Rule Extraction from Support Vector Machine: Applications to Banking and Finance

A thesis submitted during **2010** to the University of Hyderabad in partial fulfillment of the award of a **Ph.D degree** in **Computer Science**

by

Mohammad Abdul Haque Farquad

Department of/~~Centre for~~ Computer and Information Sciences

School of Mathematics/Computer and Information Sciences

University of Hyderabad

(P.O.) Central University, Gachibowli

Hyderabad – 500 046

Andhra Pradesh

India

CERTIFICATE

This is to certify that the thesis entitled "**Rule Extraction from Support Vector Machine: Applications to Banking and Finance**" submitted by **Mohammad Abdul Haque Farquad** bearing Reg. No **04MCPC03** in partial fulfillment of the requirements for the award of Doctor of Philosophy in **Computer Science** is a bonafide work carried out by him/~~her~~ under ~~my~~/our supervision and guidance.

The thesis has not been submitted previously in part or in full to this or any other University or Institution for the award of any degree or diploma.

(Prof. S. Bapi Raju) (Dr. V. Ravi)

Signature of the Supervisor/s

Head of the Department/Centre Dean of the School

DECLARATION

I **Mohammad Abdul Haque Farquad** hereby declare that this thesis entitled "**Rule Extraction from Support Vector Machine: Applications to Banking and Finance**" submitted by me under the guidance and supervision of **Professor S. Bapi Raju, Department of Computer and Information Sciences, University of Hyderabad** and **Dr. V. Ravi, Institute for Development and Research in Banking Technology, Hyderabad** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this University or any other University or Institution for the award of any degree or diploma.

Date: **24.12.2010**

Name : **Mohammad Abdul Haque Farquad**

Signature of the Student:

Regd. No. **04MCPC03**

Dedicated to:

My proud parents, Rafiuddin and Nazneen

My tremendously inspiring and supportive elder brother, Zahid

My incredibly understanding guides, Dr. Ravi and Prof. Bapi

My wonderful gifted younger brothers and sisters, Salwa, Saad, Sana, Khalid, Majid, Huda, Bushra and Tooba

My extremely, artistically entertaining friends.

Acknowledgements

First of all, I am very grateful to *Almighty* for such a beautiful life and energy to work hard. Later, I would like to thank my parents for their kind support towards achieving my goals, this thesis would not have been possible without my parents' direct or indirect encouragement, I love you mom and dad.

This thesis would not have been possible without the help and support of the kind people around me, to only some of whom it is possible to give particular mention here. Above all, I would like to thank those who made this thesis possible such as my supervisors who helped me with the research material and my brother who is an inspiration for me. I owe my greatest gratitude to my supervisors Dr. V. Ravi, Associate Professor, IDRBT, Hyderabad and Dr. S. Bapi Raju, Professor, Department of Computer and Information Sciences, University of Hyderabad, whose encouragement, supervision and support from the preliminary to the concluding level enabled me to develop an understanding of the subject.

At the outset of my research, I was a bit scared of Dr. Ravi's very strict and demanding nature, but now I realize the importance of the kind of training I was undergoing in his supervision. The best thing, apart from many others, that I could learn from him is the importance of an effective experimentation for solving real life financial problems using my proposed methodologies. Throughout my research, he has been very particular about empirical analysis. My second supervisor Prof. Bapi Raju has always shown confidence in me apart from scanning my research work and progress. Despite being busy, he spared time and made himself available for discussion and encouragement at times, for which I am very grateful. In all, both of my supervisors contributed a lot in building my confidence as a good researcher.

It is an honour for me to thank Prof. Hrishikesh Mohanty and Dr. Durga Bhavani, for serving on my examining committee. I thank you for your precious time in reviewing my research progress and providing thoughtful comments and suggestions. I also thank Mr. B. Sambamurthy, Director, IDRBT, Mr. Krishna Mohan, Former Director IDRBT, Prof. T. Amaranath, Dean, School of Mathematics Computers and Information Sciences, Prof.

Arun Agarwal, Head of the Department, Department of Computer and Information Sciences, University of Hyderabad for extending their cooperation.

I would like to express my sincere thanks to colleagues and friends at IDRBT and University of Hyderabad and elsewhere, they have been generous with their support. Among them are, Zahid (my brother), Azeez, Habeeb, Saleem, Aijaz, Pavan, Srikanth, Kishore, Manoj, Daniel, Bhavani, Clare, Amer, Rushdie, Khalif, Orif, Amit, Nikunj, Satish, Vinay, Anil, Shiva, Karthik, Sunil, Ravi, Shakeel, Naveen, Rohit, Vasu, Jagan, Suresh, Dilip, Jaikumar, Pramod, Amareshwar, Mahesh, Anki, Adi, Vijay, etc. for their entertaining friendship and moral support to accomplish this thesis. If I miss any name I apologise to them for not able to mention but my friends support and due encouragement helped me to complete this thesis.

I would like to acknowledge the financial, academic and technical support provided by IDRBT, Hyderabad, particularly the award of a Research Fellowship that provided me the necessary financial support to carry on this research. I also thank the staff of library, IDRBT, Mr. Ratna Kumar, Mr. Prakash and Mr. Gyan, who helped me through the literature updates. Furthermore, I would like to thank Mr. Murthy and Mr. Prakash, who provided me the technical resources whenever requested. Last but not least, I would like to thank RKHS team as a whole for serving me the most hygienic and energetic food.

I would also like to thank my colleagues and friends in Vivek Vardhini P.G. College, Hyderabad. Last, but by no means least, I thank my friends in my hometown and elsewhere for their support and encouragement throughout, some of whom have already been named. For any errors or inadequacies that may remain in this work, of course, the responsibility is entirely my own.

. . . Thank you all.

M.A.H. Farquad

Contents

List of Tables

List of Figures

ABSTRACT

Although Support Vector Machines have been used to develop highly accurate classification and regression models in various real-world problem domains, the most significant barrier is that SVM generates *black box* model that is difficult to understand. The procedure to convert these opaque models into transparent models is called *rule extraction*. This thesis investigates the task of extracting comprehensible models from trained SVMs, thereby alleviating this limitation. The primary contribution of the thesis is the proposal of various algorithms to overcome the significant limitations of SVM by taking a novel approach to the task of extracting comprehensible models. The basic contribution of the thesis are systematic review of literature on rule extraction from SVM, identifying gaps in the literature and proposing novel approaches for addressing the gaps. The contributions are grouped under three classes, *decompositional*, *pedagogical* and *eclectic/hybrid* approaches. *Decompositional* approach is closely intertwined with the internal workings of the SVM. *Pedagogical* approach uses SVM as an oracle to re-label training examples as well as artificially generated examples. In the *eclectic/hybrid* approach, a combination of these two methods is adopted.

The thesis addresses various problems from the finance domain such as *bankruptcy prediction in banks/firms*, *churn prediction in analytical CRM* and *Insurance fraud detection*. Apart from this various benchmark datasets such as *iris*, *wine* and *WBC* for classification problems and *auto MPG*, *body fat*, *Boston housing*, *forest fires* and *pollution* for regression problems are also tested using the proposed appraoch. In addition, rule extraction from unbalanced datasets as well as from active learning based approaches has been explored. For classification problems, various rule extraction methods such as FRBS, DT, ANFIS, CART and NBTree have been utilized. Additionally for regression problems, rule extraction methods such as ANFIS, DENFIS and CART have also been employed. Results are analyzed using accuracy, sensitivity, specificity, fidelity, AUC and t-test measures. Proposed approaches demonstrate their viability in extracting accurate, effective and comprehensible rule sets in various benchmark and real world problem domains across classification and regression problems. Future directions have been indicated to extend the approaches to newer variations of SVM as well as to other problem domains.

Chapter 1

Introduction to Rule Extraction

Over the last three decades, data mining and machine learning have been remarkably successful in extracting interesting knowledge and hidden patterns from the ever growing databases. The ability to learn from examples is an important aspect of intelligence and this has been an area of study for researchers in artificial intelligence, statistics, cognitive science, and related fields. Algorithms that are able to learn inductively from examples have been applied to numerous difficult, real-world problems of practical interest (Widrow et al., 1994; Langley and Simon, 1995). Inductive learning with comprehensibility is a central activity in the growing field of knowledge discovery in databases and data mining (Fayyad et al., 1996). Predictive accuracy and the comprehensibility are two main driving elements to evaluate any learning system. It is observed that the learning method which constructs the model with the best predictive accuracy is not the method that produces the most comprehensible model. Artificial neural networks (ANN) and support vector machines (SVM), for example, are amongst the most successful machine learning techniques applied in the area of data mining (Byun and Lee, 2002; Burbidge et al., 2001; Cai et al., 2000; 2004; El-Naqa et al., 2002; Guo and Li, 2003; Joachim, 1999; Kalatzis et al., 2003), but produce *black box* models that are difficult to understand by the end user. This thesis explores the following question: can we take an arbitrary, incomprehensible model produced by a learning algorithm, and re-represent it (or closely approximate it) in a language that better facilitates comprehensibility?

1.1 Rule Extraction: Motivation

Support Vector Machines (SVMs) have proved to be good alternative algorithm compared to other machine learning techniques specifically for solving classification and regression problems. However just like artificial Neural Networks (ANN), SVMs also produce black

box models with the inability to explain the knowledge learnt by them in the process of training. Comprehensibility is very crucial in some applications like medical diagnosis, security and bankruptcy prediction etc. The process of converting opaque models into transparent models is often called *Rule Extraction*. Using the rules extracted one can certainly understand in a better way, how a prediction is made. Rule extraction from support vector machines follows the footsteps of the earlier effort to obtain human-comprehensible rules from ANN in order to explain the knowledge learnt by ANN during training. Much attention has been paid during last decades to find effective ways of extracting rules from ANN and very less work has been reported towards representing the knowledge learnt by SVM during training.

1.2 Rule Extraction: Significance

Andrews et al. (1995) describe the motivation behind rule extraction from neural networks. A brief review of the arguments of Andrews et al. (1995) will help to establish aims and significance for rule extraction from SVM techniques.

1.2.1 Provision of user explanation capability

In symbolic artificial intelligence (AI), the term "explanation" refers to an explicit structure which can be used internally for reasoning and learning, externally for the explanation of results to the user. Gallant (1988) observes that an explanation capability enables a novice user to gain insights into the problem at hand. Davis et al. (1977) argues that even limited explanation can positively influence acceptance of the system by the user. Traditionally, researchers have experimented with various forms of user explanation, in particular *rule traces*. It is obvious that explanations based on rule traces are too rigid and inflexible (Gilbert, 1989) because rules may not be equally useful to the user. Further, the granularity of the rule traces' explanation is often inappropriate (Gilbert, 1989; Andrews et al., 1995).

1.2.2 Transparency

The creation of a "*user explanation*" capability is the primary objective for extracting rules from neural networks and SVMs, with the provision of "*transparency*" of the internal

states of a system. Transparency means that internal states of the machine learning system are both accessible and can be interpreted unambiguously. Such capability is mandatory if neural network or SVM based solutions are to be accepted into "*safety-critical*" problem domains such as air traffic control, operations of power plants, medical diagnosis, etc (Andrews et al., 1995).

1.2.3 Data exploration

A learning system might discover salient features in the input data whose importance was not previously recognized (Craven and Shavlik, 1994).

1.3 Rule Extraction: A Taxonomy

More broadly taxonomy for rule extraction from ANN has been introduced (Andrews et al., 1995; Tickle et al., 1998) which includes five evaluation criteria: *translucency, rule quality, expressive power, portability* and *algorithmic complexity*. These evaluation criteria are now commonly used for rule extraction from SVMs, in particular, the translucency and rule quality metrics (Martens et al., 2009; Martens et al., 2006). Rule extraction from neural networks has previously almost exclusively been used to generate propositional rule sets (Hayward et al., 2000). While this is sufficient for many applications where rule sets can be effectively used, it is clearly desirable to provide a more general explanation capability.

A significant research effort has been expended in the last few decades to address the deficiency in the understandability of ANN (Saito and Nakano, 1988; Thrun, 1995; Craven, 1996; Jackson and Craven, 1996). Craven (1996) presented a complete overview on this research. The generally used strategy to understand a model represented by a trained neural network is to translate the model into a more comprehensible language (such as a set of if-then rules or a decision tree). This strategy is investigated under the rubric of rule extraction.

Craven (1996) defines the task of rule extraction from neural network as follows:

"*Given a trained neural network and the data on which it was trained, produce a description of the network's hypothesis that is comprehensible yet closely approximates the network's prediction behaviour*."

Although the task of rule extraction has only been formally formulated in the context of interpreting neural networks, this formulation can be generalized to any other opaque model such as SVM.

Over the last few years, a number of studies on rule extraction from SVM have been introduced. The research strategy in these projects is often based on this idea: develop an algorithm for rule extraction based on the perception (or "view") of the underlying SVM which is either explicitly or implicitly assumed within the rule extraction technique. In the context of rule extraction from neural networks the notion of "*translucency*" describes the degree to which the internal representation of the ANN is accessible to the rule extraction technique (Andrews et al. 1995; Tickle et al. 1998). Therefore, rule extraction algorithms are classified into three types: Decompositional, Pedagogical and Eclectic. Figure 1.1 shows the categorization of rule extraction algorithms in general and the methodology based on which this categorization is proposed.

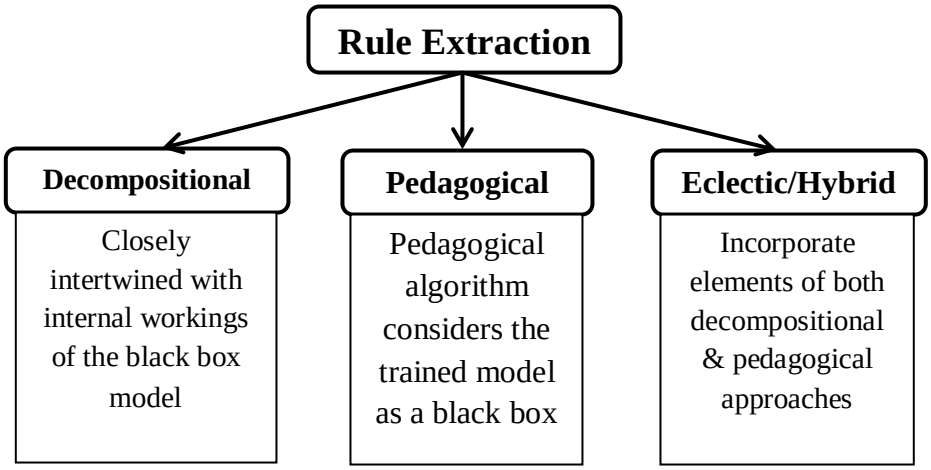


Figure 1.1: Various Categories of Rule Extraction Approaches

1.3.1 Decompositional Approaches

A decompositional approach is closely intertwined with internal workings of the SVM, namely with support vectors and the constructed hyperplance (Nunez et al., 2002a; 2002b;

2002c; Fung et al., 2005, Barakat and Bradely, 2007, Martens et al., 2009). Towell and Shavlik (1993) focused on extracting symbolic rules from the trained feedforward neural network. The rules thus extracted are more general and yielded superior performance compared to earlier approaches. They concluded that their approach is capable of producing more human comprehensible rules. Arbatli and Akin (1997) extracted rules form neural network using Genetic Algorithm.

Fu (1994) analyzed the ability of certainty-factor-based activation function which can improve the network generalization performance from a limited amount of training data. He applied the proposed rule extraction approach to molecular genetics and concluded that it outperformed the standard C4.5 decision tree algorithm. Later, Craven and Shavlik (1996) proposed a learning-based rule extraction approach to provide the explanation capability to trained neural network. The proposed algorithm is able to extract both conjunctive and M-of-N rules, and they concluded that it is more efficient than conventional search-based approaches.

1.3.2 Pedagogical Approaches

Pedagogical algorithm considers the trained model as a black box (Clark and Niblett, 1989; Craven and Shavlik, 1996; Martens et al., 2006). Instead of looking at the internal structure, these algorithms directly extract rules which relate the inputs and outputs of the SVM. These techniques typically use trained SVM model as an oracle to label or classify artificially generated training examples that are later used by a symbolic learning algorithm. The idea behind these techniques is the assumption that the trained model can better represent the data than the original dataset.

Many approaches to rule extraction have set up the task as a search problem. This search problem involves exploring a space of candidate rules and testing individual candidate against the network to see if they are valid rules. One of the first rule-extraction methods developed by Saito and Nakano (1988) employs a breadth-first search process to extract conjunctive rules in binary problem domains. Gallant (1988) developed a similar rule-extraction technique, which like the method of Saito and Nakano, manages the combinatorics of searching for rules by limiting the search depth. The principal difference

between the two approaches is the procedure used to test rules against the network. Unlike Saito and Nakano's method, Gallant's rule-testing procedure is guaranteed to accept only rules that are valid.

Thrun (1995) developed a method called validity interval analysis (VIA) that is a generalized and more powerful version of Gallant's technique. VIA tests rules by propagating activation intervals through a network after constraining some of the input and output units. Thrun frames the problem of determining validity intervals as a linear programming problem. Search methods for rule extraction from neural networks work by finding those combinations of inputs that make the neuron active. By sorting the input weights to a neuron and ordering the weights suitably, it is possible to prune the search space. Using this said concept Krishnan et al. (1999) proposed a rule extraction approach which extracts crisp rules from the neural network. McGarry et al. (1999) proposed a rule extraction approach to extract rules from Radial Basis Function Networks. Based on the neurons of the network, later, Fan and Li (2002) extracted diagnostic rules from trained feed forward neural network. They applied the rule extraction procedure for detecting a high-pressure air compressor's (HPAC) suction and discharge valve faults from static measurements including temperatures and pressures of various stages of the compressor.

1.3.3 Eclectic/Hybrid Approaches

Some authors also consider a third category: eclectic rule extraction techniques, which incorporate elements of both the decompositional and pedagogical approaches (Nunez et al., 2006, Barakat and Diederich, 2005). Sato and Tsukimoto (2001) proposed a hybrid approach of rule extraction, where they employed decision tree algorithm with trained neural networks to generate rules. Campos and Ludermir (2005) presented Literal and ProRulext algorithms to extract rules from the trained artificial neural network. Aliev et al. (2008) used fuzzy recurrent neural network for extraction of rules for battery charging.

1.4 Rule Quality Criteria

The quality of the extracted rules is a key measure of the success of the rule extraction algorithm. Four rule quality criteria were suggested for rule extraction algorithm (Andrews

et al., 1995; Tickle et al., 1998): *rule accuracy*, *fidelity*, *comprehensibility* and *portability*. In this context, a rule set is considered to be *accurate* if it can correctly classify previously unseen examples.

$$Accuracy = \frac{\#\text{ of Patterns Correctly Classified by Rules}}{\text{Total number of patterns in Test data}} \times 100$$

When we deal with two-class classification problem, specifically in finance domain, we need to consider Sensitivity, Specificity and AUC as well.

$$Sensitivity = \frac{\#\text{ of Positive samples Correctly Classified as Positive by Rules}}{\text{Total number of Positive samples in Test data}} \times 100$$

$$Specificity = \frac{\#\text{ of Negative samples Correctly Classified as Negative by Rules}}{\text{Total number of Negative Samples in Test data}} \times 100$$

Positive samples are referred to the class of objective of the study. For example, if we are solving the problem of Churn prediction, then predicting churn is object of the study, hence, instances related to churn are positive samples. Likewise, negative samples for the above example will be the samples for non-churner class or loyal customers-class.

A Receiver Operating Characteristics (ROC) graph (Fawcett, 2006) has long been used in signal detection theory to depict the trade-off between hit accuracies and false alarm accuracies of classifiers. The ROC graph is a two dimensional graph which represents various classifiers based on their output results in the point form in a region, which has FP rate (100-Specficity) on the X-axis and TP rate (Sensitivity) on the Y-axis. ROC provides richer measure of classification performance than scalar measures such as accuracy, error rate or error cost. Higher the area under ROC, better the classifier (refer to Figure 1.2 below). The area under ROC curve (AUC) of a Classifier A can be calculated as the sum of the areas of Triangle - OAD, Rectangle - DAEH and Triangle - AEG. Likewise the AUC of classifier B will be the sum of the areas of Triangle – OBC, Rectangle – CBFH and Tringle – BGF. Thus, classifier B is superior to classifier A on the AUC criterion.

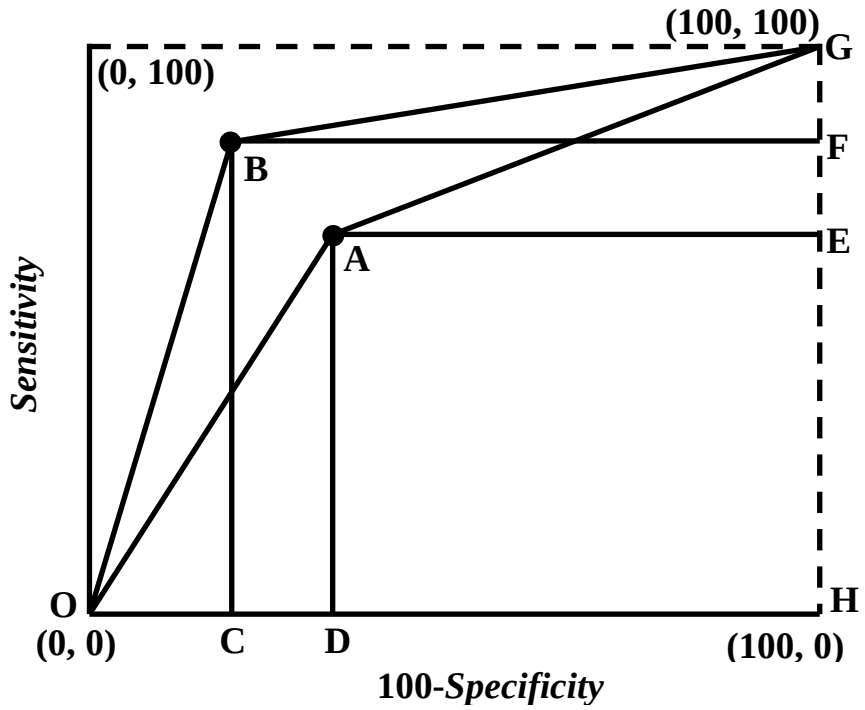


Figure 1.2: AUC – Area Under ROC Curve

Similarly a rule set is considered to display a high level of *fidelity* if it can mimic the behaviour of the machine learning technique from which it was extracted.

$$Fidelity = \frac{\#\text{ of Patterns where Classification of Rules AGREE with the Classification of SVM}}{\text{Total number of patterns in Test data}}$$

An extracted rule set is deemed to be *consistent* if, under different training sessions the machine learning technique generates same rule sets that produce the same classifications of unseen examples. Finally the *comprehensibility* of a rule set is determined by measuring the size of the rule set (in terms of number of rules) and the number of antecedents per rule. Expressive power refers to type or the language of the extracted rules. Propositional (If-Then) rules are the most commonly extracted rules. However, other types are also extracted such as fuzzy rule set (Faifer et al., 1999), and finite state machines (Giles and Omlin, 1993). *Portability* refers to the independence of a rule-extraction method from the ANN architecture and/or a training method.

