(2), 3427-3431.

[36] Patil, D. D., Wadhai, V. M., & Gokhale, J. A. (2010). Evaluation of decision tree pruning algorithms for complexity and classification accuracy. *International Journal of Computer Applications*, 11(2).

[37] You, J. P., Wei, J. M., & Wang, M. Y. (2007). RST in decision tree pruning. *Computer Engineering & Science,* 1, 038.

[38] Wang, H., & Chen, B. (2013, November). Intrusion detection system based on multi-strategy pruning algorithm of the decision tree. *In International Conference on Grey Systems and Intelligent Services*, 445-447.

[39] Chen, J., Wang, X., & Zhai, J. (2009). Pruning decision tree using genetic algorithms. *In International Conference on Artificial Intelligence and Computational Intelligence (AICI),* 3, 244-248.

[40] Esposito, F., Malerba, D., Semeraro, G., & Tamma, V. (1999). The effects of pruning methods on the predictive accuracy of induced decision trees. *Applied Stochastic Models in Business and Industry*, 15 (4), 277-299.

[41] Ling, C. X., & Sheng, V. S. Cost-sensitive learning and the class imbalance problem. 2008.

[42] Zadrozny, B., & Elkan, C. (2001, August). Learning and making decisions when costs and probabilities are both unknown. *In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 204-213). ACM.

[43] Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine learning*, 42(3), 203-231.

[44] Wang, T., Qin, Z., Jin, Z., & Zhang, S. (2010). Handlings over-fitting in test cost-sensitive decision tree learning by feature selection, smoothing and pruning. *Journal of Systems and Software*, 83(7), 1137-1147.

[45] Esmeir, S., & Markovitch, S. (2011). Anytime learning of any cost classifiers. *Machine Learning*, 82(3), 445-473.

[46] Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. E. (1998, April). Pruning decision trees with misclassification costs. In *European Conference on Machine Learning* (pp. 131-136). Springer, Berlin, Heidelberg.

[47] Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, *17*(3), 37.

[48] Bharati, M., & Ramageri, M. (2010). Data mining techniques and applications.

[49] Turing, A. M. (2009). Computing machinery and intelligence. *Mind*, Vol.49, 1950, pp. 433-460. Doi: 10.1093/mind/LIX.236.433.

[50] Lin, W. Y., Hu, Y. H., & Tsai, C. F. (2012). Machine learning in financial crisis prediction: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* 42(4), 421-436.

[51] Ahmed, A. M., Rizaner, A., & Ulusoy, A. H. (2016). Using data mining to predict instructor performance. *Procedia Computer Science,* 102, 137-142.

[52] Wang, Y., & Xia, S. T. (2017, March). Unifying attribute splitting criteria of decision trees by Tsallis entropy. *In Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on* (pp. 2507-2511). IEEE.

[53] Basarabă, L., & Pescaru, D. (2011, May). Using decision tree for efficient data classification in Cyber-physical systems. *In Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on* (pp.385-390). IEEE.

[54] Jiang, L., & Li, C. (2010). An empirical study on attribute selection measures in decision tree learning. *Journal of Computational Information Systems*, 6(1), 105-112.

[55] Fayyad, U. M., & Irani, K. B. (1992, July). The attribute selection problem in decision tree generation. *In AAAI* (pp. 104-110).

[56] Varpa, K., Iltanen, K., & Juhola, M. (2008). Machine learning method for knowledge discovery experimented with otoneurological data. *Computer methods and programs in biomedicine*, 91(2), 154-164.

[57] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, 1(1), 81-106.

[58] Quinlan, J. R. (1993). C4. 5: Programming for machine learning. *Morgan Kauffmann*, *38*, 48.

[59] Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(1), 14-23.*ISSN*, 2249-8958.

[60] Lior, R. (2014). *Data mining with decision trees: theory and applications* (Vol. 81). World Scientific.

[61] Quinlan, R. (2004). Data mining tools See5 and C5. 0.

[62] Rokach, L., & Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications*. Volume 69 of Series in machine perception and artificial intelligence.