1.5 Thesis Overview

In this thesis, novel algorithms are presented and evaluated for the task of extracting comprehensible descriptions from hard-to-understand learning systems i.e. support vector machine. The hypothesis advanced by this research is that it is possible to develop algorithms for extracting symbolic descriptions from trained SVMs that: (i) produce more comprehensible, high-fidelity descriptions of trained SVMs using fuzzy logic approaches, (ii) scale to analyze medium scale unbalanced data, (iii) various modifications of the training data such as, *Case-SA* dataset which represents the support vectors with corresponding actual target values, *Case-SP* dataset which represents the support vectors with corresponding predicted target values and *Case-P* dataset which represents the training set with corresponding predicted target values and (iv) applications are extended to Bankruptcy Prediction, Churn Prediction in Bank Credit Cards, Insurance Fraud Detection and Regression Analysis. During this research work, various rule learning algorithms have been analyzed such as, Fuzzy Rule Based System (FRBS), Decision Tree (DT), Classification and Regression Tree (CART), Adaptive Network based Fuzzy Inference System (ANFIS), Dynamic Evolving Fuzzy Inference System (DENFIS) and Naive-Bayes Tree (NBTree). Description of the rule extraction techniques on SVM and SVR are given in Appendix A. Table 1.1 below presents various approaches proposed by us for extracting *if-then* rules from SVM.

Table 1.1: Various approaches proposed for Rule Extraction from SVM

#	Dataset Modification	Rule Generating Algorithms	Output	Problem
1	Support Vector + Actual Class Labels **(*Case-SA* dataset)**	FRBS, DT	Fuzzy *if-then* Rules	Classification
2	Feature Selection + Training set + SVM Predicted Class Labels **(*Case-P* dataset)**	CART, ANFIS, DENFIS	Decision Tree, Fuzzy Inference System	Classification and Regression
3	Support Vector + SVM Predicted Class Labels **(*Case-SP* dataset)**	CART, ANFIS, DENFIS	CART Tree, Fuzzy Inference Systems	Regression
4	Feature Selection + Balancing Techniques + Support Vectors + SVM Predicted Class Labels **(*Case-SP* dataset)**	Naïve Bayes Tree	NBTree	Classification (Data Mining)
5	Feature Selection + ALBA with modified boundary + Support Vectors + SVM Predicted Class Labels **(*Case-SP* dataset)**	Naïve Bayes Tree	NBTree	Classification (Data Mining)

1.6 Datasets used

Various datasets are analyzed during this research work which includes benchmark datasets, medical diagnosis dataset, finance datasets and regression datasets.

Benchmark datasets (UCI Machine Learning Repository):

Iris and

Wine.

Medical Diagnosis dataset (UCI Machine Learning Repository):

Wisconsin Breast Cancer (*WBC*).

***Finance datasets*:**

Bankruptcy prediction in banks datasets include: *Spanish banks, Turkish banks, US banks and UK banks*. The *Spanish banks' dataset* is obtained from (Olmeda and Fernandez 1997). Spanish banking industry suffered the worst crisis during 1977-85 resulting in a total cost of 12 billion dollars. The ratios used for the failed banks were taken from the last financial statements before the bankruptcy was declared and the data of non-failed banks was taken from 1982 statements, financial ratios considered are presented in Table C.1 in Appendix C. This dataset contains 66 banks where 37 went bankrupt and 29 were healthy banks.

Turkish banks' dataset is obtained from (Canbas and Kilic 2005) and is available at (http://www.tbb.org.tr/english/bulten/yillik/2000/ratios.xls). Banks association of Turkey published 49 financial ratios. Initially, Canbas and Kilic (2005) applied univariate analysis of variance (ANOVA) test on these 49 ratios of previous year for predicting the health of the bank in present year. Canbas and Kilic (2005) found that only 12 ratios (refer to Table C.2 in Appendix C) act as early warning indicators that have the discriminating ability (i.e. significance level is <5% in ANOVA) for healthy and failed banks, one year in advance. Among these variables, 12^{th} variable has some missing values meaning that the data for some of the banks are not given. So, we filled those missing values with the mean value of the variable following the general approach in data mining (Fayyad et al. 1996). The resulting dataset contains 40 banks where 22 banks went bankrupt and 18 banks were healthy.

The *US banks' dataset* contains 129 banks from the Moody's Industrial Manual where banks went bankrupt during 1975-1982 (Rahimian et al. 1996). This dataset includes 65 bankrupt banks and 64 healthy banks. Financial ratios pertaining to US banks data are presented in Table C.3 of Appendix C.

The *UK banks' dataset* is obtained from Beynon and Peel (2001). This dataset consists of 10 features, 30 bankrupt banks and 30 healthy banks data. Financial ratios pertaining to UK banks data are presented in Table C.4 of Appendix C.

The *churn prediction in bank credit card customers' data* is obtained from a Latin American Bank that suffered from an increasing number of churns with respect to their credit card customers and decided to improve its retention system. Two groups of variables are available for each customer: sociodemographic and behavioural data, which are described in Table C.10 in Appendix C. The dataset comprises 22 variables, with 21 predictor variables and 1 class variable. It consists of 14814 records, of which 13812 are nonchurners and 1002 are churners, which means there are 93.24% nonchurners and 6.76% churners. Hence, the dataset is highly unbalanced in terms of the proportion of churners versus nonchurners (Business Intelligence Cup 2004).

Automobile insurance fraud detection data is provided by Agnoss Knowledge Seeker Software and this is the only available fraud detection dataset. Originally named "carclaims.txt", it can be found in the accompanying compact disc from Pyle, (1999). This dataset contains 11338 records from January 1994 to December 1995, and 4083 records from January 1996 to December 1996. It has a 6% fraudulent and 94% legitimate distribution, with an average of 430 claims per month. The original dataset has 6 numerical features and 25 categorical features, including the binary class label (fraud or legal). Description of original fraud detection dataset's feature is presented in Table 7.1.

Regression dataset (UCI Machine Learning Repository and StatLib repository):

AutoMPG,
Body fat,
Boston Housing,
Forest Fires and
Pollution.

1.7 Experimental Setup

Results and discussions in this thesis is carried out in a little different fashion. We first divided the dataset into two parts in 80:20 ratios. 20% data is then named validation set and stored aside for later use and 80% of the data is used as training set. Then 10-fold cross validation was performed on the 80% of the data i.e. training data for building the model

and extracting rules. Later, the efficiency of the rules is evaluated against validation set i.e. 20% of the original data. Empirical results presented in this thesis are average results on validation set, intermediate average results obtained during 10-fold cross validation are not presented in this thesis. Figure 1.3 presents the experimental setup followed throughout the research work presented in this thesis.

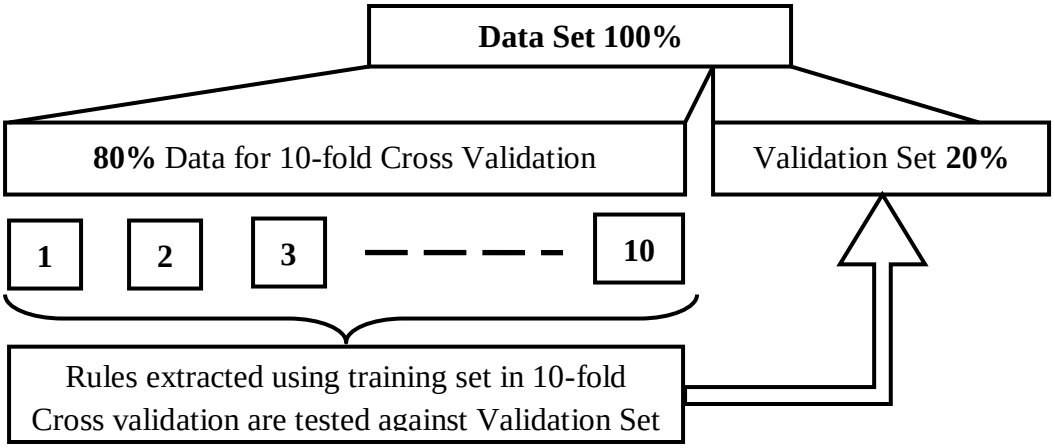


Figure 1.3: Experimental Setup

1.8 Thesis Outline

The remaining chapters of this thesis are organized as follows;

Chapter 2 provides the literature survey of the approaches proposed for extracting rules from SVM. The first section describes in detail about the rule extraction approaches proposed in decompositional, pedagogical and eclectic/hybrid category. Next section provides the details about the gaps observed during literature survey and the final section in this chapter provides an outline of the proposed methods.

Chapter 3 presents the proposed decompositional rule extraction approach for SVM which is one of the main contributions of this thesis. First section of the chapter provides the motivation behind extracting fuzzy rules for solving bankruptcy prediction problems. In the second section details about the proposed approach are presented. Third section provides the literature survey of bankruptcy prediction. Datasets used and results and

discussions are presented in the following sections and final section presents the conclusions of the chapter.

Chapter 4 presents a *pedagogical rule extraction approach* proposed which uses SVM as feature selection algorithm and using reduced feature set rules are extracted. With the motivation in the first section, the proposed approach is presented in detail in next section. Applications of the proposed approach for solving classification and regression problems are discussed in the next section. Following section provides the detailed Results and discussions. Final section concludes the chapter.

Chapter 5 presents an *eclectic rule extraction technique* which analyzes medium scale unbalanced dataset pertaining to finance. At the outset the motivation for the proposed approach is presented. Introduction to customer relationship management (CRM) is then presented and literature of churn prediction problem is reviewed in the next section. Subsequent section presents the proposed approach in detail. The proposed approach can also be considered as an application of analytical CRM. Fourth section provides the details about the literature to deal with unbalanced datasets. Dataset description and Results and discussions is presented in the following sections. Final section concludes the chapter.

Chapter 6 presents an *eclectic active learning based rule extraction approach* which involves active learning to modify the training data. In the beginning section of the chapter motivation behind the proposed approach is presented. Later, section two provides the details about the proposed approach and active learning. Next sections in this chapter present the details of problems analyzed followed by Results and discussions. Final section concludes the chapter.

Chapter 7 presents an *eclectic rule extraction approach* proposed for solving regression problems. First section presents the motivation behind the proposed approach. Proposed approach is then presented in detail in second section. Brief description about the datasets used for empirical study is presented in the next section. Fifth section discusses the results and the implications. Final section concludes the chapter. Finally, Chapter 8 presents the overall conclusion and contributions of this thesis, and proposed future work.

Chapter 2

Rule Extraction from SVM: An Introduction

This Chapter provides the literature survey of the approaches proposed for extracting rules from SVM. The first section describes in detail about the rule extraction approaches proposed in decompositional, pedagogical and eclectic/hybrid category. Next section provides the details about the gaps observed during literature survey and the final section in this chapter provides an outline of the proposed methods.

2.1 Rule Extraction from SVM

In this section, an overview of rule extraction from SVM is presented. The rule extraction approaches are grouped into three categories based on the SVM components utilized for rule extraction. In particular, the first category of algorithms uses support vectors only i.e. decompositional approaches. The second treats SVM as a black box and uses the developed SVM as an oracle, this is also called as pedagogical approaches, and the third category approaches utilize the support vectors, the decision function and the training data as well i.e. eclectic or hybrid approaches.

2.1.1 Decompositional Approaches

Methods in this group extract rules utilizing the SVM's support vectors and the separating hyperplane. Different methods have been used to extract rules from support vectors as summarized in the following paragraphs.

Most recently, Martens et al., (2009) proposed an active learning based approach (ALBA) for rule extraction from SVM. They first extracted support vectors and the distance between support vectors and the actual training set is calculated. Later, additional training samples are generated which are "close to" randomly selected support vectors based on the distance consideration. Class labels for these samples and training samples are then

obtained using the developed SVM model. The generated examples are then used with the training data to train different rule learning algorithm that learn what SVMs have learned. Steps involved in ALBA are presented in Figure 2.1. As decision tree works better with more number of samples, the authors argued that generating more number of samples near support vectors and using these samples in combination with the training data whose target class is replaced by the prediction of SVM will increase the performance of the decision tree algorithm. They employed decision tree and RIPPER to generate rules.

The main drawback of their approach is that ALBA is not feasible to deal with real life data mining problems, where the class distribution is unbalanced and dataset size is medium scale or large scale. The approach resulting in high complexity of the system and the rule set because they used the generated instances with all the training instances thereby increasing the number of instances for rule induction algorithm to learn from.

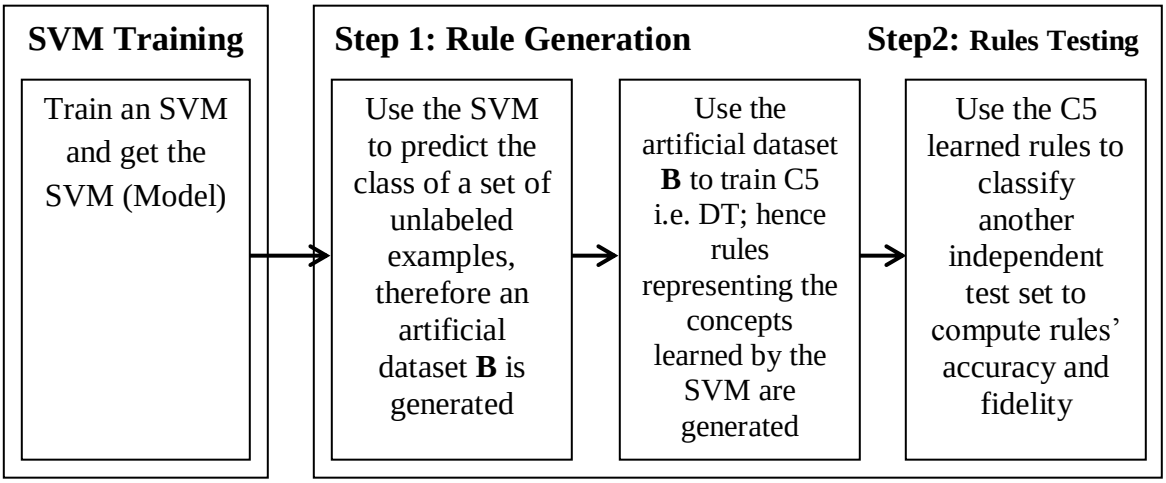


Figure 2.1: Active Learning Based Approach by Martens et al., (2009)

One of the most recent methods in this group termed SQRex-SVM (Barakat and Bradley, 2007) extracts rules directly from support vectors of an SVM using a modified sequential covering algorithm and based on ordered search of the most discriminative features as measured by inter-class separation. An initial set of rules is formed from these features which are then pruned using user defined criteria and utilizes the concept of coverage or probabilistic normal (PN) space (Furnkranz and Flach, 2005) and its relationship to the area under the receiver operating characteristic curve (AUC).

Steps involved in SQRex-SVM algorithm:

SVM training

Train an SVM and get the SVM model.

Rule Generation:

- Use the SVM to predict the class of the training examples that become SVs.
- Get the True Positive and True Negative model SVs, then define a ranked list of most discriminative features based on inter-class separation.
- Obtain best threshold for each of these features and rank them based on True Positive and False Positive rates over the support vectors.
- Formulate rules for the positive class from the ranked features and the corresponding thresholds in earlier step.
- Sequentially, learn/refine rules if possible and add them to rule set if their performance is above user defined criterion.

Rule Pruning

Prune the rules that do not lead to a significant increase in AUC of the whole rule set.

Performance of the rules is evaluated in terms of accuracy, fidelity, comprehensibility, True Positive rate, False Positive rate and AUC. Rules produced by SQRex-SVM exhibit both good generalisation performance and comprehensibility, in addition the extracted rules have high fidelity to the SVM from which they were extracted.

Chaves et al., (2005) proposed a method of extracting fuzzy rules from SVMs. The main idea of their approach is to project each feature, in each of the support vectors along its coordinate axes, forming a number of fuzzy sets with triangular membership functions of the same length (Kecman, 2001). Fuzzy membership degrees are then computed and each of the support vectors is then assigned to the fuzzy set with the highest membership degree. One fuzzy rule is then extracted from each support vector.

The example fuzzy rule is given below;

If x_1 is C_{11} and x_2 is C_{21} then class Z.
where, C_{11} is fuzzy set C_1 of feature x_1, C_{21} is fuzzy set C_2 of feature x_2.

Steps involved in (Chaves et al., 2005) rule extraction approach are described below.

SVM training

Train an SVM and get the support vectors.

Rule Generation

- Project each feature in the support vectors along its coordinate axes.
- Define fuzzy sets for each feature using triangular membership function.
- Evaluate the membership degree for each of the support vector projections and assign it to the set with maximum membership value.
- Generate a rule from each support vector with each feature and fuzzy set of maximum membership value as antecedent and the support vector class label as the consequent.

One limitation of this algorithm is that it may not be suitable for categorical and binary features as the evaluation of membership degree value is non-trivial in these cases. Rules extracted by this method appear to be of poor quality based on the accuracy (i.e. 53.2%) and fidelity (74.59%) reported. Furthermore, the extracted rule set comprehensibility is low as the number of the extracted rules is large, which is as many as the number of support vectors. However, the authors argue that if the objective is explanation, then poor classification performance is acceptable.

Fu et al., (2004) suggested a method RulExSVM for rule extraction from non-linear SVMs, trained with radial basis function (RBF) kernel. The idea behind RulExSVM is to find hyper-rectangles whose upper and lower corners are defined by finding the intersection of each of the support vectors with the separating hyperplane.

RulExSVM proceeds in three phases: initial rule generation, tuning and rule set pruning. In the initial phase, hyper-rectangles are defined by the intersections of lines extended from

each of the support vectors with the SVM decision boundary obtained. In the tuning phase, these hyper-rectangles are resized to exclude outliers from other classes. Finally, redundant rules are removed in the pruning phase. One limitation of this approach is that it is only valid for rule extraction from RBF kernel SVMs. However, the authors argue that this method could be extended to other types of kernels though they provide no framework for this. Another potential limitation is the requirement for normalization of all features to lie between (0, 1) before training. Normalization should be handled with care in applications such as medical diagnosis because some normalization methods may be susceptible to noise in the data. Furthermore, it adds an extra burden on the rule extraction process, as the features have to be transformed back to the original values for the extracted rules to make sense.

Figure 2.2 A shows the lines extended from SVs A1 and A2, and the hyper-rectangles defined by their intersections with the SVM decision boundary. Figure 2.2 B shows the tuning phase, where the two hyper-rectangles were chopped along coordinate axes to exclude outliers from other classes, while Figure 2.2 C shows the pruning phase which removes the overlapping hyper- rectangle (redundant rules).

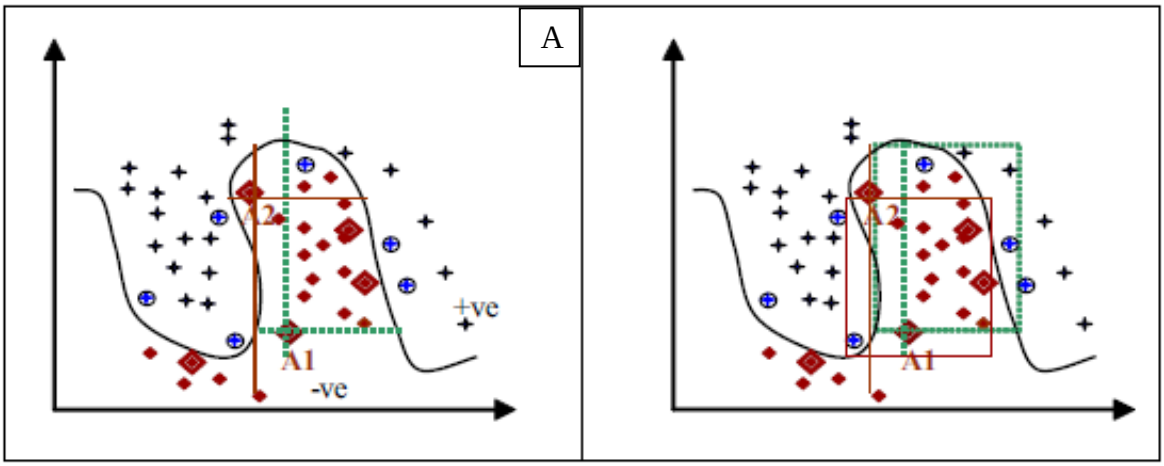


Figure 2.2 A: Rule generation phase: lines extended and hyper-rectangles constructed for A1 and A2 SVs

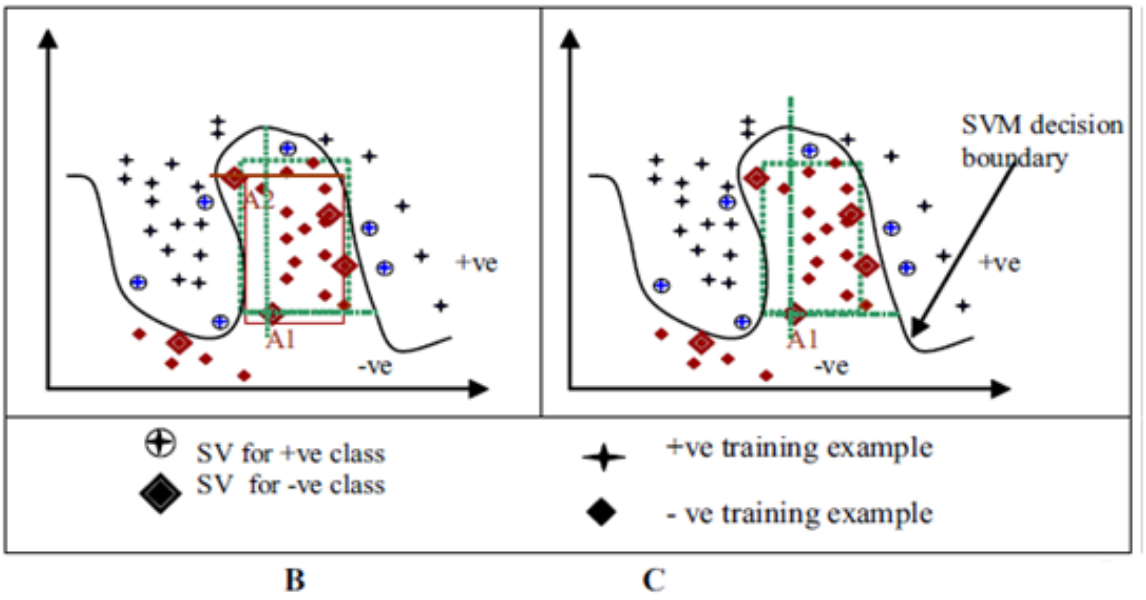


Figure 2.2 B: Tuning Phase: Excluding Outliers

Figure 2.2 C: Pruning Phase: Removing Overlapped Hyper-Rectangles

Figure 2.2: RulExSVM phases in a two dimensional space. Taken from Fu et al. (2004).

The first method by Nunez et al., (2002a, 2002b, 2002c; 2006) introduced the SVM+prototype method. The main idea of this method is to utilize a clustering algorithm to determine prototype vectors for each class which are then used together with the support vectors to define regions (ellipsoids and hyper-rectangles) in the input space. Later, each ellipsoid is transformed to a rule. Figure 2.3 A shows the first iteration of the algorithm where an initial ellipsoid (hyper-rectangle) is defined with outliers (positive partition test). Figure 2.3 B shows a later iteration after the division of the ellipsoid (hyper-rectangle) to exclude outliers.

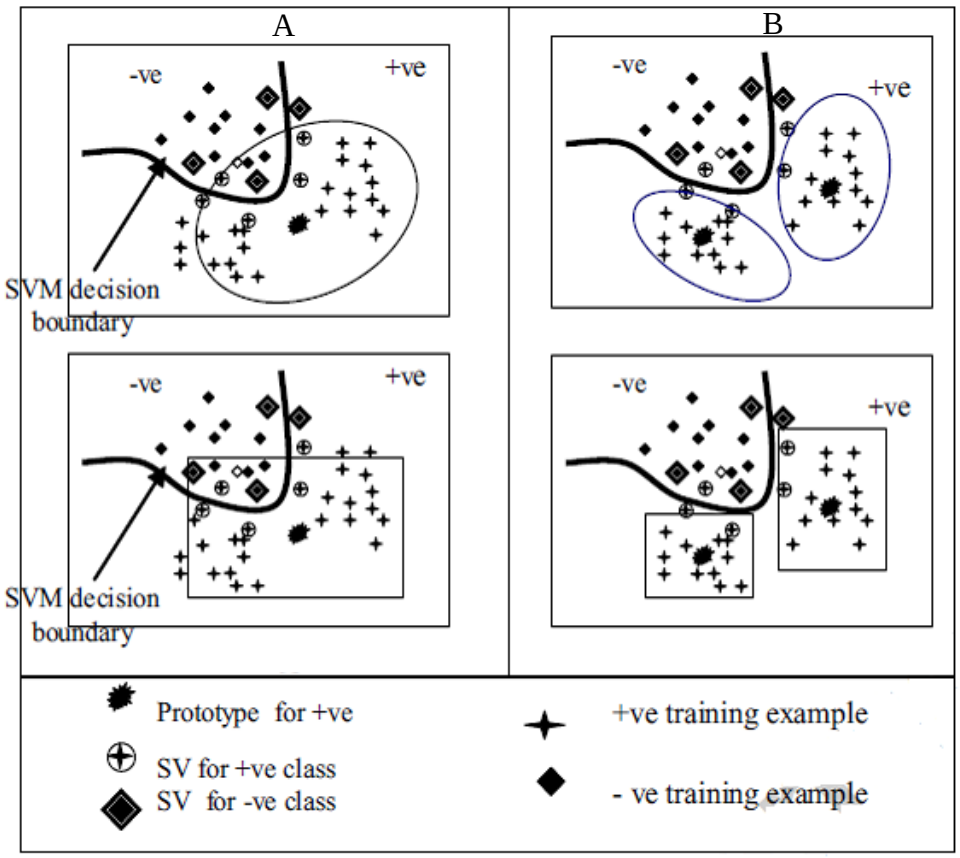


Figure 2.3 A: One Region (Cluster) including Outliers
Figure 2.3 B: Region Division to Exclude Outliers
Figure 2.3: SVM+Prototype Phases. Taken from Nunez et al., (2002)

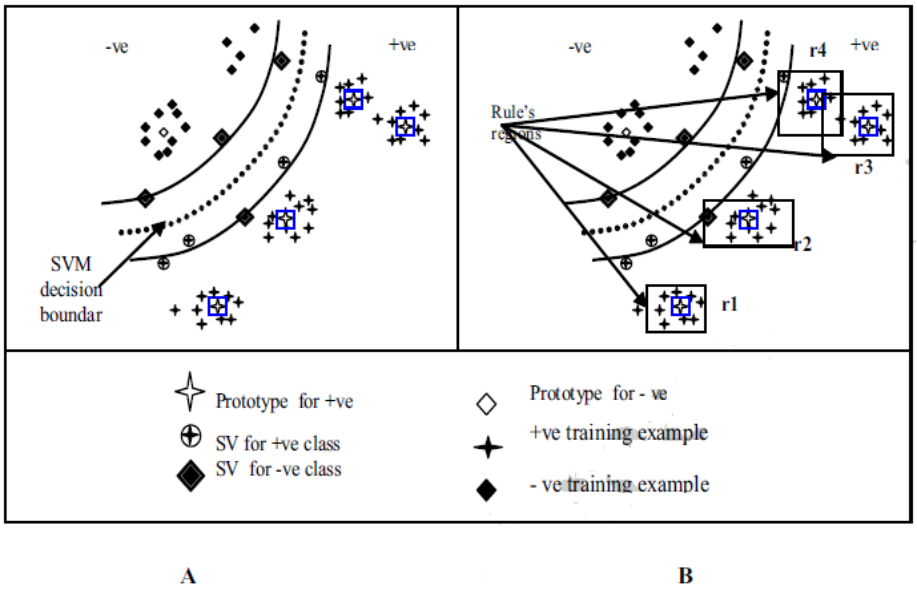
However, the algorithm has two main limitations: first is that the number and the quality of the extracted rules depend upon the initial values of the prototype vectors which define the centers of the clusers. The second limitation is that the method does not scale well. In the case of larger number of input features and/or training examples, the comprehensibility of the extracted rule set will deteriorate as more clusters are likely to be formed, while all the features are present as rules antecedents (Martens et al., 2009).

In a similar study Zhang et al., (2005) proposed the hyper-rectangle rule extraction (HRE) which also utilizes a clustering algorithm. The algorithm first employs support vector clustering algorithm (Ben-Hui et al., 2001) to find prototype vectors for each class. Small hyper-rectangles are then defined around these prototypes using a nested generalized

exemplar algorithm, which are incrementally grown until the stopping criterion is met. Stopping criterion is to control the size of the hyper-rectangles, hence quality rules are generated.

Figure 2.4 A shows the initially defined, small hyper-rectangles around the prototype vectors. Figure 2.4 B shows larger hyper-rectangles. Given a user defined Minimum Confidence Threshold (MCT), the algorithm defines the following criteria to stop growing the hyper-rectangles:

1. All examples of a cluster are covered by a hyper-rectangle and the confidence is less than the (MCT), as in the r1 hyper-rectangle in Figure 2.4 B;
2. The hyper-rectangle covers a support vector of the cluster, as in the r4 hyper-rectangle in Figure 2.4 B;
3. The hyper-rectangle covers a training vector from a different class and the confidence is less than the MCT, as in the r2 hyper-rectangle in Figure 2.4 B;
4. The hyper-rectangle covers a prototype of a different cluster as in the r3 hyper-rectangle in Figure 2.4 B.



2.4 A. Iteration "1"
2.4 B. Iteration "*n*" showing different stopping Criteria
Figure 2.4: Hyper-rectangle region generated for each cluster per class, Zhang et al., 2005

In another study, Fung et al., (2005) suggested a different approach for rule extraction from linear SVMs based on a linear programming formulation of the SVMs with a linear kernel. The approach handles rule extraction as multiple constrained optimization problems to generate a set of non-overlapping rules. Each extracted rule defines a non-empty hyper-cube which has one vertex that lies on the separating hyper-plane. The rule extraction algorithm iterates over training data in each half space, to find rules for the examples which have not been covered by any previous rule using a depth-first search.

One limitation of this approach is that the extracted rules cover training data, so they may not provide an explanation for new unseen data. However, authors have suggested modifying the volume maximisation rule extraction method to generate only one rule with maximum volume that may contain (generalize over) the test examples.

In a related study Chen and Wei (2007) proposed a novel and interesting rule extraction algorithm for gene expression data to improve the comprehensibility of SVMs. The algorithm constitutes one component of a multiple kernel SVM (MK-SVM) scheme, consisting of feature selection namely MK-SVMI, prediction modelling and rule extraction namely MK-SVMII. In the feature selection module, a new single feature kernel (MKI) is proposed which transforms the computationally expensive feature selection problem into finding sparse feature coefficients representing the weight of a single feature kernel. Features with zero coefficients have no impact on the output of SVM and can be discarded.

The rule extraction method proposed for the MK-SVMII is similar to the one addressed in Fung et al. (2005). However, in MK-SVMII, support vectors are used as vertices of hyper-cubes, where a series of hyper-cubes approximates the subspace of each class. They suggested the following rule quality measure in addition to the comprehensibility of the rules. *Soundness*, which measures the number of times a rule is correctly fired. *Completeness*, which measure the number of times the sample is correctly classified by a specific rule and false-alarm, which measures the number of times each rule is misfired (Chen and Wei 2007). The extracted rules by this method have good comprehensibility, good generalization performance and compact gene subsets were found. Furthermore, the

authors argue that multiple good diagnostic gene subsets found to be useful in defining possible pathways of genetic network.

2.1.2 Pedagogical Approaches

Methods in this group treat the SVM as a black box, and extract rules that describe the relationship between the model's inputs and outputs (Barakat and Diederich, 2004a, 2004b; Torres and Rocco, 2005; Martens et al., 2006)

The idea is to create artificially labelled samples where the target class of the training data is replaced by the class predicted by the SVM. The modified training set is then used with another machine learning technique with explanation capability such as decision tree learner that learns what the SVM has learned.

The main steps involved in such techniques are;

SVM Training

Train an SVM and get the SVM model.

Rule Generation

- Use the SVM to predict the class of the training set examples, therefore an artificial dataset (AD) is generated.
- Use the artificial dataset (AD) to train a C5 decision tree, or any other algorithm which generates a comprehensible model. Hence, rules representing the concepts learned by the SVM are generated.

Rule Testing

Use the generated rules to classify another independent test set to compute rule accuracy and fidelity.

The most commonly used decision tree learners are modified TREPAN (Browne et al. 2004), C5 (Quinlan, 1993), REX (Markowska-Kaczmar and Trelak, 2003) and CART (Brieman et al., 1994). However, the methods in this category are not classifier-specific as

they do not utilize any model-specific components in rule extraction. The aim is simply to model the output of the original classifier using another classifier with better comprehensibility. It is observed that relatively high accuracy and high fidelity rules are obtained using these methods.

In a related study, He et al., (2006) suggested SVM_DT as a method for interpreting prediction of protein secondary structure. The proposed SVM_DT method is motivated by the need to integrate the good generalization performance of SVMs and the comprehensibility of decision trees. In particular, the method utilizes an SVM as a preprocessing step for decision tree. The authors argue that the use of an SVM as a preprocessing step can generate better and/or cleaner data than the original dataset, where some bad ingredients and weak cases can be reduced. This argument is valid, if optimum training parameters of SVM can be found. The authors also argue that the extracted rules have biological meaning and can be interpreted (He et al., 2006).

2.1.3 Eclectic/Hybrid Approaches

Methods in this group extract rules utilizing the internal working of the SVM and also consider SVM as a black box. During eclectic approaches, first SVM model is developed and predictions for support vector instances are obtained and only the modified support vectors are then used for generating rules.

Barakat and Diederich, (2005) proposed an eclectic approach for rule extraction from SVM. They used only the support vectors of the data and replaced the target values of the support vectors using the SVM model. The modified data is then used to train decision tree algorithm and the rules are generated. They reported their results on benchmark datasets only and concluded that using support vector instances results in reduction of the dataset size and the number of rules extracted. Later, Barakat and Bradely (2006) reported AUC as one of the important measure to evaluate the efficiency of the rule extraction algorithm.

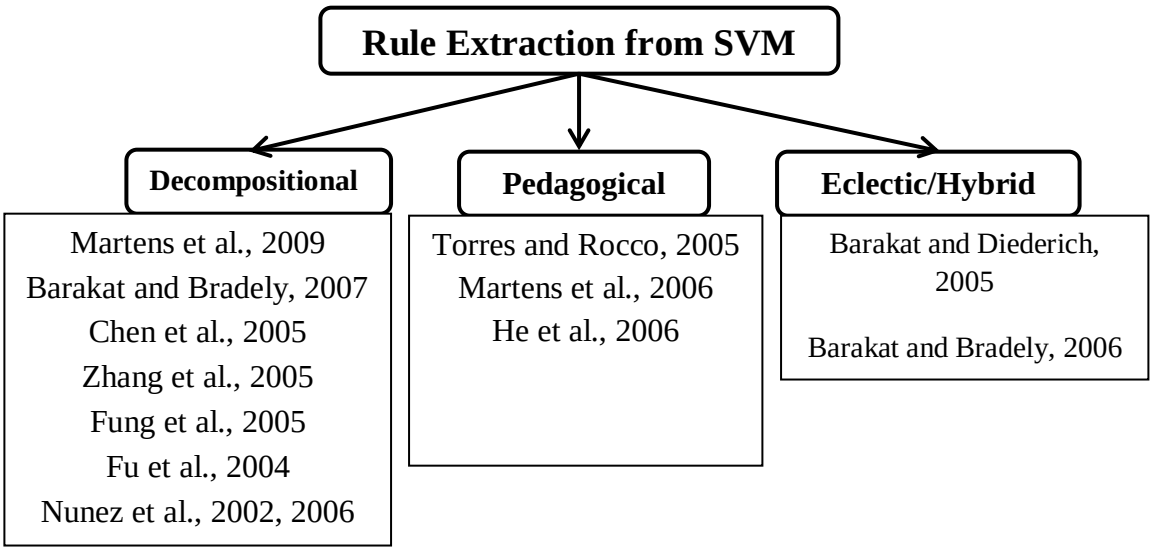


Figure 2.5: Taxonomy of the literature on Rule Extraction from SVM

2.2 Gaps Identified in Literature

Figure 2.5 presents the taxonomy of the rule extraction approaches proposed for extracting rules from SVM. It is observed that most of the efforts have been put to extract rules using SVM's internal workings or using hyper-plane learned by SVM during training i.e. the decompositional approaches.

During earlier research of rule extraction from SVM, researchers have focused on extracting rules for classification problems only. Most of the attempts solved benchmark classification problems and focused on the rule extraction strategies to extract rules. They argued that even if accuracy is low, having a transparent model to represent the knowledge learnt by SVM during training is acceptable (Zhang et al., 2005).

In literature no rule extraction approach is reported by the researchers for solving regression problems. Further, only decision tree algorithm is employed for extracting rules from SVM. Small scale medical problems were solved using decision tree for rule generation (Barakat and Diederich, 2004a, 2004b, 2005; Torres and Rocco 2005; Martens et al., 2006, 2009; He et al., 2006).

While researchers analyzed the efficiency of SVM for solving classification problems, the efficiency of SVM for feature selection in the context of rule extraction was totally ignored during earlier research.

Researchers ignored the importance of support vectors and did not explore the efficiency of SVM for solving medium scale unbalanced data mining problems. Real world problems deal with data which are medium scale and highly unbalanced in nature. Standard machine learning approaches are biased towards learning better about majority class instances than that of learning about minority class instances.

It is observed that efficiency of SVM is explored in the form of support vectors extraction and SVM-RFE for feature selection separately but the efficiency of SVM for support vectors extraction and feature selection simultaneously was totally ignored in earlier research, which reduces the data vertically and horizontally, respectively. This strength is unique to SVM among all other intelligent techniques.

2.3 Outline of Proposed Approaches

During the research work presented in this thesis, various rule extraction approaches are proposed to solve various finance problems such as, bankruptcy prediction in banks, churn prediction in credit card customers and fraud detection in automobile insurance. In this thesis decompostional and pedagogical approaches are proposed to solve bankruptcy prediction problem and eclectic/hybrid approaches are proposed to solve churn prediction in bank credit card customers, fraud detection in automobile insurance and regression problems. Rule extraction from SVM to solve churn prediction in bank credit card customers' problem is an important study of analytical customer relationship management (CRM) in finance. Figure 2.6 presents the classification of the proposed approaches into decompositional, pedagogical and eclectic approaches of rule extraction.

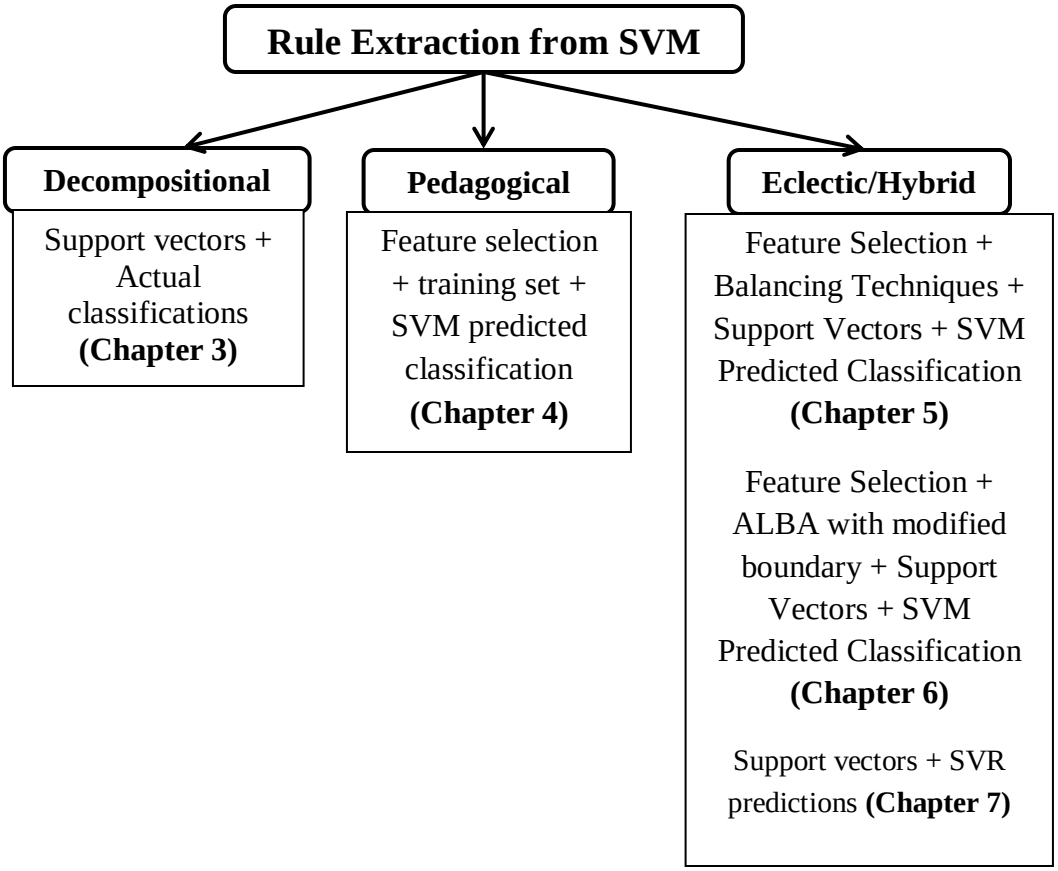


Figure 2.6: Classification of the proposed approaches for Rule Extraction from SVM

Part I

Decompositional Approach

Chapter 3

Fuzzy Rule Extraction using SVM for Bankruptcy Prediction in banks

This Chapter presents the proposed *decompositional rule extraction approach* for SVM which is one of the main contributions of this thesis. First section of the chapter provides the motivation behind extracting fuzzy rules for solving bankruptcy prediction problems. In the second section details about the proposed approach are presented. Third section provides the literature survey of bankruptcy prediction. Datasets used and Results and discussions is presented in the following sections and final section presents the conclusions of the chapter.

3.1 Motivation

As discussed in the first chapter, SVM produces most generalized model but does not represent the knowledge learnt by it during training in a human comprehensible form. In this chapter a decompositional rule extraction algorithm is proposed to extract fuzzy rules using Fuzzy Rule Based Systems called, SVM+FRBS. Fuzzy logic (Zadeh 1965) appears to be very well suited for the creation of small rules as fuzzy rules have a higher 'information density' i.e. each rule encapsulates a richness of information and meaning.

Rule: IF a person is a *"heavy_smoker"*
THEN the risk of cancer is *"high"*

where the two fuzzy concepts *"heavy_smoker"* and *"high"* can be represented by their membership function values. The interpretability of SVM is improved by fuzzy *if-then* rules. Bankruptcy prediction in banks problem is solved using the proposed rule extraction approach. When we solve financial problems, the comprehensibility of the system plays a

vital role and fuzzy rules provide better human understandability and comprehensibility of the system. It is observed that fuzzy rules are easier to understand and using which management can make better policies to avoid heavy losses to the organization. It is to be noted here that the datasets we analyze are very small sized. However, this application can be extended to medium scale or large scale datasets as well.

3.2 Proposed Fuzzy Rule Extraction Technique

During this research study we propose a novel decompositional rule extraction using SVM approach and applied it to predict bankruptcy in banks. FRBS is employed for rule generation purpose which extracts fuzzy rules leading to a significant improvement in system understanding and prediction accuracy as well. DT and ANFIS also employed for rule generation purposes. The hybrid is carried out in two steps;

1. Extraction of support vectors from SVM.
2. Support vectors are then fed to FRBS for fuzzy rule extraction.

3.2.1 Extraction of support vectors from SVM

Various kernels are employed for training SVM and considering the accuracy of SVM as driving force, the SVM model with best accuracy is selected. Support vectors are then extracted using the best model obtained, resulting in *Case-SA* dataset. *Case-SA* dataset is the new dataset consists of support vectors with their actual target values. Support vectors are the instances which are lying on the boundary of SVM and observed to be the right instances to learn the decision boundary. The instances away from the boundary do not add any value to decision boundary learning process of SVM. The extracted support vector set i.e. *Case-SA* dataset is then fed to FRBS, DT and ANFIS for rule generation.

3.2.2 Rule Generation

Extracted support vector set is then fed to FRBS, DT and ANFIS for rule generation purpose. Rules extracted using FRBS are presented in the results and discussion section.