[63] Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3), 221-234.

[64] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434.*

[65] Yadav, S., & Shukla, S. (2016, February). Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification. *In Advanced Computing (IACC), 2016 IEEE 6th International Conference on* (pp. 78-83). IEEE.

[66] Olson, D. L., & Delen, D. (2008). *Advanced data mining techniques*. Springer Science & Business Media.

[67] Jensen, D. D., & Cohen, P. R. (2000). Multiple comparisons in induction algorithms. *Machine Learning,* 38(3), 309-338.

[68] Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data mining: Practical machine learning tools and techniques. *Kaufmann, Burlington.*

[69] Buntine, W. (1992). Learning classification trees. *Statistics and computing*, 2(2), 63-73.

[70] Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.

[71] Wu X., Kumar V., Quinlan J. R., Ghosh J., Yang Q., & Motoda H. Top 10 algorithms in data mining. *Knowledge and Information Systems*. 2008 January 1;14 (1):1-37.

[72] Berzal F., Cubero J. C., Cuenca F., Martín-Bautista M. J. On the quest for easy-to-understand splitting rules. *Data & Knowledge Engineering*. 2003 January 31;44 (1):31-48.

[73] Katz G., Shabtai A., Rokach L., & Ofek N. Conf D tree: A statistical method for improving decision trees. *Journal of Computer Science and Technology*. 2014 May 1;29(3):392-407.

[74] Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1987). Occam's razor. *Information Processing Letters*, 24(6), 377-380.

[75] Quinlan J. R., & Rivest R. L. Inferring decision trees using the minimum description length principle. *Information and Computation*. 1989 March 1;80 (3):227-48.

[76] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and Regression Trees. The Wadsworth Statistics and Probability Series, Wadsworth International Group, *Belmont California* (pp. 356).

[77] Quinlan J. R. Simplifying decision trees. *International Journal of Man-Machine Studies*. 1987 September 1; 27(3):221-34.

[78] Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine learning*, *5*(3), 239-266.

[79] Quinlan, J. R. (1987). Decision trees as probabilistic classifiers. *In Proceedings of the Fourth International Workshop 8n Machine Learning* (pp. 31-37).

[80] Mehta, M., Rissanen, J., & Agrawal, R. (1995, August). MDL-Based Decision Tree Pruning. In *KDD* (Vol. 21, No. 2, pp. 216-221).

[81] Niblett, T., & Bratko, I. (1987, January). Learning decision rules in noisy domains. In *Proceedings of Expert Systems' 86, the 6th Annual Technical*

Conference on Research and development in expert systems III (pp. 25-34). Cambridge University Press.

[82] Quinlan, J. R. (1987). Simplifying decision trees. *International journal of man-machine studies*, 27(3), 221-234.

[83] Ahmed, A. M., Rizaner, A., & Ulusoy, A. H. (2018). A novel decision tree classification based on post-pruning with Bayes minimum risk. *PloS one*, *13*(4), e0194168.

[84] Asuncion, A., & Newman, D. (2007). University of California Irvine (UCI) machine learning repository.

[85] Bahnsen, A. C., Stojanovic, A., Aouada, D., & Ottersten, B. (2013, December). Cost sensitive credit card fraud detection using Bayes minimum risk. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*(Vol. 1, pp. 333-338). IEEE.

[86] Bahnsen, A. C., Aouada, D., & Ottersten, B. (2015). Example-dependent cost-sensitive decision trees. *Expert Systems with Applications*, *42*(19), 6609-6619.

[87] Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data mining and knowledge discovery*, 1(3), 291-316.

[88] Kubat, M., Holte, R., & Matwin, S. (1997). Learning when negative examples abound. *Machine learning: ECML-97*, 146-153.

[89] Loh, W. Y. (2011). Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1), 14-23.

[90] Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

[91] Margineantu, D. D., & Dietterich, T. G. (1999). *Learning decision trees for loss minimization in multi-class problems.* Corvallis, OR: Oregon State University, Dept. of Computer Science.