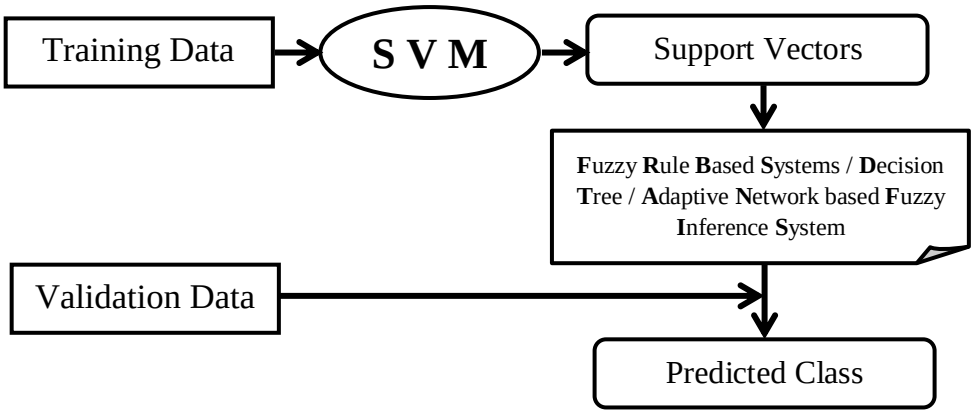


Figure 3.1: Work Flow Diagram of the proposed Hybrid Approach

3.3 Literature Review of Bankruptcy Prediction in Banks and Firms

Bankruptcy is a legally declared inability or impairment of ability of an individual or organization to pay its creditors. Since 1960 bankruptcy prediction in banks and corporate firms is the most researched area in the field of statistics and machine learning (Altman 1968). Prior prediction of bankruptcy helps top management to Figure out the causes of bankruptcy and implements different profiting policies and adopts various strategies towards stabilizing the firm's growth and stability. Bankruptcy of a financial firm affects its creditors, auditors, stock holders and senior management. Hence, they are all interested in bankruptcy prediction in the early stages only (Wilson and Sharda, 1994).

According to Federal Deposit Insurance Corporation Improvement (FDICI) act of 1991, regulators used CAMEL rating which evaluates banks financial health according to their basic functional areas *viz.*, ***C****apital adequacy,* ***A****sset quality,* ***M****anagement expertise,* ***E****arnings strength,* ***L****iquidity,* and ***S****ensitivity to market risk*. While CAMELS ratings clearly provide regulators with important information, Cole and Gunther (1995) reported that these CAMELS ratings decay rapidly**.** Fraser (1976) concluded that banks perform better by holding relatively more securities and fewer loans in their portfolios. Altman (1968) and Beaver (1966) pioneered the research in failure predictions of businesses. Altman (1968) investigated a set of financial and economic ratios in a bankruptcy prediction context. Korobow et al. (1976) stated that early warning systems are one that

challenges our understanding of the nation's financial system. It is clear that improved methods of early detection of financial weaknesses in banking system could help bank regulatory authorities to anticipate and mitigate future problems. Karels and Prakash (1987) rigorously investigated the normality conditions of financial ratios in the context of discriminant analysis.

Various statistical techniques such as regression analysis (Bell, 1997; Hu and Tseng, 2007) and logistic regression (Ohlson, 1980; Pantalone and Platt, 1987) have been used to predict the financial state of the company (healthy, distressed, high probability of bankruptcy) by making use of the company's financial data. In practice, these assumptions are rarely satisfied. Hence, researchers started applying non-parametric techniques, which cover the entire gamut of intelligent techniques.

Machine learning approached such as neural networks (Odom and Sharda, 1990; Tam, 1991; Salchenberger et al. 1992; Wilson and Sharda, 1994; Lee, et al., 1996; Jo, et al., 1997; Zhang et al. 1999; Baek and Cho, 2003; Charalambous et al., 1999; Charalambous et al., 2000; Swicegood and Clark 2001; Pai and Pai, 2004), Decision Tree (Tam and Kiang, 1992), Support Vector Machines (Gestel et al., 2003; Shin et al., 2005; Min and Lee, 2005), Genetic Algorithms (Shin and Lee, 2002), Genetic Programming (Etemadi et al., 2009), Fuzzy Rule Based Classifier (Ravikumar and Ravi, 2006a; 2006b), Rough set theory (McKee, 2000), Bayesian networks (Aghaie and Saeedi, 2009), Logistic Regression (Andres et al. 2005), Wavelett Neural Networks (Becerra et al. 2005), Semi online RBF (Ravi et al. 2008), Isotonic Separation (Ryu and Yu 2005), Integrated Early Warning System (IEWS) (Canbas et al. 2005), Autoassociative Neural Networks (Pramod and Ravi 2007) and Self Organizing Maps (SOM) (Nakaoka et al., 2006) are some of the applications for bankruptcy prediction in banks and firms. Further, researchers proposed hybrid approaches for bankruptcy prediction using two or more than two intelligent machine learning techniques, such as MDA assisted neural networks, ID3 assisted neural networks and SOM assisted neural network (Lee et al., 1996), GA and SVM (Min et al., 2006), ensemble method (Ravikumar and Ravi 2006 and 2007; Ravi et al. 2008; Sun and Li 2008; ; Chauhan, et al., 2008; Ravisankar, and Ravi 2009), etc.

A more detailed survey paper about the applications of neural networks for bankruptcy prediction is presented by Atiya (2000). A comprehensive review about bankruptcy prediction from 1968-2005 is presented by Ravikumar and Ravi (2007). Bankruptcy prediction using k-nearest neighbour (*k-nn*) weighted analytical hierarchy process (AHP) is proposed (Park and Han, 2002). They applied the proposed approach in Korean firms and concluded that *k-nn* with AHP outperform other models of prediction tested. Vieira et al. (2004) evaluated the efficiency of ANN, Linear Genetic Programming and SVM for bankruptcy prediction using balanced and unbalanced datasets. They concluded that LGP performs best with balanced datasets and SVM shows superior performance dealing with unbalanced datasets.

Shin and Lee (2002) provided a new dimension to the research towards bankruptcy prediction in firms and businesses and proposed a genetic algorithm based rule extraction approach. The generated rules can also be used as early warning system by the management and management can take proper and appropriate steps at right time towards the stability of the firm. Yuan, (2008) proposed a rule extraction approach with the integration of GA and its application to bankruptcy prediction problem.

At this juncture, researchers understood the need of applying fuzzy logic for bankruptcy prediction. Fuzzy logic models are understood better by the user than neural network (i.e. black box) models. The importance of fuzzy clustering combined with self organizing neural networks was studied by Alam et al., (2000). They reported that the proposed approach is a promising tool to predict potentially failing banks. Tung et al., (2004) highlighted the comprehensibility of the prediction process using neuro-fuzzy systems.

Ravikumar and Ravi (2006) have proposed a bankruptcy prediction approach based on fuzzy rule based classifier. They considered the classification problem as multiobjective combinatorial optimization problem, where their intension is to minimize the number of rules with increasing classification rate as well using Threshold Accepting algorithm. They concluded that with 2 partition the proposed approach outperformed MLP and obtained higher accuracy and lowest type-I error. Noguesia et al., (2005) proposed data reduction approach based on Fast Fuzzy Clustering algorithm. They reported that data preprocessing

using the proposed data reduction approach improves the performance of the classifier. It is observed that no rule extraction using SVM approach is proposed to represent the knowledge of SVM for solving bankruptcy prediction problem. In the present study a decompositional rule extraction technique is proposed to extract rules using SVM.

3.4 Results and Discussions

The proposed approach is tested on two benchmark datasets namely, iris and wine obtained from UCI machine learning repository, and three bankruptcy prediction in banks datasets *viz., Spanish Banks, Turkish Banks* and *US Banks*. In the following sections, description of the data and empirical study is presented. Bankruptcy datasets features' information is presented in Table 1 through Table 3 in Appendix C. We employed decision tree (DT) and adaptive network based fuzzy inference system (ANFIS) i.e. SVM+DT, SVM+ANFIS in order to evaluate the effectiveness and validity of the SVM+FRBS hybrid and SVM+RBF is chosen as a baseline hybrid even though we do not get any rules out of it.

Ishibuchi et al. (1999) examined the performance of a fuzzy genetic algorithm-based machine learning method for multidimensional pattern classification problem with continuous features, where each fuzzy if-then rule is handled as an individual and a fitness value is assigned to each rule. It is observed that grid based fuzzy partitions cannot handle high-dimensional data i.e. when we use the grid-type fuzzy partition with the increase in comprehensibility the number of fuzzy if-then rules increases exponentially as the number of input variables increase. *Don't care* antecedent is introduced to deal with the curse of dimensionality and to produce less number of fuzzy if-then rules by genetic operations. Data mining software KEEL is used to employ FRBS and the description of FRBS is given in Appendix A. DT is employed using RapidMiner software, ANFIS is employed in MATLAB and RBF is employed using Knime software tool. Details about DT and ANFIS are presented in Appendix A.

Table 3.1 presents the results and rules obtained using Iris and Wine datasets. Only FRBS and DT are employed to extract rules using iris and wine datasets. It is observed that using Iris dataset, hybrid classifier SVM+FRBS yielded 100% accuracy, while standalone FRBS

classifier yielded 96% accuracy. 6 rules are extracted for Iris dataset using FRBS and are presented in Table 3.2. It is observed that for iris-versicolor class two rules are generated, whereas for iris-virginica and iris-setosa classes only and three rules are generated, respectively. As regards the Wine dataset our proposed hybrids SVM+FRBS and SVM+DT yielded same accuracy of 97.14%. However, stand-alone FRBS yielded less percentage of accuracy of 94.29%. Sample rules extracted using wine data are presented in Table 3.3 and full rule set i.e. 14 rules is presented in Table D.1 in Appendix D. It is observed from the empirical results that rules extracted using FRBS are more generalized and the fuzzy rules improves the comprehensibility of the system. It is observed that from three different types of wines, the rules extracted for type A wine are four, for type B wine the number of rules is 10 and the number of rules extracted for type C wine are four.

Table 3.1: Average Results on Validation set

Classifiers	***I R I S***	***W I N E***
FRBS	96 %	94.29 %
SVM + FRBS	**100 %**	**97.14 %**
DT	93.33 %	100 %
SVM + DT	**93.33 %**	**97.14 %**

Table 3.2: Fuzzy Rules Extracted using Iris Dataset

#	Antecedent	Consequent
1	if "petal-width is medium"	Iris-Versicolor
2	if "petal-lenght is low"	Iris-Setosa
3	if "sepal-length is low" and "petal-width is low"	Iris-Setosa
4	if "petal_length is medium"	Iris-Versicolor
5	if "petal_length is high"	Iris-Virginica
6	if "sepal_length is low" and "petal_width is low"	Iris-Setosa

Table 3.3: Sample Fuzzy Rules Extracted using Wine Dataset

Rule #	**Antecedents**	**Consequent**
1	If "Flavanoids is High"	A
2	If "Alcohol is Low" and "Flavanoids is Medium"	B
3	If "Alcohol is Low"	B
4	If "Flavanoids is High" and " Hue is Medium"	A

In two class classification problems, sensitivity and specificity are calculated with accuracy of the system as discussed in the Section 1.5 of Chapter 1. Table 3.4 presents the number of samples in each dataset *viz.*, *Spanish*, *Turkish* and *US banks* datasets, after dividing the datasets into 80% (training) and 20% (validation). Results of *Spanish Banks* dataset presented in Table 3.5. The empirical study indicates that our hybrid method SVM+FRBS yielded 92.31% classification accuracy with 83.33% sensitivity and 100% specificity whereas the proposed approach SVM+RBF yielded best sensitivity of 100%. It is observed that all the classifiers tested yielded similar accuracies of 92.31%, but stand-alone FRBS yielded least accuracy of 69.23% with 33.33% sensitivity and 100% specificity. Despite yielding similar accuracy of 92.31%, it is observed that the hybrids SVM+FRBS, SVM+DT and SVM+ANFIS yielded low sensitivity. Such large variation is observed in the results because of small sized data. Fuzzy rules set extracted using SVM+FRBS approach is presented in Table 3.6. Equal number of rules i.e. 7 is extracted for bankrupt banks and non-bankrupt banks. It is observed from the rule set that *Reserve/Loans* and *Cash Flow/Loan* are very much important to predict bankruptcy in banks.

Table 3.4: Bankruptcy prediction dataset division into Training and Validation

Dataset	Features	Total instances			Training (80%)			Validation (20%)		
		Total	Bankrupt	Healthy	Total	Bankrupt	Healthy	Total	Bankrupt	Healthy
Spanish	9	66	29	37	53	23	30	13	6	7
Turkish	12	40	18	22	32	14	18	8	4	4
US	5	130	66	64	104	52	51	26	13	13

Table 3.5: Average results for Spanish Banks on Validation Set

Experiments	Accuracy %	Sensitivity %	Specificity %
RBF	92.31	83.33	100
SVM + RBF	**92.31**	**100**	**85.71**
Decision Tree	84.62	66.67	100
SVM + Decision Tree	**92.31**	**83.33**	**100**
ANFIS	92.31	100	85.71
SVM + ANFIS	**92.31**	**83.33**	**100**
Fuzzy Rule Base System	69.23	33.33	100
SVM + FRBS	**92.31**	**83.33**	**100**

Table 3.6: Fuzzy rules extracted for Spanish banks dataset

Rule #	**Antecedents**	**Consequent**
01	If "R/L is Medium" and "CF/L is Medium" then	Non-Bankrupt
02	If "CAC/TA is Medium" and "R/L is Low" and "CF/L is Medium"	Bankrupt
03	If "R/L is Medium" and "CF/L is Low"	Bankrupt
04	If "R/L is Low"	Bankrupt
05	If "CA/TA is Medium" and "R/L is Low"	Bankrupt
06	If "CF/L is Low"	Bankrupt
07	If "CA/TA is Medium" and "R/L is Medium" and "CF/L is Medium"	Non-Bankrupt
08	If "R/L is High"	Non-Bankrupt
09	If "CS/S is High"	Bankrupt
10	If "CA/TA is High" and "CF/L is Medium"	Non-Bankrupt
11	If "R/L is High" and "CF/L is High"	Non-Bankrupt
12	If "R/L is Low" and "CF/L is Medium"	Bankrupt
13	If "R/L is Medium" and "CF/L is High"	Non-Bankrupt
14	If "CA/TA is Medium" and "CAC/TA is Medium" and "R/L is Medium" and "CF/L is Medium"	Non-Bankrupt

Table 3.7 presents the results of *Turkish Banks* dataset. It is observed from empirical results that the hybrids SVM+RBF and SVM+ANFIS yielded 100% classification accuracy, sensitivity and specificity as well. It is observed that despite using reduced training set i.e. *Case-SA* dataset the hybrids SVM+RBF, SVM+ANFIS and SVM+FRBS yielded 100% sensitivity which is the one of the major achievement in the current study. It is observed that using all the training dataset, stand-alone DT and FRBS have shown poor performance with respect to accuracy and sensitivity. Using Turkish banks data three rules are generated for bankrupt banks and only one rule is extracted for non-bankrupt banks.

Table 3.7: Average results for Turkish Banks on Validation Set

Experiment	Accuracy %	Sensitivity %	Specificity %
RBF	100	100	100
SVM + RBF	**100**	**100**	**100**
Decision Tree	75	50	100
SVM + Decision Tree	**87.5**	**75**	**100**
ANFIS	100	100	100
SVM + ANFIS	**100**	**100**	**100**
Fuzzy Rule Base System	75	75	75
SVM + FRBS	**87.5**	**100**	**75**

It is observed from the rules extracted are that low *Interest Expenses/Average Profitable Assets* results non-bankrupt banks. For bankrupt banks it is observed that low Networking Capital/Total Assets, Standard Capital Ration or Interest Expenses/Average Non-profitable Assets results in bankruptcy. Fuzzy rules extracted for Turkish bank dataset using SVM+FRBS hybrid is tabulated in Table 3.8.

Table 3.8: Fuzzy Rules Extracted for Turkish Banks dataset

Rule #	**Antecedents**	**Consequent**
01	If "NC/TA is Low"	Bankrupt
02	If "SCR is Low"	Bankrupt
03	If "IE/APA is Low"	Non-Bankrupt
04	If "IE/ANA is Medium"	Bankrupt

Results of US banks dataset is presented in Table 3.9. It is observed that the proposed hybrid SVM+FRBS yielded good classification accuracy of 96.15% with 92.31% sensitivity and 100% specificity. Fuzzy rules extracted using US banks dataset are presented in Table 3.10. It is observed that except the hybrid SVM+ANFIS all the other hybrid yielded 92.31% accuracy. The hybrid SVM+DT yielded cent percent sensitivity, whereas other hybrids yielded less sensitivity. It is observed from the rules that 5 rules are extracted representing the behaviour of non-bankrupt banks, whereas 6 rules are extracted to represent the bankrupt banks nature. It is observed that low *Retained Earnings/Total Assets* and medium *Earnings before Interest and Taxes/Total Assets* plays vital role in bankruptcy prediction whereas high *Retained Earnings/Total Assets* and low *Sales/Total Assets* represents the behaviour of non-bankrupt banks.

Table 3.9: Average results for US Banks on Validation Set

Experiment	Accuracy %	Sensitivity %	Specificity %
RBF	100	100	100
SVM + RBF	92.31	92.31	92.31
Decision Tree	92.31	100	84.62
SVM + Decision Tree	**92.31**	**100**	**84.62**
ANFIS	92.31	92.31	92.31
SVM + ANFIS	88.46	84.62	92.31
Fuzzy Rule Base System	92.31	84.62	100
SVM + FRBS	**96.15**	**92.31**	**100**

Table 3.10: Fuzzy rules extracted for US banks dataset

Rule #	Antecedents	Consequent
01	If "RE/TA is High" and "MVE/TA is Medium"	Non-Bankrupt
02	If "RE/TA is High"	Non-Bankrupt
03	If "EIT/TA is High"	Non-Bankrupt
04	If "RE/TA is Low"	Bankrupt
05	If "RE/TA is High" and "EIT/TA is High"	Non-Bankrupt
06	If "RE/TA is Medium" and "EIT/TA is Medium" and "MVE/TA is Low"	Bankrupt
07	If "RE/TA is Medium" and "EIT/TA is Medium" and "S/TA is Medium"	Bankrupt
08	If "EIT/TA is Medium" and "S/TA is Medium"	Bankrupt
09	If "RE/TA is High" and "S/TA is Low"	Non-Bankrupt
10	If "WC/TA is Medium" and "EIT/TA is Low"	Bankrupt
11	If "EIT/TA is Low" and "S/TA is Medium"	Bankrupt

AUC (Area Under the ROC Curve) (refer Section 1.4 of Chapter 1) is calculated for all the classifiers and the hybrids and presented in Table 3.11. The idea of AUC is that bigger the area better the classifier. It is observed that using Spanish banks data it is SVM+RBF hybrid yielded best AUC of 9385.5, using Turkish banks data the hybrids SVM+RBF and SVM+ANFIS covered the whole area under ROC and using US banks data it is observed that the proposed hybrid SVM+FRBS yielded 9615.5 AUC. It is observed using AUC obtained using various classifiers tested for bankruptcy prediction is that they hybrids perform equally well with stand-alone approaches. It is observed that despite yielding marginally less accuracy the proposed approach SVM+FRBS extracted fuzzy rules with high understandability of the system.

Table 3.11: AUC of the classifiers

	Spanish Banks	Turkish Banks	USA Banks
DT	8333.5	7500	9231
SVM + DT	9166.5	8750	9231
RBF	9166.5	10000	10000
SVM + RBF	**9385.5**	10000	9231
ANFIS	9385.5	10000	9231
SVM + ANFIS	9166.5	**10000**	8846
FRBS	6666.5	7500	9231
SVM + FRBS	9166.5	8750	**9615.5**

A rule set is considered to display a high level of fidelity if it can mimic the behaviour of the machine learning technique from which it was extracted, in our study it is SVM. It is observed that our proposed approach displays high fidelity compared to other classifiers tested and the results are presented in Table 3.12. It is observed from the fidelity calculated that using Spanish banks data our proposed approaches SVM+FRBS and SVM+DT behave 83% exactly as of SVM. When using Turkish data it is observed that SVM+FRBS behave more likely as of SVM with 91.81% fidelity. Using US banks data the fidelity obtained by SVM+FRBS is 92.31% i.e. the proposed approach SVM+FRBS behaves 92.31% time as SVM. It is observed from fidelity calculation that fuzzy rules extracted using SVM behaves more like SVM itself.

Table 3.12: Fidelity of various hybrids

	Spanish	Turkish	US
SVM+DT	**83.04**	90.24	85.71
SVM+RBF	78.1	89.21	86.98
SVM+ANFIS	77.85	85.21	73.06
SVM+FRBS	**83**	**91.81**	**92.31**

3.5 Conclusions

It is well known that SVM produces black box model, meaning that it does not explicitly convey the knowledge learnt by it during training. Rule extraction techniques aim to remove this disadvantage by bringing in comprehensibility and interpretability to SVM. This chapter presents a novel hybrid approach for extracting fuzzy *if-then* rules by invoking SVM and FRBS in tandem. In order to test the validity and the effectiveness of the proposed hybrid, we tested it against other classifiers such as DT, ANFIS and RBF and hybrids such as SVM+DT, SVM+ANFIS and SVM+RBF. We used two benchmark datasets *viz.,* iris and wine, and three well-known bank bankruptcy datasets *viz., Spanish banks, Turkish banks* and *US banks* to carry out the experiments. It is observed that the hybrid classifier performs better or similar to the stand-alone classifiers on benchmark datasets. The results obtained using bankruptcy prediction datasets indicate that the proposed hybrid SVM+FRBS shows superior performance using *Spanish* and *US* bank

datasets. For Turkish bank dataset SVM+RBF surpasses other classifiers in terms of accuracy. The fidelity of SVM+FRBS is superior across banks compared to other approaches. It is concluded that hybrid classifier yielded better classification accuracy and comprehensible fuzzy rules compared to standalone classifiers. It is also observed that the rules extracted also show the behaviour of the features towards bankruptcy prediction and fuzzy rules improves the comprehensibility of the system.

Part II

Pedagogical Approach

Chapter 4

Rule Extraction from SVM using Feature Selection for solving classification and regression problems

This Chapter presents a *pedagogical rule extraction approach* proposed which uses SVM as feature selection algorithm and using reduced feature set rules are extracted. With the motivation in the first section, the proposed approach is presented in detail in next section. Applications of the proposed approach for solving classification and regression problems are discussed in the next section. Following section provides the detailed Results and discussions. Final section concludes the chapter.

4.1 Motivation

In machine learning and statistics, feature selection is the technique of selecting a subset of relevant features for building robust learning models. Feature selection eliminates redundant and irrelevant features and helps improve the performance of learning models by alleviating the effect of the curse of dimensionality. Further, it helps in enhancing generalization capability of the systems under development. Feature selection speeds up the learning process and improves model interpretability to higher level. Researchers argued that support vectors represent the knowledge learnt by SVM during training. During the present study we utilize feature selection aspect of SVM to represent the knowledge learnt by SVM during training and argued that feature selection using SVM also represents the knowledge learnt by SVM during training. It is also observed that rules extracted using reduced feature are less in number and smaller in size, resulting in improved comprehensibility of the system without compromising the accuracy of the system. Classification and regression problems are solved using the proposed approach, where DT, CART are used to generate rules for classification problems and CART, ANFIS

and DENFIS are employed to generate rules for regression problems. Bankruptcy prediction problems and benchmark regression problems are solved using the proposed rule extraction approach.

4.2 Proposed Rule Extraction Approach

The proposed pedagogical rule extraction approach consists of three major steps. The first two steps of the process are illustrated in Figure 4.1 and third step is shown in Figure 4.2.

1. *Feature selection using SVM-RFE,*
2. *Building SVM/SVR models and*
3. *Rule generation*.

4.2.1 Feature Selection using SVM-RFE

SVM-RFE (Recursive Feature Elimination) (Guyon, 2002) algorithm is employed for feature selection purpose. Nested subsets of features are selected in a sequential backward elimination manner, which starts with all the feature variables and removes one feature variable at a time. At each step, the coefficients of the weight vector w of a linear SVM are used to compute the feature ranking score. The feature say, the i^{th} feature with the smallest ranking score $c_1 = (w)^2$ is eliminated, where w represents the corresponding component in the weight vector w. Using $c = (w)^2$ as the ranking criterion corresponds to removing the feature whose removal changes the objective function the least. This objective function is chosen to be $J = ½ ||w||^2$ in SVM-RFE. Appendix A presents a more detailed description of SVM-RFE algorithm.

4.2.2 Building SVM/SVR Models

SVM and SVR models are developed for classification and regression tasks respectively. Classification accuracy is the main driving element for developing the SVM model. The predictive accuracy measured in terms of Root Mean Squared Error (RMSE) for regression problems is the driving element to build SVR model. The developed models are then used for predicting the target values of the training instances. The actual target values of the training set are then replaced by the predictions of SVM/SVR models and a new modified

training set *Case-P* dataset is obtained. *Case-P* dataset is modified training set with corresponding predicted target values of the training instances where the predictions are obtained using SVM/SVR models.

4.2.3 Rule Generation

During rule generation phase we analyzed *Case-P* dataset. Rules are generated using two machine learning techniques viz. DT and CART and two soft computing techniques viz. ANFIS and DENFIS. Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data (see Section 1.7 in Chapter 1). Generated rules are then tested against the validation set and the results are presented in Results and discussions section as explained in section 1.7 of chapter 1. Prediction accuracy of the rules is determined in terms of accuracy for classification problems and RMSE for regression problems on validation set.

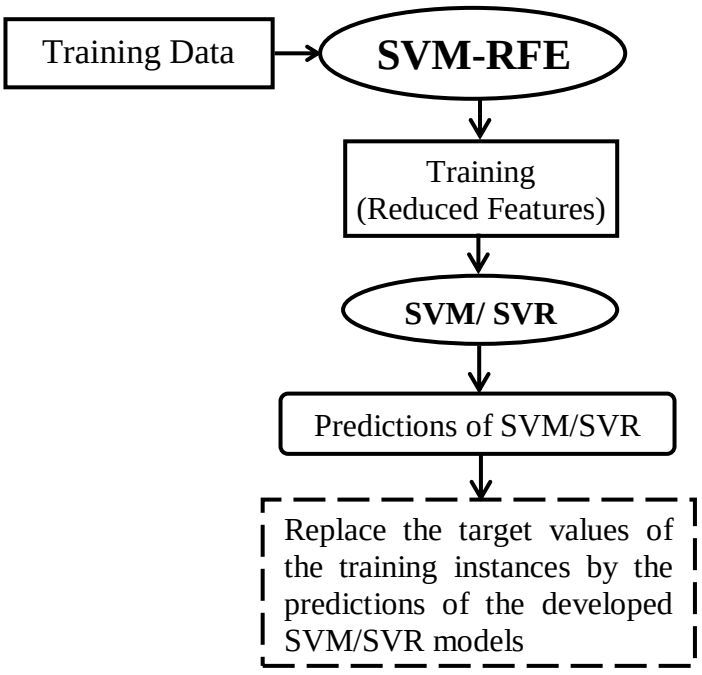


Figure 4.1: First Phase of the proposed hybrid (Predictions of SVM/SVR)

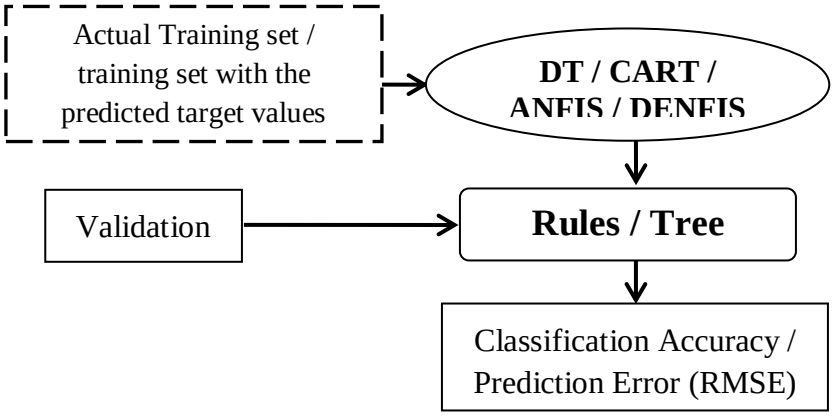


Figure 4.2: Rule generation phase

Note: In the earlier study we used only support vectors i.e. *Case-SA* dataset, here in the present study we used all the training data i.e. *Case-P*.

4.3 Problems Analyzed

We chose publicly available benchmark datasets such as *iris* and *Wine*, medical diagnosis dataset i.e. *Wisconsin Breast Cancer*, bankruptcy prediction in banks datasets *viz.*, *Spanish banks, Turkish banks, US banks* and *UK banks* datasets as classification problems. Classification datasets analysed in this research study are presented in Table 4.1. Information about the classification datasets analyzed is presented in Appendix C. Table 4.2 presents the details of the division of the datasets into Training and Validation sets. For regression analysis datasets are obtained from UCI machine learning repository (http://archive.ics.uci.edu/ml/) and StatLib (http://lib.stat.cmu.edu/datasets/) repository.

Table 4.1: Classification dataset Information

Dataset	Total instances	Features	# of Classes
IRIS	150	4	3
Wine	178	13	3
WBC	683	9	2
Spanish Banks	66	9	2
Turkish Banks	40	12	2
US Banks	129	5	2
UK Banks	60	10	2

Table 4.2: Bankruptcy prediction dataset division into Training and Validation

Dataset	Features	Total instances			Training (80%)			Validation (20%)		
		#	Bankrupt	Healthy	#	Bankrupt	Healthy	#	Bankrupt	Healthy
Spanish	9	66	29	37	53	23	30	13	6	7
Turkish	12	40	18	22	32	14	18	8	4	4
US	5	130	66	64	104	52	51	26	13	13
UK	10	60	30	30	48	24	24	12	6	6

The datasets viz., *Auto MPG, Body Fat, Boston Housing, Forest Fires* and *Pollution* are used to evaluate the efficiency of the proposed approach for solving regression problems.

4.3.1 Auto MPG dataset

This dataset concerns city-cycle fuel consumption in *miles per gallon*. This dataset contains 398 instances with eight features. This dataset is available in UCI machine learning repository (Asuncion and Newman 2007). The features are described in Table C.5 in Appendix C.

4.3.2 Body Fat dataset

This dataset is obtained from StatLib repository http://lib.stat.cmu.edu. It estimates the *percentage of body fat* determined by underwater weighing and various body circumference measurements for 252 men (Penrose, Nelson and Fisher 1985). Its features are described in Table C.6 in Appendix C.

4.3.3 Boston Housing dataset

This dataset is obtained from UCI machine learning repository (Asuncion and Newman 2007). It concerns housing values in suburbs of Boston and contains 506 instances with 17 features. Table C.7 in Appendix C presents the feature description.

4.3.4 Forest Fires dataset

This dataset is also obtained from UCI machine learning repository (Asuncion and Newman, 2007). This is a difficult regression task where the aim is to predict the *burned area of forest* in the northeast region of Portugal by using meteorological and other data.

This dataset contains 517 instances with 13 features and the features information is described in Table C.8 in Appendix C.

4.3.5 Pollution dataset

This dataset is obtained from StatLib repository http://lib.stat.cmu.edu. This dataset contains 60 instances with 16 features (McDonald and Schwing, 1973). Table C.9 in Appendix C presents the description of features.

4.4 Results and Discussions

MATLAB is used for SVM/SVR and ANFIS algorithms. DENFIS is implemented under the software package Neucom_Student (www.theneucom.com) and CART (Brieman et al. 1984) is implemented as a package available at (http://www.salford-systems.com/). The results obtained using all the classification datasets with and without feature selection are presented in Table 4.3 through Table 4.17. Results using regression data with full features and with reduced features are tabulated in Table 4.18 through Table 4.27. 10-fold cross validation is performed throughout the study, and the results obtained against the validation set are presented here.

4.4.1 Classification Problems

DT and CART are employed to generate rules for solving classification problem. Table 4.3 presents the average results for iris data. It is observed that the proposed hybrid SVM+CART using *Case-P* dataset obtained better accuracy of 94.67% which represents the knowledge learnt by SVM. One rule for each class of iris flower is obtained. It is observed that the corresponding hybrids SVM+DT and SVM+CART yielded slightly less accuracy than that of stand-alone DT and CART, respectively. The rules extracted using SVM+CART using *Case-P* dataset is presented in Table 4.4. It is also observed that SVM+CART yielded better results compared to SVM+DT hybrid. It is observed that decompositional rule extraction approach presented in chapter 3 yielded 6 rules for iris dataset (see Table 3.1 and Table 3.2 of Chapter 3) whereas in the present study only 3 rules are extracted with acceptable decrease in the accuracy resulting in better

comprehensibility. Considering accuracy of the system, it is observed that the approach presented in Chapter 3 perform better than that of the approach presented in this Chapter.

Table 4.3: Average results obtained using IRIS data on validation set

Technique	**Accuracy**
SVM	96.33
DT	93
SVM+DT (*Case-P)*	92
CART	95.33
SVM+CART (*Case-P)*	**94.67**

Table 4.4: Rule set extracted using SVM+CART (Case-P) for Iris data (all features)

Rule #	**Antecedents**	**Class**
1	If *Petal-Length* <= 2.45	Iris – Setosa
2	If *Petal-Length* > 2.45 and *Petal-Width* <= 1.75	Iris – Versicolor
3	If *Petal-Length* > 2.45 and *Petal-Width* > 1.75	Iris – Virginica

Average results obtained using Wine dataset are presented in Table 4.5. It is observed that using full featured data, SVM+CART yielded 94.86% accuracy. Three different sets of reduced features i.e. 7, 8 and 9 features are formed and the experiments are repeated again. The most important features selected in *wine* datasets are *Flavanoids, Proline, Color Intensity, Alcohol, Hue, OD280/OD315 of Diluted Wines, Malic Acid, Alcalinity of Ash* and *Ash*. It is observed from the empirical results that proposed approach with reduced feature yielded more than 94% accuracy. Rules obtained from SVM+CART using *Case-P* dataset with 8 features are presented in Table 4.6. It is observed from the Results and discussions of Chapter 3 that the proposed decompositional approach yielded 14 rules using wine dataset (see Table 3.1 and Table 3.3 of Chapter 3), whereas in the current study the proposed pedagogical approach with reduced features yielded 4 rules only. Considering accuracy of the system, the approach presented in this chapter yielded low accuracy compared to the approach presented in Chapter 3.

Table 4.5: Average results on Validation set using all and reduced features of Wine data

# Features	Technique	Accuracy
All Features	SVM	94.29
	DT	94.83
	SVM+DT (*Case-P)*	93.72
	CART	98.86
	SVM+CART (*Case-P)*	**94.86**
7	SVM	94.58
	DT	95.71
	SVM+DT (*Case-P)*	93.43
	CART	99.14
	SVM+CART (*Case-P)*	**94**
8	SVM	94.58
	DT	95.71
	SVM+DT (*Case-P)*	93.43
	CART	99.14
	SVM+CART (*Case-P)*	**94.57**
9	SVM	94.58
	DT	95.71
	SVM+DT (*Case-P)*	93.43
	CART	99.14
	SVM+CART (*Case-P)*	**94.57**

Table 4.6: Rule set extracted using SVM+CART (Case-P) for Wine data (8 features)

Rule #	Antecedents	Consequent
1	If *Proline* <= 842.5 and *OD280/OD315 Of Diluted Wines* <= 2.125 and *Alcalanity Of Ash* <= 18.25	B
2	If *Proline* <= 842.5 and *OD280/OD315 Of Diluted Wines* <= 2.125 and *Alcalanity Of Ash* > 18.25	C
3	If *Proline* <= 842.5 and *OD280/OD315 Of Diluted Wines* > 2.125	B
4	If *Proline* > 842.5	A

Dealing with application problems like medical diagnosis and bankruptcy prediction, sensitivity and specificity of the classifier is also calculated, where sensitivity is the percentage of number of correctly classified bankrupt banks to the total number of bankrupt banks and the specificity is the percentage of number of correctly classified non-bankrupt banks to the total number of non-bankrupt banks. Receiver Operating

Characteristics (ROC) graph (Fawcett 2006) is to depict the trade-off between sensitivity and false positive rates of classifiers. The idea behind area under Receiver Operating Characteristics curve is that "bigger the area under curve, better the classifier is".

Average results obtained using WBC dataset are presented in Table 4.7. It is observed that the proposed hybrid SVM+CART using *Case-P* dataset outperformed other classifiers, yielding 77.86% accuracy, 77.86% sensitivity, 76.67% specificity and AUC of 7726.5 considering all the features. The most important features selected are *Uniformity of Cell Size, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Mitoses* and *Uniformity of Cell Shape*. Using 5 most important features the hybrid SVM+CART using *Case-P* yielded accuracy of 74.02%, sensitivity of 71.11%, specificity of 74.09% and AUC of 7260. It is observed from the t-distribution values that the stand alone CART and other classifiers are statistically not significant. Rules extracted using SVM+CART with 5 important features are presented in Table 4.8.

Using AUC criteria the classifiers are compared with t-test at $n_1+n_2-2=10+10-2=18$ degrees of freedom the classifiers are compared at 10% level of significance. We tested if the difference in performances is statistically significant. The value of t-test for 18 degrees of freedom at 10% level of significance is 1.73. That means, if the t-test value between two different classifiers is more than 1.73, then we can say that the difference between techniques is statistically significant and otherwise not significant. Final column in the results table presents the t-test values obtained considering AUC of the classifier tested. The classifier selected as best is presented with hyphen. It is observed from t-test values that our proposed hybrid approaches perform similar to the stand-alone classifiers. 4 rules were extracted for normal cases and 3 rules were extracted for cancer cased. It is also observed from the rules that only 3 rules for cancer cases yielded better sensitivity.

Table 4.7: Average results on Validation set using all and reduced features of WBC data

# Features	Technique	Accuracy	Sensitivity	Specificity	AUC	t-test
All Features	SVM	79.476	79.48	71.11	7529.5	
	DT	73.083	78	73.41	7570	0.2849
	SVM+DT (*Case-P)*	77.444	77.44	72.73	7568.6	0.2686
	CART	74.964	78	73.41	7570	0.2849
	SVM+CART (*Case-P)*	**77.86**	**77.86**	**76.67**	**7726.5**	**-**
4	SVM	68.498	56.91	70.81	6386	
	DT	71.951	53.02	77.05	6503.5	1.4591
	SVM+DT (*Case-P)*	72.987	52.55	77.49	6502	1.4788
	CART	73.986	75.78	73.07	7442.5	-
	SVM+CART (*Case-P)*	**73.983**	**70.91**	**75.34**	**7312.5**	**0.2055**
5	SVM	66.691	54.32	67.02	6067	
	DT	71.579	52.18	75.57	6387.5	1.6445
	SVM+DT (*Case-P)*	71.385	53.09	74.77	6393	1.6467
	CART	74.061	75.56	73.30	7443	-
	SVM+CART (*Case-P)*	**74.02**	**71.11**	**74.09**	**7260**	**0.2894**
6	SVM	69.325	61.68	70.11	6589.5	
	DT	70.978	50.36	75.80	6308	1.7758
	SVM+DT (*Case-P)*	67.719	48.68	71.93	6030.5	2.2322
	CART	73.76	76.67	72.27	7447	-
	SVM+CART (*Case-P)*	**72.5**	**70.67**	**74.32**	**7249.5**	**0.3142**

Table 4.8: Rule set extracted using SVM+CART (Case-P) for WBC data (5 features)

Rule #	Antecedents	Consequent
1	If *Bland Chromatin* <= 2.5 and *Uniformity of Cell Size* <= 1.5 and *Bare Nuclei* <= 1.5 and *Single Epithelial Cell Size* <= 2.5	Cancer
2	If *Bland chromatin* <= 2.5 and *Uniformity of Cell Size* <= 1.5 and *Bare Nuclei* <= 1.5 and *Single Epithelial Cell Size* > 2.5	Normal
3	If *Bland chromatin* <= 2.5 and *Uniformity of Cell Size* <= 1.5 and *Bare Nuclei* > 1.5	Normal
4	If *Uniformity of Cell Size* > 5.5 and *Single Epithelial Cell Size* <= 4.5 and *Bland chromatin* < 5.5	Normal
5	If *Uniformity of Cell Size* > 5.5 and *Single Epithelial Cell Size* <= 4.5 and *Bland chromatin* > 5.5 and *Bland chromatin* <= 9.5	Cancer
6	If *Uniformity of Cell Size* > 5.5 and *Bland chromatin* > 2.5 and *Bland chromatin* <= 9.5 and *Single Epithelial Cell Size* > 4.5	Normal
7	If *Uniformity of Cell Size* > 5.5 and *Bland chromatin* > 9.5	Cancer

Results obtained using *Spanish banks* data are presented in Table 4.9. Using *Spanish banks* dataset it is observed from the results that the hybrids SVM+DT and SVM+CART using *Case-P* perform better than their corresponding standalone DT and CART approaches using. 6 most important feature selected are *Reserves/Loans, Cost of Sales/Sales, Cash Flows/Loans, Current Assets-Cash/Total Assets, Current Assets/Total Assets* and *Net Income/Total Equity Capital*. The proposed hybrid SVM+CART using *Case-P* with 6 most important features yielded the best accuracy of 94.62% with 86.66% sensitivity, 100% specificity and AUC of 9333. Table 4.10 presents the rules extracted from SVM+CART using *Case-P* with 6 most important features. It is observed in the current study is that only two rules are extracted which yielded 94.62% accuracy, whereas in the previous study (see Chapter 3) the highest accuracy yielded was 92.31% only and the number of rules extracted is 14.

Table 4.9: Average results using all and reduced features of Spanish banks data

# Features	Technique	Accuracy	Sensitivity	Specificity	AUC	t-test
All Features	SVM	92.31	83.33	100	9166.5	
	DT	85.39	68.34	100	8417	4.8229
	SVM+DT (*Case-P)*	89.23	76.67	100	8833.5	2.0764
	CART	91.54	83.33	98.18	9075.5	1.3816
	SVM+CART (*Case-P)*	**92.31**	**83.33**	**100**	**9166.5**	-
4	SVM	87.69	76.67	96.37	8652	
	DT	86.16	70.00	100	8500	4.9195
	SVM+DT (*Case-P)*	86.16	81.67	92.86	8726.5	2.1755
	CART	91.54	81.66	100	9083	0.7171
	SVM+CART (*Case-P)*	**92.31**	**83.33**	**100**	**9166.5**	-
5	SVM	86.09	73.33	96.37	8485	
	DT	86.16	70.00	100	8500	4.9195
	SVM+DT (*Case-P)*	92.31	83.33	100	9166.5	0
	CART	91.54	81.66	100	9083	0.7171
	SVM+CART (*Case-P)*	**92.31**	**83.33**	**100**	**9166.5**	-
6	SVM	92.31	83.33	100	9166.5	
	DT	86.16	70.00	100	8500	5.46
	SVM+DT (*Case-P)*	92.31	83.33	100	9166.5	1.51
	CART	91.54	81.66	100	9083	1.81
	SVM+CART (*Case-P)*	**94.62**	**86.66**	**100**	**9333**	**-**

It is observed that feature selection using SVM-RFE not only improves the accuracy of the system but also yields less number of rules resulting in high comprehensibility. Based on t-test value it is observed that our proposed approach turned out to be statistically significant and yielded the best accuracy as well.

Table 4.10: Rule set extracted using SVM+CART (*Case-P*) for Spanish banks data (6 features)

Rule #	**Antecedents**	**Consequent**
1	if *CS/S* <= 0.89835	Non-Bankrupt
2	if *CS/S* > 0.89835	Bankrupt

Table 4.11 presents the results obtained using Turkish banks dataset. It is observed from the results that the hybrid SVM+DT using *Case-P* dataset with full features yielded best results among the hybrids extracting rules. The proposed hybrid SVM+DT with full features yielded 86.25% accuracy, 72.5% sensitivity, 100% specificity and 8625 AUC. The most important 8 selected features are *Interest Expenses/Average Profitable Assets, (Share Holders' Equity + Total Income)/(Deposits + Non-Deposit Funds), Interest Income/Interest Expenses, (Share Holders' Equity + Total Income)/Total Assets, Networking Capital/Total Assets, Interest Expenses/Total Expenses, Interest Expenses/Average Non-Profitable Assets* and *(Share Holders' Equity + Total Income)/(Total Assets + Contingencies and Commitments).* It is observed based on the t-test values obtained is that the classifiers behave similarly i.e. statistically not significant.

Table 4.12 shows the rule set extracted using the hybrid SVM+DT using *Case-P* considering 8 most important features. It observed that number of rules extracted is very low, only 2 rules are extracted for bankrupt banks yielding the sensitivity of 62.5%. It is observed that the proposed approach in Chapter 3 yielded 100% accuracy whereas in this study the proposed approach yielded highest accuracy of 86.25% only. As Turkish dataset is smallest in this category, no much difference is observed between rules extracted in Chapter 3 and rules extracted in current study in this chapter.

Table 4.11: Average results using all and reduced features of Turkish banks data

# Features	Technique	Accuracy	Sensitivity	Specificity	AUC	t-test
All Features	SVM	86.25	82.5	90	8625	
	DT	86.25	72.5	100	8625	0.2898
	SVM+DT (*Case-P)*	**86.25**	**72.5**	**100**	**8625**	**0.4114**
	CART	88.75	72.5	100	8625	-
	SVM+CART (*Case-P)*	85	75	87.5	8125	2.1302
6	SVM	87.5	75	100	8750	
	DT	85	72.5	100	8625	0.2898
	SVM+DT (*Case-P)*	80	65	95	8000	2.5797
	CART	87.5	75	100	8750	-
	SVM+CART (*Case-P)*	78.75	65	95	8000	2.2043
7	SVM	81.25	70	92.5	8125	
	DT	90	77.5	100	8875	-
	SVM+DT (*Case-P)*	73.75	52.5	100	7625	3.1924
	CART	87.5	75	100	8750	0.3349
	SVM+CART (*Case-P)*	76.25	52.5	100	7625	3.1924
8	SVM	87.5	75	100	8750	
	DT	81.25	62.5	100	8125	1.6214
	SVM+DT (*Case-P)*	81.25	65	97.5	8125	1.8246
	CART	87.5	75	100	8750	-
	SVM+CART (*Case-P)*	80	62.5	97.5	8000	2.2043

Table 4.12: Rule set extracted using SVM+DT (*Case-P*)
for Turkish banks data (8 features)

Rule #	Antecedents	Consequent
1	If *NC/TA* <= -3.2	Bankrupt
2	If *NC/TA* > -3.2 and II/IE > 143.9	Non-Bankrupt
3	If *NC/TA* > -3.2 and II/IE <= 143.9	Bankrupt

As the features available for US banks data are 5 only, we did not employ feature selection for this dataset. Instead *Case-P* dataset is analyzed during the study in this chapter. Average results obtained using US banks data are presented in Table 4.13. It is observed from the empirical results that the proposed hybrid SVM+CART using *Case-P* dataset yielded 91.92% accuracy, 94.62 sensitivity, 86.93% specificity and 9077.2 AUC. It is observed that using *Case-SA* dataset (see Chapter 3) the number of rules extracted are more but yielded better accuracy of 96.15% whereas despite extracting less rules the proposed approach in the current study did not yield better accuracy. The rules extracted from the

hybrid SVM+CART using *Case-P* are presented in Table 4.14. The t-test values indicate that the difference in the classifiers is statistically insignificant i.e. classifiers performed similarly.

Table 4.13: Average results of US banks dataset on Validation set

Technique	Accuracy	Sensitivity	Specificity	AUC	t-test
SVM	93.46	95.39	91.54	9346.15	
DT	92.31	94.62	90.00	9231	0
SVM+DT (*Case-P)*	91.15	94.62	86.93	9077.2	0.912
CART	92.31	94.62	90.00	9231	-
SVM+CART (*Case-P)*	**91.92**	**94.62**	**86.93**	**9077.2**	**0.912**

Table 4.14: Rule set extracted using SVM+CART (*Case-P*) for US banks data (all features)

Rule #	Antecedents	Consequent
1	if *RE/TA* <= 0.2537 and *ME/TA* <= 0.5483	Bankrupt
2	if *ME/TA* > 0.5483 and *RE/TA* <= 0.1478 and *EIT/TA* <= 0.1649	Bankrupt
3	if *ME/TA* > 0.5483 and *RE/TA* <= 0.1478 and *EIT/TA* > 0.1649	Non-Bankrupt
4	if *ME/TA* > 0.5483 and *RE/TA* > 0.1478 and *RE/TA* <= 0.2537	Non-Bankrupt
5	if *RE/TA* > 0.2537 and *EIT/TA* <= -0.00145	Bankrupt
6	if *RE/TA* > 0.2537 and *EIT/TA* > -0.00145	Non-Bankrupt

Table 4.15 presents the average results obtained using UK bank dataset. It is observed from the results that the proposed hybrid SVM+CART using all features yielded 72.5% accuracy with 70% sensitivity, 73.33% specificity and 7166.8 AUC. t-test statistics indicate that all the classifiers tested using UK banks data are statistically insignificant, i.e. every classifier is performing similar.

Table 4.15: Average results on Validation using all and reduced features of UK banks data

# Features	Technique	Accuracy	Sensitivity	Specificity	AUC	t-test
All Features	SVM	76.66	73.33	80	7666.5	
	DT	73.33	73.33	71.67	7250	-
	SVM+DT (*Case-P)*	70	70	70	7000	0.3367
	CART	73.33	60	80	7000	0.3141
	SVM+CART (*Case-P)*	**72.5**	**70**	**73.33**	**7166.8**	**0.1121**
5	SVM	66.67	63.34	70	6667	
	DT	60	63.33	65	6416.5	0.9854
	SVM+DT (*Case-P)*	**66.67**	**55**	**68.33**	**6166.5**	**1.4244**
	CART	72.5	65	80	7250	-
	SVM+CART (*Case-P)*	65	55	70	6250	1.2836
6	SVM	65.83	63.34	68.34	6584	
	DT	66.67	60	81.66	7083	0.1085
	SVM+DT (*Case-P)*	64.16	63.33	65	6416.5	0.9451
	CART	71.97	63.33	80	7166.5	-
	SVM+CART (*Case-P)*	**66.15**	**55**	**73.33**	**6416.5**	**0.9898**
7	SVM	65.83	60	71.67	6583.5	
	DT	69.17	63.34	75	6917	0.3222
	SVM+DT (*Case-P)*	68.53	55	75	6500	0.7763
	CART	71.67	63.33	80	7166.5	-
	SVM+CART (*Case-P)*	**70.84**	**65**	**73.33**	**6916.5**	**0.3219**

The most important 7 selected features for UK bank data are *Current Assets-Stock/Current Liabilities, LAG(Number of days between account year end and the date of annual report), Funds Flow/Total Liabilities, Current Assets-Current Liabilities/Total Assets, Current Assets/Current Liabilities, Current Liabilities/Total Assets* and *Sales*. Using reduced feature set data it is observed that the hybrid SVM+CART using *Case-P* with 7 most important features obtained the accuracy of 70.84%, sensitivity of 65%, specificity of 73.33% and AUC of 6916.5. It is observed that less number of rules is extracted using reduced feature data without losing much accuracy of the system. Rules extracted using the reduced set of features is presented in Table 4.16.

Table 4.16: Rule set extracted using SVM+CART (*Case-P*)
for UK banks data (6 features)

Rule #	Antecedents	Consequent
1	if *LAG* <= 264.5	Non-Bankrupt
2	if *LAG* > 264.5 and *CA-S/CL* <= 1.01755	Bankrupt
3	if *LAG* > 264.5 and *CA-S/CL* > 1.01755	Non-Bankrupt

A rule set is considered to display a high level of *fidelity* if it can mimic the behaviour of the machine learning technique from which it was extracted i.e. SVM in our study. Average fidelity obtained is presented in Table 4.17 below. It is observed based on the fidelity that our proposed approaches behave more than 97% exactly like SVM for all the classification problems tested in this study. It is also observed that the proposed hybrid approach yielded high fidelity for classification problems with reduced features.

Table 4.17: Average Fidelity of SVM+DT and SVM+CART hybrids

Dataset (# features)	SVM+DT	SVM+CART
Iris (all)	99.26	98.5
Spanish (all)	97.9	98.95
Spanish (4)	89.53	96.67
Spanish (5)	97.92	98.31
Spanish (6)	97.71	98.31
Turkish (all)	96.92	98.62
Turkish (6)	98.26	98.91
Turkish (7)	97.59	100
Turkish (8)	97.22	98.57
UK (all)	97.5	95.81
UK (5)	96.95	96.72
UK (6)	97.4	97.17
UK (7)	97.22	97.96
US (all)	98.27	96.43
WBC (all)	95.5	93.09
WBC (4)	96.06	94.44
WBC (5)	95.7	94.14
WBC (6)	95.98	93.86
Wine (all)	98.15	99.75
Wine (7)	98.61	98.98
Wine (8)	98.45	99.22
Wine (9)	98.45	99.22

At this juncture, it is worthwhile to compare the results of the present study with that of the study presented in Chapter 3. In Chapter 3 we presented a decompositional rule extraction approach from SVM, where fuzzy rules are extracted and applied to bankruptcy prediction in banks. In the present study we employed SVM-RFE for feature selection and proposed a pedagogical rule extraction approach whereas the study in Chapter 3 does not employ any feature selection algorithm. It is observed that more number of fuzzy rules are extracted using full feature data compared to that of the rules extracted using reduced features data. The fidelity yielded by the proposed hybrid approach with reduced features in this study is higher than that of the fidelity obtained using all features (see Table 3.12 of Chapter 3).

4.4.2 Regression Problems

The datasets considered for regression analysis are *Auto mpg, Body fat, Boston housing, Forest fires* and *Pollution*. Dealing with regression problems CART, ANFIS and DENFIS are used for rule generation purpose. The accuracy of the rules on a test and validation dataset for prediction problems is measured in terms of the Root Mean Squared Error (RMSE). Subset of features is selected using SVM-RFE (Guyon et al. 2002) and the efficiency of the selected features is evaluated using the rules extracted. Rules are extracted using CART, ANFIS and DENFIS using *Case-P* dataset resulting in the hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS.

Table 4.18 shows the average RMSE obtained using *Auto MPG* dataset with all features and with reduced features as well. The most important 4 important feature selected are *Displacement, Weight, Model year* and *Origin.* It is observed that the hybrid SVR+CART, SVR+ANFIS and SVR+DENFIS using *Case-P* performed better than their corresponding stand alone techniques. Looking at the overall results it is observed that SVR+CART using *Case-P* with 3 features yielded best prediction accuracy with RMSE value of 0.0153 where as the SVR+CART using *Case-P* with all the features yielded RMSE of 0.0235. Sample rules extracted using 3 features of *Auto MPG* data are presented in Table 4.19 and total rules extracted are presented in Table D.2 in Appendix D.

Table 4.18: Average RMSE obtained using all and reduced features of *Auto MPG* dataset

# features	SVR	CART	SVR+CART *Case-P*	ANFIS	SVR+ANFIS *Case-P*	DENFIS	SVR+DENFIS *Case-P*
All	0.0109	0.0484	0.0235	0.1607	0.1207	0.3134	0.107
3	0.01226	0.04769	**0.0153**	0.1009	0.1109	0.0973	0.1112
4	0.01167	0.04767	0.01751	0.1138	0.108	0.253	0.1998

Table 4.19: Sample Rule set extracted using SVR+CART (*Case-P*) for *Auto MPG* dataset (3 features)

Rule #	Antecedents	Prediction
1	If Displacement <= 0.114987 and Model Layer <= 0.625 and Weight <= 0.12617	0.55459
2	If Displacement <= 0.114987 and Model Layer <= 0.625 and Weight > 0.12617	0.486769
3	If Displacement <= 0.114987 and Model Layer > 0.625 and Weight <= 0.147292	0.64483

Average RMSE obtained using *Body fat* data is presented in Table 4.20. The RMSE obtained using the hybrid SVR+DENFIS using *Case-P* with all the features is 0.0085. The 7 most important subsets of the features selected are *Density determined from underwater weighing, Weight (lbs), Height (inches), Hip circumference (cm), Knee circumference (cm), Neck circumference (cm), Abdomen 2 circumference (cm), Chest circumference (cm), Biceps (extended) circumference (cm)*. It is observed from the results obtained that the hybrid SVR+DENFIS using *Case-P* with 5 most important features outperforms other classifiers yielding RMSE 0.0045. It is observed that number of rules extracted using reduced feature data is less than that of using full feature data. Rules extracted from the hybrid SVR+DENFIS using *Case-P* with 5 features are presented in Table 4.21.

Table 4.20: Average RMSE obtained using all and reduced features of *Body fat* dataset

# features	SVR	CART	SVR+CART *Case-P*	ANFIS	SVR+ANFIS *Case-P*	DENFIS	SVR+DENFIS *Case-P*
All	0.00014	0.0134	0.0163	0.057	0.0137	0.0311	0.0085
5	0.000026	0.1288	0.01505	0.0146	0.0052	**0.0037**	**0.0045**
7	0.000032	0.01337	0.02535	0.0138	0.0058	0.0042	0.0051
9	0.000039	0.01342	0.01514	0.0221	0.0061	0.0961	0.0054

Table 4.21: Rule set extracted using SVR+DENFIS (*Case-P*)
for *Body Fat* dataset (5 features)

Rule #	**Antecedents**	**Prediction**
1	If DEN is GMF(0.50, 0.67) and W is GMF(0.50, 0.21) and H is GMF(0.50, 0.75) and HC is GMF(0.50, 0.21) and KC is GMF(0.50, 0.40)	Y = 2.04 -1.04*DEN +0.00*W -0.03*H +0.00*HC -0.03*KC
2	If DEN is GMF(0.50, 0.28) and W is GMF(0.50, 0.37) and H is GMF(0.50, 0.05) and HC is GMF(0.50, 0.49) and KC is GMF(0.50, 0.58)	
3	If DEN is GMF(0.50, 0.26) and W is GMF(0.50, 0.58) and H is GMF(0.50, 0.79) and HC is GMF(0.50, 0.63) and KC is GMF(0.50, 0.42)	
4	If DEN is GMF(0.50, 0.25) and W is GMF(0.50, 0.95) and H is GMF(0.50, 0.86) and HC is GMF(0.50, 0.95) and KC is GMF(0.50, 0.95)	

* GMF(*x*, *y*) indicates Gaussian Membership function with mean *x* and variance *y*.

Average results obtained using *Boston Housing* data are presented in Table 4.22. It is observed from the results that the hybrid SVR+CART using *Case-P*, considering all the features obtained the best RMSE of 0.0216. The 9 most important features selected are *ZN: proportion of residential land zoned for lots over 25,000 sq.ft., RM: average number of rooms per dwelling, DIS: weighted distances to five Boston employment centres, PTRATIO: pupil-teacher ratio by town, LSTAT: % lower status of the population, TAX: full-value property-tax rate per $10,000, B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town, RAD: index of accessibility to radial highways and NOX: nitric oxides concentration (parts per 10 million)*. Using 5 most important features the RMSE obtained by SVM+CART using *Case-P* dataset is 0.0352. Sample rules extracted using SVR+CART with *Case-P* dataset with all features are presented in Table 4.23 and the actual rules extracted are presented in Table D.3 in Appendix D.

Table 4.22: Average RMSE obtained using all and reduced features
of *Boston Housing* dataset

# feature	**SVR**	**CART**	**SVR+CART *Case-P***	**ANFIS**	**SVR+ANFIS *Case-P***	**DENFIS**	**SVR+DENFIS *Case-P***
All	0.0101	0.0656	**0.0216**	0.5636	0.1068	1.1477	0.0198
5	0.0116	0.0645	**0.0352**	0.0943	0.108	0.4619	0.1092
7	0.0103	0.0603	0.0573	0.0902	0.1024	0.1171	0.1034
9	0.0091	0.0511	0.0363	0.1347	0.0982	0.1443	0.1319

Table 4.23: Sample Rule set extracted using SVR+CART (*Case-P*) for *Boston Housing* dataset (all features)

Rule #	**Antecedents**	**Predictions**
1	If PTRATIO <= 0.75 and RM <= 0.55 and LSTAT <= 0.158527	0.460798
2	If PTRATIO <= 0.75 and RM <= 0.55 and LSTAT > 0.16 and LSTAT <= 0.27	0.407351
3	If LSTAT <= 0.27 and PTRATIO <= 0.75 and RM > 0.55 and RM <= 0.62	0.494458

Table 4.24 presents the prediction accuracies obtained using *Forest fires* data, with and without feature selection. It is observed from the obtained prediction accuracies that the hybrid using *Case-P* performs better than their corresponding stand-alone techniques. Using *Forest fires* dataset 8 most important feature subsets are extracted *viz., X - x-axis spatial coordinate within the Montesinho park map: 1 to 9, Month, DC - Drought Code, RH - relative humidity, wind - wind speed in km/h, rain - outside rain in mm/m2, day and ISI - Initial Spread Index*. SVR+CART hybrid using *Case-P* dataset outperform all other techniques tested with 7 most important features and obtained RMSE of 0.0005. Sample rules extracted using the hybrid SVR+CART using *Case-P* dataset with 7 most important features are presented in Table 4.25 and total rules are presented in Table D.4 in Appendix D.

Table 4.24: Average RMSE obtained using all and reduced features of *Forest Fires* dataset

# Feature	**SVR**	**CART**	**SVR+CART *Case-P***	**ANFIS**	**SVR+ANFIS *Case-P***	**DENFIS**	**SVR+DENFIS *Case-P***
All	0.0001	0.0418	0.00058	0.3462	0.0097	0.2513	0.0097
6	0.000096	0.0537	0.0006	0.0359	0.0098	0.0186	0.0098
7	0.000096	0.1028	**0.0005**	0.1785	0.0099	0.0197	0.0099
8	0.000183	0.0522	0.00067	0.3825	0.0105	0.0206	0.0099

Table 4.25: Sample Rule set extracted using SVR+CART (*Case-P*) for *Forest Fires* dataset (7 features)

Rule #	Antecedents	Prediction
1	If X-Axis <= 0.6875 and Month <= 0.681818 and Wind <= 0.077778	0.00328504
2	If Month <= 0.681818 and X-Axis <= 0.1875 and RH <= 0.705882 and Wind > 0.077778 and Wind <= 0.677778 and DC <= 0.111528	0.00267747
3	If Month <= 0.681818 and X-Axis <= 0.1875 and Wind > 0.077778 and Wind <= 0.677778 and DC > 0.111528 and DC <= 0.755365 and RH <= 0.329412 and Day <= 0.0833335	0.00300294
4	If Month <= 0.681818 and X-Axis <= 0.1875 and Wind > 0.078 and Wind <= 0.68 and RH <= 0.33 and Day > 0.08 and DC > 0.11 and DC <= 0.69	0.00287374

Average RMSEs obtained using *pollution* dataset is presented in Table 4.26. The selected 9 most important feature are *JANT Average January temperature in degrees F, JULT Average July temperature in degrees F, EDUC Median school years completed by those over 22, NONW % non-white population in urbanized areas, 1960, POOR % of families with income<$3000, HC Relative hydrocarbon pollution potential, NOX Same for nitric oxides, PREC Average annual precipitation in inches, POPN Average household size, HOUS % of housing units which are sound and with all facilities and DENS Population per sq. mile in urbanized areas, 1960* in order of their importance. It is observed that the hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS using *Case-P* performed better than their corresponding standalone CART, ANFIS and DENFIS using all features and reduced features as well. Best prediction accuracy is obtained by SVR+CART using *Case-P* with all the features i.e. RMSE of 0.039. Rules extracted using the hybrid SVR+CART with all the features are presented in Table 4.27.

Table 4.26: Average RMSE obtained using all and reduced features of *Pollution* dataset

# features	SVR	CART	SVR+CART *Case-P*	ANFIS	SVR+ANFIS *Case-P*	DENFIS	SVR+DENFIS *Case-P*
All	0.006	0.1127	**0.0399**	0.1034	0.0946	0.1395	0.077
7	0.00931	0.1198	0.0492	0.3444	0.0929	0.0878	0.0962
9	0.00138	0.1022	0.0769	0.1689	0.1193	0.0987	0.0906
11	0.0021	0.1013	0.0911	0.1144	0.0984	0.0796	0.0899

Table 4.27: Rule set extracted using SVR+CART (*Case-P*)
for *Pollution* dataset (all features)

Rule #	Antecedents	Prediction
1	If NONW <= 0.401857 and EDUC <= 0.893939 and HOUS <= 0.468619	0.611489
2	If NONW <= 0.401857 and EDUC <= 0.893939 and HOUS > 0.468619	0.464645
3	If NONW <= 0.401857 and EDUC > 0.893939	0.309484
4	If NONW > 0.401857	0.655993

4.4.3 Overall observations

For classification problems, it is observed that using *iris* and *wine* data the hybrids SVM+CART using *Case-P* dataset yielded better accuracy compared to the hybrid SVM+DT. For *Spanish banks* data, the hybrid SVM+CART using *Case-P* dataset performs the best. Using *UK banks* data and *WBC* data, it is observed that the hybrid SVM+CART using *Case-P* performs best when all the features are considered. It is observed that reduced feature data helps in extracting smaller rules and less number of rules without compromising the prediction accuracy of the system. It is observed from the results obtained for regression problems that the hybrids SVM+CART, SVM+ANFIS and SVSM+DENFIS using *Case-P* outperform their corresponding stand-alone CART, ANFIS and DENFIS using all features and reduced features as well.

4.5 Conclusions

In this chapter a pedagogical rule extraction method is presented. Feature selection using SVM-RFE is employed and reduced feature data is then used for rule generation purpose. DT and CART are employed for extracting rules to solve classification problems whereas CART, ANFIS and DENFIS are employed to extract rules to solve regression problems. For classification problems, mixed results are obtained where without much compromise to accuracy comprehensibility of the system is achieved. For benchmark problem like *iris*, *wine* it is observed that the hybrid rule extraction performed equally well with SVM. For *Spanish bank* dataset it is the hybrid SVM+CART using *Case-P* dataset performed the

best. When all the features are considered it is the hybrid SVM+CART using *Case-P* performed best for UK banks data and WBC data. AUC of the classifiers are considered to calculate t-test values, it is observed based on t-test values that our proposed hybrids are statistically insignificant compared with the best classifier. It is concluded that the proposed approach mimic the behaviour of SVM model more than 97% correctly for all classification problems analyzed in the present study. It is concluded from regression results that the proposed hybrids SVM+CART, SVM+ANFIS and SVM+DENFIS using *Case-P* obtained the best prediction accuracy compared to their corresponding stand-alone CART, ANFIS and DENFIS using all features and with reduced feature set as well.

Using the predictions of SVM/SVR for rule generation we ensure that these rules actually mimic the behaviour of trained SVM/SVR models. The following conclusions are made from the current work.

- Using SVM-RFE for feature selection improves the prediction accuracy for classification and regression problems.
- Less number of rules is generated using the reduced set of features without compromising the accuracy of the model.
- The antecedents per rule also become less while improving the accuracy.
- The proposed hybrid approach always performs better than the corresponding stand-alone approaches tested, specifically for regression analysis.

Part III

Eclectic/ Hybrid Approach

Chapter 5

Rule Extraction from SVM for Data Mining on Unbalanced Datasets

This Chapter presents an *eclectic rule extraction technique* which analyzes medium scale unbalanced dataset pertaining to finance. At the outset the motivation for the proposed approach is presented. Introduction to customer relationship management (CRM) is then presented and literature of churn prediction problem is reviewed in the next section. Subsequent section presents the proposed approach in detail. The proposed approach can also be considered as an application of analytical CRM. Fourth section provides the details about the literature to deal with unbalanced datasets. Dataset description and Results and discussions is presented in the following sections. Final section concludes the chapter.

5.1 Motivation

It is observed that rule extraction from SVM has been successfully applied for small scale and balanced problems. On the other hand SVM has shown superior performance dealing with large and unbalanced datasets. No rule extraction approach is reported in literature for analyzing medium scale and unbalanced datasets. Further, NBTree (Naive Bayes Tree) algorithm was never employed to generate rules from SVM. In many finance applications it is observed that all or more than 90% of the data belong to one class and very few instances are available for the other class usually the most important class and objective of the study. It is observed that the standard intelligent algorithms are biased towards majority class and ignore minority class data. This chapter presents an eclectic rule extraction technique which analyzes medium scale and highly unbalanced dataset i.e. *churn prediction in bank credit card customers*. The objective of the study is to discover about-to-churn bank credit card customers. It is of high importance to know in advance about such customers and take proper steps to retain them. Hence, rule extraction for solving

churn prediction problems provides better understanding about the customer needs and behaviour. Rules extracted using SVM for churn prediction problem can also be used as an early warning system that alerts the management about "*about-to-churn*" customers' behaviour. This is a very important application of analytical Customer Relationship Management (CRM) in finance.

5.2 Customer Relationship Management (CRM)

CRM is a process or methodology used to learn more about customers' need and behaviours in order to develop stronger relationship with them. CRM involves the continuous use of refined information about current and potential customers in order to anticipate and respond to their needs and draws on a combination of business process and Information Technology to discover the knowledge about the customers and answer questions like, "who are the customers?", "what do they do?" and "what do they like?". Therefore the effective management of information and knowledge is central and critical to the concept of CRM for;

- Product tailoring and service innovation (web-sites tailored to customer needs, taste experience and the development of mass customisation)
- Providing a single and consolidated view of the customer
- Calculating the lifetime value of the customer
- Designing and developing personalized transactions
- Multichannel based communication with the customer
- Cross-selling/up-selling various products to customers

Various definitions of CRM put emphasis on different perspectives. CRM's technological perspective was stressed in (Yuan and Chang, 2001; Peppers and Rogers, 1995; Shaw et al., 2001; Verhof and Donkers, 2001), its knowledge management perspective was emphasized in (Massey, Montoya-weins and Holcom, 2001) and its business re-engineering and continuous improvement perspective was presented in (Anton, 1996).

We can think about CRM at three levels, Strategic, Analytical and Collaborative.

Strategic CRM: It is focused on development of a customer-centric business culture. Product, production and selling are the three major business orientations identified by Kotler (Kotler, 2000).

Analytical CRM: Analytical CRM builds on the foundation of customer information. Customers' data may be found in enterprise wide repositories, sales data (purchasing history), financial data (payment history and credit score), marketing data (campaign response, loyalty scheme data) and service data. With the application of Data Mining, the bank/service organisation can then analyze this data and intelligent analysis provides answers to questions, such as, "who are our most valuable customers?", "which customer have the highest propensity to switch to competitors?", "which customers would be most likely to respond to particular offer?" and so on.

Collaborative CRM: Staff members from different departments can share information collected when interacting with customers. Collaborative CRM's ultimate goal is to use information collected by all departments to improve the quality of services provided by the company (Edward 2007).

Churn prediction problem is an analytical CRM application and using rules extracted from SVM service providers can get transparent and efficient insight about their customers and can make better policies to retain existing customers.

5.3 Churn Prediction Problem

Over the decade and half, the number of customers with banks and financial companies is increasing by the day and this has made the banks conscious of the quality of the services they offer. The phenomenon, called '*churn*' i.e. shifting loyalties from one service provider to another occurs due to reasons such as availability of latest technology, customer-friendly bank staff, low interest rates, proximity of geographical location, varied services offered, etc. Hence, there is a pressing need to develop models that can predict which existing '*loyal*' customer is going to churn out or attrite in near future.

Service organisations need to be proactive in understanding the customers' current satisfaction levels before they attrite (Bolton, 1998). Research indicates that the online bank customers are less price-conscious than traditional bank customers with less probability of churning out (Mols, 1998). Targeting customers on the basis of their (changing) purchase behaviour could help the organisations do better business and loyalty reward programmes helps the organizations build stronger relationships with customers (Bolton et al. 2000).

In the financial services industry two "critical" churn periods are identified (Larivière, and Van den Poel, 2004), the first period is the early years after becoming a customer and the second period is after being a customer for some 20 years. A comparative study on Logistic Regression and Neural Network for subscriber data of a major wireless carrier is carried out (Mozer et al., 2000) and it is concluded that using sophisticated neural net $93 could be saved per subscriber.

Machine learning techniques such as Multilayer Perceptron, Hopfield Neural network, Self Organising Map (Smith and Gupta, 2000), Decision Tree (Richeldi and Perrucci 2002; Euler, 2005; Chu, Tsai and Ho, 2007; Wezel and Potharst, 2007; Yuan and Chang, 2001; Wang et al., 2009), Multivariate Regression Analysis (Bloemer, Ruyter and Peeters, 1998), Multilayer perceptron, C4.5 decision trees, hierarchical neuro-fuzzy systems and a data mining tool named rule evolver based on genetic algorithms (GAs) (Ferreira et al., 2004), logistic regression and random forest (Buckinx, and Van den Poel, 2005; Mutanen, 2006; Neslin et al., 2006; Hung, Yen, and Wang, 2006; Xie et al., 2009), emergent self-organising feature maps (ESOM) (Ultsch, 2002), Neural Networks (Mozer et al. 2000), SVM (Coussement and Van den Poel 2008; Zhao and Xing-hua 2008; Huang et al. 2010), one-class SVM (Zhao et al. 2005), SVM-RFE (Cao and Shao 2008), hybridization of neural network with genetic algorithms (Hadden et al., 2007; Pendharkar, 2009), ensemble with majority voting (Kumar and Ravi 2008), Hybrid Neural Networks (Tsai, and Lu, 2009) and Generalized Additive Models (Coussement et al., 2009) are employed to solve churn prediction problems. Glady, Baesens, and Croux (2009) proposed a churn prediction model using customer life time value (CLV), which is defined as the discounted value of

future marginal earning, based on customers' activity. Hu (2005) presented a comparative study of different machine learning algorithms. According to (Lemon, While, and Winer, 2002), "the trend in marketing towards building relationships with customers continues to grow and marketers have become increasingly interested in retaining customers over the long run". Hyung-Su and Young-Gul (2009) suggested a performance measurement framework called CRM score card to diagnose and asses a firm's CRM practice.

Churn prediction problem is one of the most important applications of analytical CRM in finance. Banks would be interested to know their *about-to-churn* customers and the proposed rule extraction approach do not only provide better predictions but also comprehensibility of the system is improved. Feature selection using SVM-RFE algorithm in the first phase reduces the dimensionality of the data by yielding the key features in the data. Thus, less number of rules and smaller rules is extracted resulting in the improvement of the comprehensibility of the system to the user. During the research study in this chapter, various balancing techniques are employed such as, random undersampling, random oversampling, SMOTE to balance the dataset.

5.4 Proposed Eclectic Rule Extraction Approach

Churn prediction in bank credit card customers' problem is solved using the proposed approach. The churn prediction dataset is highly unbalanced with 93:7 class distributions where 93% of the samples are available for loyal customers and only 7% of the data is available to learn about churn customers. The churn prediction dataset is obtained from Chile in 2004, information about the dataset features is presented in Table C.10 in Appendix C. Balancing techniques such as, SMOTE, random undersampling, random oversampling and combined undersampling and oversampling are employed. The efficiency of SVM for feature selection and rule extraction from SVM using unbalanced and balanced data is analyzed. Extracting support vectors and feature selection using SVM in tandem results in vertically and horizontally reduced data. This newly generated data is then used for rule generation.

The proposed hybrid approach is composed of three phases and is depicted in Figure 5.1.

1. *Feature selection using SVM-RFE.*
2. *Support vector extraction using SVM.*
3. *Rule generation using NBTree.*

5.4.1 Feature Selection using SVM-RFE

SVM-RFE (Recursive Feature Elimination) (Guyon, 2002) algorithm is employed for feature selection purpose. Nested subsets of features are selected in a sequential backward elimination manner, which starts with all the features variables and removes one feature variable at a time. At each step, the coefficients of the weight vector *w* of a linear SVM are used to compute the feature ranking score. The feature say, the i^{th} feature with the smallest ranking score $c_1 = (w)^2$ is eliminated, where *w* represents the corresponding component in the weight vector w. Using $c = (w)^2$ as the ranking criterion corresponds to removing the feature whose removal changes the objective function the least. This objective function is chosen to be $J = ½ ||w||^2$ in SVM-RFE.

5.4.2 Support Vector Extraction using SVM

Dealing with churn prediction data, sensitivity of the classifier is considered the most important factor for developing the SVM model and for extracting support vectors. Later, the corresponding actual target values of support vectors are replaced by the predicted target values of SVM, resulting in *Case-SP* dataset whereas support vectors with corresponding actual target values is called *Case-SA* dataset. For comparative study, the corresponding actual target values of training instances are also replaced by the predictions of SVM model, resulting in *Case-P* dataset. By using the newly generated *Case-P* and *Case-SP* we ensure that the rules extracted actually represent the knowledge learnt by the SVM.

5.4.3 Rule Generation using NBTree

NBtree (Kohavi, 1996) is employed for rule generation purpose. It attempts to utilize the advantages of both decision trees (i.e. segmentation) and naïve bayes (evidence accumulation from multiple features). A decision tree is built with univariate splits at each node, but with Navie-Bayes classifiers at the leaves instead of the predictions for single

class. It is concluded that NBTree's induction process is useful for larger datasets (Kohavi, 1996). Rules are generated under 10-fold cross validation method of testing and the generated rules are later tested against the validation set.

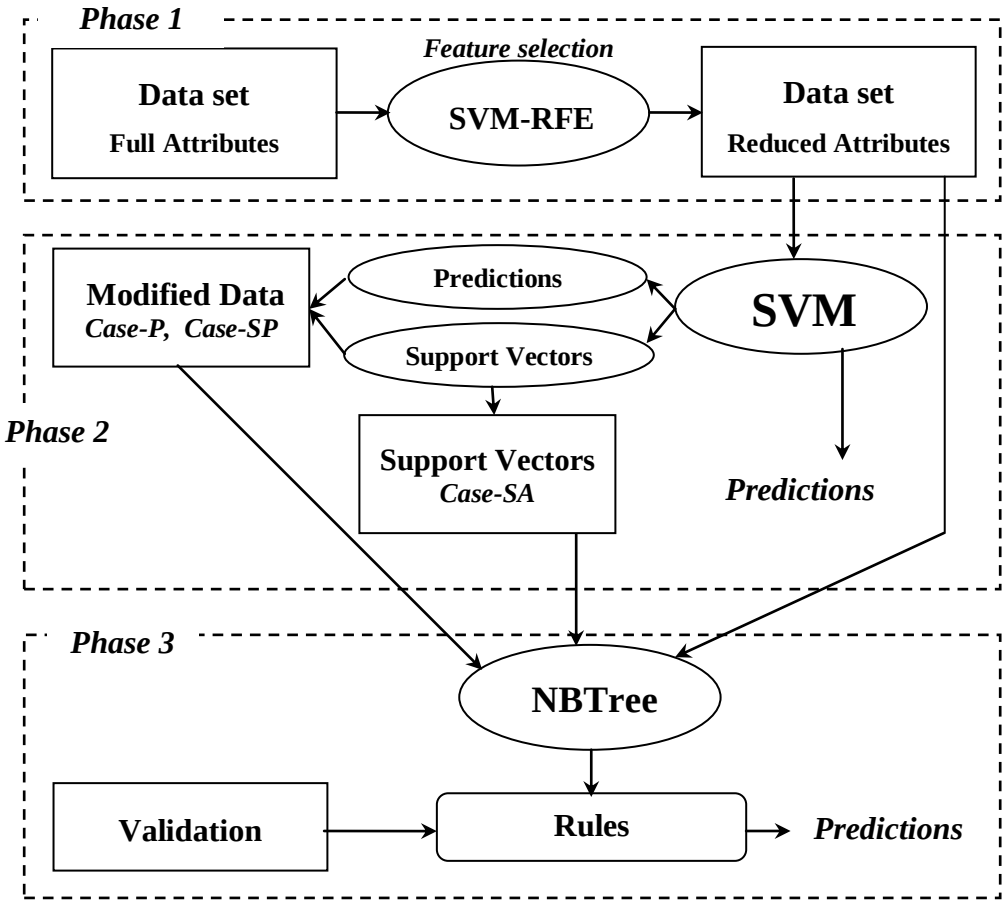


Figure 5.1: Rule extraction using selected features of data

Note. ***Case - P***: training set with corresponding Predicted target values.
Case - SA: Support vectors with corresponding Actual target values.
Case - SP: Support vectors with corresponding Predicted target values.

5.5 Dataset Description

The churn prediction dataset is obtained from a Latin American Bank that suffered from an increasing number of churns with respect to their credit card customers and decided to improve its retention system. Two groups of variables are available for each customer: sociodemographic and behavioural data, which are described in Table C.10 in Appendix C.

The dataset comprises of 22 variables, with 21 predictor variables and 1 class variable. It consists of 14814 instances, of which 13812 instances are pertaining to loyal customers and 1002 instances represent churned customers. Thus, there are 93.24% loyal customers and 6.76% churned customers. Hence, the dataset is highly unbalanced in terms of the proportion of churners versus nonchurners (Business Intelligence Cup 2004).

5.6 Data Imbalance Problem

In many real time problems, almost all the instances belong to one class, while far fewer instances are labelled as the other class, usually the more important class. It is obvious that traditional classifier seeking an accurate performance over a full range of instances are not suitable to deal with imbalanced learning task, since they tend to classify all the data into majority class, which is usually not the objective of the study and less important. Research studies (Weiss, 1995; Fawcett and Provost, 1997; Jackowicz and Stephens, 2002; Kubat et al., 2004; Visa and Ralescu, 2005) show that many standard machine learning approaches result in poor performance, specifically dealing with large unbalanced datasets.

Class imbalance problem exists in many application domains, such as telecommunications (Hilas, 2009), detection of oil spoils in satellite radar images (Kubat et al., 2004), learning word pronunciation (Davel and Bernard, 2004), text classification (Sebastiani, 2002), risk management (Galindo and Tamayo, 2000), information retrieval (Chen, 1995), medical diagnosis (Kononenko, 2001), intrusion detection (Lee et al., 1999) and fraud detection (Sanchez et al., 2009).

5.6.1 Literature Review of Techniques Dealing with Unbalanced Data

Since late 1960's, researchers put their efforts towards developing strategies to deal with class imbalance problems and proposed various methodologies towards dealing with such imbalance problems. Methods to deal with imbalanced problems include, resizing training set, adjusting misclassification costs and recognition based learning. Resizing training set is a simple strategy that includes, oversampling minority class samples (Ling and Li, 1998) and downsizing majority class samples (Kubat and Matwin, 1997). Cost sensitive classifiers (Domingos, 1999) have been developed to handle the problem with different

misclassification error costs, but may also be used for unbalanced dataset. Recognition based learning approach learn rules from the minority class examples with or without using the examples of minority class (Kubat et al., 2004).

In the earliest stages of this research, undersampling using condensed nearest neighbour (CNN) (Hart, 1968), Edited Nearest Neighbor (ENN) (Wilson 1972), Selective undersampling using Tomak-Links concept (Kubat and Matwin 1997), ENN with Neighbourhood cleaning rule (Laurikkala 2001) are proposed. Further, Chawla et al., (2002) proposed SMOTE (Synthetic Minority Oversampling TEchnique), where synthetic (artificial) samples are generated rather than oversampling by replacement.

Maloof (2003) reported that sampling has the same results as moving the decision threshold or adjusting the cost matrix. Estabrooks et al., (2004) conducted a study to evaluate the effectiveness of oversampling and undersampling. They concluded that combining different expressions of re-sampling approach is an effective solution. Detailed review reports were presented (Provost, 2000; Monard and Batista, 2002; Weiss 2004; Kotsiantis et al., 2006; Kumar and Ravi 2008; Guo et al., 2009), discussing about the issues related to the problem solving using machine learning techniques when provided with unbalanced training data.

Combination of undersampling and oversampling is then proposed by Ling and Li, (1998). They used lift analysis instead of classification accuracy to measure a classifiers performance. They found that the combination of oversampling and undersampling does not provide any significant improvement in the life index. Weiss and Provost, (2001) suggested that a progressive, adaptive sampling strategy be developed that incrementally requested new samples based on the improvement in the classifiers performance. They employed C4.5 algorithm and considered error rate and AUC of the algorithm to generate new samples.

Weiss and provost (2003) proposed a heuristic, budget sensitive, progressive sampling algorithm for selecting training data that approximates optimum. They argued that, though the heuristically determined class distribution associated with the final training set is not

guaranteed to yield the best performing classifier. The classifier indeed using this class distribution performs well in practice.

Researchers have emphasised the use of clustering based preprocessing methods as an alternative for sampling of the data. Batista et al., (2004b) proposed two hybrid sampling techniques named, SMOTE+TOMEK Links and SMOTE+ENN for overlapping datasets, for better defined class clusters among majority and minority classes. Jo and Japkowicz, (2004) presented a cluster based oversampling approach. Later, Batista et al., (2004a, 2004b) proposed a hybrid sampling approach combining CNN rule with Tomek-links.

Application of boosting to deal with unbalanced problems is then proposed by Guo and Viktor, (2004). They proposed boosting method with various oversampling techniques to deal with hard to classify examples and concluded that boosting approach improves the prediction accuracy of the classifier. Huang et al., (2004) presented Biased Minimax Probability Machine to resolve the imbalance problem.

Han et al., (2005) proposed borderline SMOTE, which identifies minority samples at borderline and apply SMOTE. This is the only technique proposed to over-sample the borderline minority samples. Cohen et al., (2006) proposed k-means based undersampling method and Agglomerative Hirarchical Clustering (AHC) based oversampling method to deal with unbalanced datasets. Later, Liu et al., (2006) proposed SMOTE-Bootstrap hybrid (SMOTE-BU) method to deal with unbalanced data. Where, SMOTE is applied to over-sample the minority instances and Bootstrap is applied to under-sample the majority instances.

In recent past, Guo et al., (2009) proposed different approaches based on four levels according to the phases in the learning. Changing the class distribution mainly by re-sampling, feature selection, manipulating classifier internally at classification level and ensemble learning for the final classification.

5.6.2 Random Undersampling

Undersampling is a technique in which some of the samples belonging to the majority class are removed randomly and combined with the minority class samples. For example,

25% undersampling means that the majority class is reduced to 25% of its original size in other words, 25% of the available majority class instances are removed randomly from data. 50% undersampling means that the majority class is reduced to 50% of its original size.

5.6.3 Random Oversampling

Oversampling is a technique in which the samples belonging to the minority class are replicated a few times and combined with the majority class samples. For example, 100% oversampling means that the minority class instances are replicated once in other words, minority class instances are doubled, and 200% oversampling means that the minority class is replicated twice.

5.6.4 Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is an approach in which the minority class is oversampled by creating synthetic (or artificial) samples, rather than by oversampling with replacement. The minority class is oversampled by taking out each sample and introducing synthetic samples along the line segments that join any/all of the *k* minority class nearest neighbours. SMOTE is used to widen the data region that correponds to minority samples. This approach effectively forces the decision region of the minority class to become more general (Chawla *et al*., 2004).

5.7 Results and Discussions

Many business decision makers place high emphasis on sensitivity alone because higher sensitivity leads to greater success in correctly identifying potential churners and thereby contributing to the bottom-line of the fundamental CRM *viz.*, retaining extant loyal customers. Consequently in this chapter, sensitivity is accorded top priority ahead of specificity and accuracy. Therefore, we evaluate and discuss the performance of our proposed hybrid approach SVM+NBTree using *Case-SP* with respect to sensitivity alone. SVM-RFE algorithm is employed for feature selection and six features are selected those are, *CRED_T (Credit in month T), CRED_T-1 (Credit in month T-1), CRED_T-2 (Credit in month T2), NCC_T (Number of Credit Cards in month T), NCC_T-2 (Number of Credit*

Cards in month T2) and *T_WEB_T (Web Transaction in month T)*. Balancing methods of random undersampling, random oversampling, combination of undersampling and oversampling and SMOTE are employed and an extensive study is carried out. Empirical results obtained by the proposed hybrid using the dataset including all features and with reduced features are presented in Table 5.1 through 5.11.

Table 5.1 presents the results obtained using original unbalanced data. When all the features in data are considered, it is observed that the hybrid SVM+NBTree using *Case-SP* dataset yielded the best sensitivity of 75.45%. Using the unbalanced data with reduced features the proposed hybrid SVM+NBTree using *Case-SP* dataset obtained the sensitivity of 82.45% whereas SVM+NBTree using *Case-P* dataset yielded the highest sensitivity of 84.7%. Thus, feature selection has indeed helped in improving the sensitivity of the hybrid approaches. It is observed that proposed hybrid using reduced feature data yielded better sensitivity compared to the corresponding hybrid using full features of the data. Hybrid SVM+NBTree using *Case-P* dataset yielded best sensitivity but the number of rules extracted is above hundred, thus the comprehensibility of the system is very poor, hence it is not advisable to use. Whereas the proposed approach SVM+NBTree using *Case-SP* dataset yielded marginally less sensitivity compared to SVM+NBTree using *Case-P* dataset and the number of rules extracted is approximately twenty only resulting in improved comprehensibility of the system.

Table 5.1: Average results obtained using original unbalanced data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	64.65	80.63	79.55	82.85	72.25	74.79
NBTree	62.7	99.07	96.62	52.7	99.33	96.17
SVM+NBTree (*Case-P*)	67.65	84.9	83.74	**84.7**	74.59	75.25
SVM+NBTree (*Case-SA*)	0	100	93.25	0	100	93.25
SVM+NBTree (*Case-SP*)	75.45	79.03	78.79	**82.45**	75.89	76.34

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Table 5.2 presents the results obtained using 25% undersampling data. It is observed that the hybrid SVM+NBTree using *Case-SP* dataset yielded best sensitivity of 79.25% among all the cases tested considering all the feature of the data. When the reduced feature of the

dataset is considered the proposed approach SVM+NBTree using *Case-SP* dataset yielded 81.2% sensitivity, which is slightly less than the sensitivity obtained by the hybrid SVM+NBTree using *Case-P* dataset i.e. 82.2%. Yet again, feature selection improved the sensitivity of all the hybrid methods compared to other corresponding hybrids using full features. It is also observed that using proposed approach the number of rules extracted are less.

Table 5.2: Average results obtained using 25% under-sampled data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	71.75	78.42	77.92	82.5	75.58	76.06
NBTree	64.75	98.71	96.41	59.4	98.08	95.48
SVM+NBTree (*Case-P*)	78.25	78.64	78.61	**82.2**	76.16	76.57
SVM+NBTree (*Case-SA*)	0	100	93.25	0	100	93.25
SVM+NBTree (*Case-SP*)	79.25	76.71	76.87	**81.2**	70.98	71.67

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Table 5.3 presents the results obtained using 50% undersampling data. It is observed that considering all the features of the data, the hybrid SVM+NBTree using *Case-SP* set yielded the best sensitivity of 87.25%. Whereas the proposed hybrid SVM+NBTree using *Case-SP* set with reduced feature set obtained the sensitivity of 78.95%. It is observed that 50% undersampling degrades the performance of the hybrids with reduced features and the proposed approach SVM+NBtree using *Case-SP* dataset stands best in the list.

Table 5.3: Average results obtained using 50% under-sampled data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	78.35	72.84	73.21	77.5	91.78	90.85
NBTree	69.4	98.21	96.26	70.5	96.33	94.59
SVM+NBTree (*Case-P*)	81	75.12	75.45	77.65	91.9	90.42
SVM+NBTree (*Case-SA*)	0.05	99.99	93.24	0	100	93.25
SVM+NBTree (*Case-SP*)	**87.25**	69.83	70.99	78.95	91.19	90.37

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

When undersampling is employed it is observed that using 25% undersampling with reduced feature data yielded better sensitivity where as using 50% undersampling the

hybrid using full features data performed the best. Using 50% undersampling the class distribution ratio become 87:14, where 87% represent loyal customers' instances and 14% of the data represent churned customers.

Results obtained using 100% oversampling data are presented in Table 5.4. It is observed that the sensitivity obtained by the hybrid SVM+NBTree using *Case-SP* set with all the features and with reduced features yielded sensitivities of 81.95% and 82.6% respectively. The hybrid SVM+NBTree using *Case-P* set with reduced features obtained slightly higher sensitivity with 82.8%. It is observed that using 100% oversampling also the proposed approach SVM+NBTree using *Case-SP* dataset yielded the best sensitivity and it also extracted less number of rules. Whereas the hybrid SVM+NBTree using *Case-P* dataset with reduced features yielded more number of rules resulting in poor comprehensibility, despite yielding similar results with SVM+NBTree using *Case-SP* dataset with reduced features.

Table 5.4: Average results obtained using 100% over-sampled data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	74.7	77.21	77.03	81.95	86.12	85.79
NBTree	64.9	98.11	95.87	71.45	96.51	94.82
SVM+NBTree (*Case-P*)	78.65	78.36	78.34	**82.8**	86.33	86.09
SVM+NBTree (*Case-SA*)	0.4	99.14	92.47	0	100	93.25
SVM+NBTree (*Case-SP*)	81.95	73.86	74.41	**82.6**	83.35	83.3

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Table 5.5 presents the results obtained using 200% oversampling data. The sensitivity yielded by the hybrid SVM+NBTree using *Case-SP* dataset with all the features is 82.5%. With reduced features the proposed hybrid SVM+NBTree using *Case-SP* dataset yielded the best sensitivity of 88.35%. Once again feature selection improved the sensitivity of the hybrid. It is observed that feature selection helps the hybrid SVM+NBTree using *Case-SP* dataset to learn better about churners and yielded better sensitivity.

Table 5.5: Average results obtained using 200% over-sampled data

Model	Full features			Reduced Features		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	77.6	75.83	75.94	86.5	77.54	78.15
NBTree	66.1	97.28	95.18	72.9	96.06	94.5
SVM+NBTree (*Case-P*)	79.15	76.86	77.01	86.05	77.92	78.47
SVM+NBTree (*Case-SA*)	6.65	99.7	93.42	0	100	93.25
SVM+NBTree (*Case-SP*)	82.5	75.77	76.23	**88.35**	72.72	73.6

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Table 5.6 shows the results obtained using 300% oversampling data. It is observed that with all the feature of the data the hybrid SVM+NBTree using *Case-SP* set yielded the sensitivity of 78.3%, whereas the sensitivity obtained with reduced features is 85.3%. The sensitivity yielded by SVM+NBTree using *Case-P* set is 85.9% but the number of rules extracted are high in number. Yet again sensitivity of the hybrid is improved after feature selection is performed.

Among various experiments conducted using different oversampling percentages, it is observed that the hybrids using 200% over-sampled data yielded the best sensitivity. Using 200% oversampling the distribution of the classes in the dataset become 82%:18% where 82% of the instances are available for loyal customers and 18% instances are available for churned customers.

Table 5.6: Average results obtained using 300% over-sampled data

Model	Full features			Reduced Features		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	64.1	91.21	89.29	86.1	72.35	73.09
NBTree	63.58	97.29	94.98	73.4	95.41	93.92
SVM+NBTree (*Case-P*)	68.75	93.85	91.14	**85.9**	74.11	75.17
SVM+NBTree (*Case-SA*)	0	100	93.25	0	100	93.25
SVM+NBTree (*Case-SP*)	78.3	84.89	84.45	**85.3**	68.14	69.36

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Tables 5.7 and 5.8 show the results obtained using the dataset which is balanced using the combination of undersampling and oversampling. Table 5.7 presents the results obtained using the combination of 25% undersampling and 100% oversampling. It is observed that

the hybrid SVM+NBTree using *Case-P* dataset yielded the best sensitivity of 71.07% with all the features. With reduced features, it is observed that the proposed hybrid SVM+NBTree using *Case-SP* dataset obtained an improved and best sensitivity of 86.95% and reduced rule set as well.

Table 5.7: Average results obtained using 25% under + 100% over sampled data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	68.41	77.17	77.58	85.95	78.99	79.5
NBTree	68.6	97.5	95.55	71.65	96.17	94.52
SVM+NBTree (*Case-P*)	71.07	79.68	80.92	85.75	80.05	80.44
SVM+NBTree (*Case-SA*)	0	100	93.25	0	100	93.25
SVM+NBTree (*Case-SP*)	69.5	75.76	75.22	**86.95**	76.59	77.3

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Results obtained using the combination of 50% undersampling and 200% oversampling are presented in Table 5.8. It is observed that the hybrid SVM+NBTree using *Case-SP* dataset yielded the best sensitivity of 78.25% using all the features. It is observed that the proposed hybrid SVM+NBTree using *Case-SP* dataset using reduced features yielded an improved sensitivity of 86%, which is almost equal to the sensitivity obtained using the hybrid SVM+NBTree using *Case-P* set i.e. 86.1%. As the number of rules extracted using *Case-SP* dataset is less than that of the rules extracted using *Case-P* dataset, it is advisable to use the hybrid SVM+NBTree using *Case-SP* dataset. Once again feature selection improves the sensitivity of the rules extracted.

Table 5.8: Average results obtained using 50% under + 200% over sampled data

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	69	78.16	77.49	85.97	74.72	76.1
NBTree	72.2	94.17	92.62	76.5	93.53	92.38
SVM+NBTree (*Case-P*)	74.22	82.75	80.62	**86.1**	76.04	76.66
SVM+NBTree (*Case-SA*)	71.85	82.77	82.06	0	100	93.25
SVM+NBTree (*Case-SP*)	78.25	71.53	72.04	**86**	72.61	73.43

Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

Results obtained using SMOTE are presented in Table 5.9. It is observed that using all features the hybrid SVM+NBTree using *Case-SP* dataset obtained the best sensitivity of 79.7%. The proposed hybrid SVM+NBTree using *Case-SP* dataset with reduced features yielded the best sensitivity of 91.85%. Thus, once again feature selection improved the sensitivity of all the hybrids. It is observed that balancing using SMOTE improves the performance of the proposed approach. It is also observed that less number of rules is extracted using proposed hybrid SVM+NBTree using *Case-SP* dataset and the rules extracted using SMOTE data are presented in Table 5.10.

Table 5.9: Average results obtained using SMOTE

Model	**Full features**			**Reduced Features**		
	Sens*	**Spec***	**Acc***	**Sens***	**Spec***	**Acc***
SVM	72.4	87.16	86.17	91.05	70.03	71.45
NBTree	63.95	96.55	94.35	75.35	93.92	92.67
SVM+NBTree (*Case-P*)	74.15	88.8	87.83	91.3	71.23	72.38
SVM+NBTree (*Case-SA*)	61.05	59.01	59.21	49.2	56.77	56.26
SVM+NBTree (*Case-SP*)	79.7	83.71	83.11	**91.85**	67.12	68.67

` Sens* = Sensitivity; Spec* = Specificity; Acc* = Accuracy;

A rule set is considered to display a high level of *fidelity* if it can *mimic* the behaviour of the machine learning technique from which it was extracted i.e. SVM in our study. Apart from accuracy, sensitivity and specificity, fidelity also is an important quantity to measure the quality of the rules. Fidelity yielded by various classifiers tested during this study are presented in Table 5.11. It is observed that, using SMOTE the hybrid SVM+NBTree using *Case-SP* dataset with all the features yielded that best fidelity of 93.46%. In other words, in this case the hybrid SVM+NBTree behaves 93.46% times same as the SVM. With reduced features the hybrid SVM+NBTree using *Case-SP* dataset using 50% undersampling data yielded the best fidelity of 97.63%. The proposed approach SVM+NBTree using *Case-SP* with reduced features using 50% undersampling data behaves 97.63% exactly as SVM from which rules are extracted. It is observed that the rules extracted using reduced feature data behave much similar to SVM compared to the case when they are extracted using full feature data.

Table 5.10: Rule set extracted using SMOTE data with reduced features

Rule #	Antecedents	Consequent
1	If CRED_T-2 <= 98.379 and CRED_T-1 <= 99.626 and NCC_T <= 1.529 and CRED_T <= 607.095 and T_WEB_T <= 13.628	Churner
2	If CRED_T-2 <= 98.379 and CRED_T-1 <= 99.626 and NCC_T <= 1.529 and CRED_T > 607.095	non-Churner
3	If CRED_T-2 <= 98.379 and CRED_T-1 <= 99.626 and NCC_T > 1.529	Churner
4	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T <= 14.5	Churner
5	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T > 14.5	non-Churner
6	If CRED_T-2 <= 98.379 and CRED_T-1 > 99.626 and CRED_T-1 <= 104.8 and NCC_T-2 <= 1.071	Churner
7	If CRED_T-2 <= 98.379 and CRED_T-1 > 99.626 and CRED_T-1 <= 104.8 and NCC_T-2 > 1.071	non-Churner
8	If CRED_T-2 <= 98.379 and CRED_T-1 > 104.8 and CRED_T-1 <= 161.026	non-Churner
9	If CRED_T-2 <= 98.379 and CRED_T-1 > 104.8	Churner
10	If CRED_T-2 > 98.379	non-Churner
11	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T <= 14.5 and CRED_T <= 593.854	Churner
12	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T <= 14.5 and CRED_T <= 593.854 and NCC_T <= 0.936	Churner
13	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T <= 14.5 and CRED_T > 593.854 and NCC_T > 0.936	Churner
14	If CRED_T-2 <= 98.379 and CRED_T-1 > 95.849 and T_WEB_T > 14.5	non-Churner

Table 5.11: Average fidelity of the proposed SVM+NBTree using Case-SP

Sampling technique	Full Features	Reduced Features
Unbalanced	79.46	93.58
SMOTE	**93.46**	91.95
25% Undersampling	85.04	89.82
50% Undersampling	80.6	**97.63**
100% Oversampling	83.22	94.88
200% Oversampling	86.62	90.48
300% Oversampling	87.66	86.91
25% Undersampling + 100% Oversampling	71.54	92.36
50% Undersampling + 200% Oversampling	78.34	90.52

It is observed that the hybrid SVM+NBTree using *Case-SA* yielded the worst sensitivity when compared to other classifiers. The reason for such results is that the instances which

stand-alone NBTree could not correctly classify and standalone SVM could correctly classify indeed turned out to be the support vectors. Further, among these support vectors, many instances belong to churned customers. This fact is the reason behind *Case-SP* yielding better sensitivity compared to *Case-SA* dataset.

It is observed from the rules extracted that using the proposed hybrid SVM+NBTree using *Case-SP* dataset and reduced features, the *credit value* of the customers and the number of *online transactions* a customer performs, are the main driving elements for determining a customer to be loyal or churner. Most of the rules say that the customers with less value of the credit in any of the month may churn in future. Rules also indicate that the customers using internet (i.e. online transactions) less often may also churn in near future. Further, rules also imply that customers with high credit value and customers using online transactions are supposed to be loyal customers.

5.8 Conclusions

In this chapter a hybrid rule extraction approach from SVM is presented to predict churn in bank credit card customers. Since the dataset at hand is a highly unbalanced with 93.24% loyal and 6.76% churned customers, balancing techniques such as undersampling, oversampling, combinations of undersampling and oversampling and SMOTE are employed to balance the data. While solving the problems like churn prediction *sensitivity* is accorded high priority. Accordingly, by considering sensitivity alone, it is observed that the proposed hybrid SVM+NBTree using *Case-SP* with reduced features and balanced by SMOTE yielded the best sensitivity of 91.85%. The number of rules extracted using Case-SP data with reduced features is very less compared to the rules extracted using full feature data, resulting in improved comprehensibility of the system.

The following conclusions are made from this work;

- Feature selection using SVM-RFE selected the key features of the data.
- The number of support vectors extracted is also very less.

- *Case-SP* dataset represents the knowledge of SVM with respect to feature selection, support vector extraction and predicted class labels of support vector instances. Thus this method falls under eclectic approaches.
- Using *Case-SP* dataset makes rule extraction process faster.
- Because of the less number of features and instances i.e. *Case-SP* set, the number of rules extracted and the antecedents per rule is small, resulting in better comprehensibility of the system.
- Using *Case-P* and *Case-SP* it is ensured that the extracted rules indeed represent the knowledge learnt by the SVM.
- The extensive study done in this chapter can be considered as the study about the efficiency of SVM to deal with large scale unbalanced data.

Chapter 6

Modified Active Learning Based Approach for Rule Extraction from SVM

This Chapter presents an *eclectic active learning based rule extraction approach* which involves active learning to modify the training data. In the beginning section of the chapter motivation behind the proposed approach is presented. Later, section two provides the details about the proposed approach and active learning. Next sections in this chapter present the details of problems analyzed followed by Results and discussions. Final section concludes the chapter.

6.1 Motivation

Recently, Martens et al., (2009) proposed an active learning based rule extraction approach to generate rules from SVM. The basic principle of their proposed approach is the generation of synthetic data near support vector points. Later, these extra synthetic samples are combined with training set and the actual target values are replaced by the predictions given by the trained SVM. Finally, this modified data is used to train Decision Tree and RIPPER to generate rules. The efficiency of the proposed algorithm was evaluated on benchmark datasets and small scale datasets only. It is observed that real world applications usually have medium scale and unbalanced nature of samples. When the size of the data increases, the method proposed by Martens et al., (2009) is not feasible and there is possibility of generating more number of rules. More number of rules may provide transparency to the black box but it degrades the comprehensibility of the system. During the research study in this chapter modifications to ALBA (Martens et al., 2009) are proposed and the efficiency of the proposed approach is evaluated using medium scale unbalanced dataset. The finance applications analyzed using MALBA are; *Churn prediction* in bank credit card customers and *Fraud detection* in automobile insurance. The

financial problems solved during this research study is highly unbalance and medium scale.

6.2 Modified Active Learning Based Approach for Rule Extraction

In this research work, we propose a modified active learning based (MALBA) rule extraction procedure to extract rules from SVM using NBTree (Naive Bayes Tree) (Kohavi, 1996). The proposed approach is based on the Active Learning Based Approach proposed by Martens et al., (2009). They (Martens et al., 2009) used uniform distribution function to generate extra instances, which are supposed to be near support vectors as it is based on the distance calculated between the training instances and support vector instances. In this study we employed various distributions such as Normal Distribution and Logistic Distribution separately for generating extra synthetic data near support vectors instances. Later, these extra instances are clubbed with support vectors set instead of training set and predictions are obtained for newly generated data and support vectors. This modified data is then used to generate rules using NBTree. For comparative study rules are also generated using DT.

The proposed algorithm is composed of three phases, *feature selection phase, active learning phase* and *rule generation phase*. The proposed approach is depicted in Figure 6.1. As the real world datasets are high dimensional in nature, hence feature selection using SVM-RFE is employed to reduce the features set during *feature selection phase*. Later this reduced feature data is used in active learning phase to generated synthetic instances near support vectors. During rule generation phase the newly generated synthetic data with support vectors is used. Active learning phase consists of four steps as given below;

6.2.1 Featuer Selection phase

SVM-RFE (Guyon 2002) algorithm is employed for feature selection purpose which tries to extract key features from the data.

6.2.2 Active Learning Phase

Training data = D_{tr}

Using all features and reduced features separately, build SVM model. Sensitivity is accorded high priority to develop SVM model.

Step 1: Train SVM and obtain the support vectors using training data $D_{tr.}$

Calculate the average distance $distance_k$ *of training data to support vectors, in each dimension k.*

Step 2: **for** *k=1* to *n* **do**

$distance_k = 0$

for all support vectors SV_j **do**

for all training data instance *d* in D_{tr} **do**

$distance_k = distance_k + |d_k - SV_{j,k}|$

end for

end for

$$\text{distance}_k = \frac{\text{distance}_k}{\#SVs + N_{tr}}$$

end for

Proposed modified active learning based approach for generating synthetic instances. Generate extra data instances based on the average distance. The idea behind generating extra instances using distance is to make sure about the closeness of the generated instances to the support vectors set. Various MALBA architectures using normal and logistic distribution functions are employed separately.

Step 3: Randomly generate an extra data instance x_i close to support vectors

Generating extra instances in the range of -1 to 1

for i = 1 to 500/1000 **do**

for *k = 1* to *n* **do**

$$x_{i,k} = sv(j,k) + \left[(2 * rand - 1) \times \frac{disance_k}{2} \right] \; (0 < rand < 1]$$

end for

end for

Generating extra instances using Logistic distribution function

for *i* = 1 to 500/1000 **do**

for *k* = 1 to *n* **do**

$x_{i,k}$ = rand;

end for

end for

for *i* =1 to 500/1000 **do**

for *k* = 1 to *n* **do**

$u_{i,k}$ = 1 / (1+exp(-$x_{i,k}$));

end for

end for

Generating extra instances using Normal distribution function (Box-Muller 1958)

for *k* = 1 to *n* **do**

for *i* = 1 to 500/1000 **do**

$y_{i,k}$ = sqrt(-2 x log($x_{i,k}$)) x cos(2 x *pi* x ($x_{i+1,k}$));

$y_{i+1,k}$ = sqrt(-2 x log($x_{i,k}$)) x sin(2 x *pi* x ($x_{i+1,k}$));

end for

end for

Obtain class labels for generated data and support vectors set using the trained SVM model as an oracle.

Step 4: Provide a class label y_i for extra generated instances and support vector set using the trained SVM as oracle.

6.2.3 Rule Generation Phase

NBTree and DT algorithms are employed using modified dataset and efficiency of the rules is then evaluated in terms of accuracy, fidelity and number of rules.

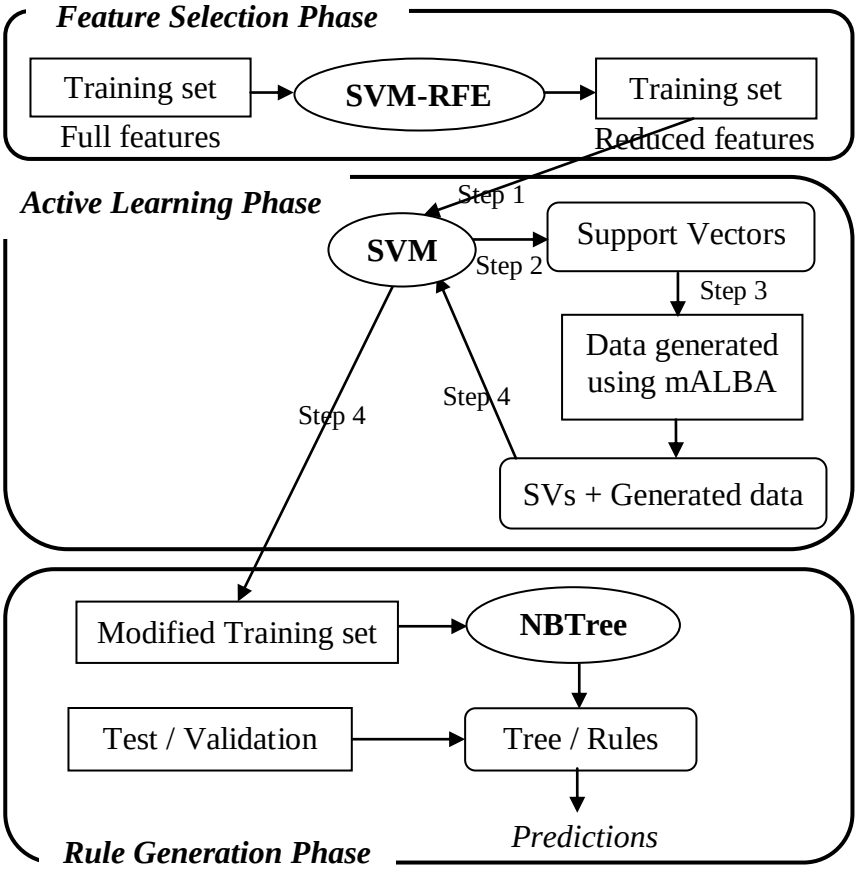


Figure 6.1: Architecture of the proposed rule extraction approach

The current study in this study is different from ALBA (Martens et al., 2009) approach in several ways, such as;

- They generated the instances using $\left[(rand-0.5)*\frac{disance_k}{2}\right]$, which generated the instances near support vectors in the range of -0.5 to 0.5, whereas in this research study we used $\left[(2*rand-1)\times\frac{disance_k}{2}\right]$, which generates the data around the support vectors in the range of -1 to 1, therefore the region around support vector set is increased.
- They appended the generated data to training set, whereas in this study the generated data is appended to the support vectors set. Hence, the complexity of the rule extraction approach is minimized without compromising the performance of the classifier.
- They employed C4.5 and RIPPER for rule generation purpose, whereas in this study we employed NBTree and J48 (decision tree) for rule generation.
- They employed only uniform distribution for extra data generation purpose, whereas in this study, we employed Normal Distribution function and Logistic Distribution function for data generation.
- They analyzed benchmark and small scale problems only, whereas in this research study, we analyzed real time finance applications *viz., Churn prediction* in Bank credit card customers' and *Fraud detection* in automobile insurance. The datasets analysed in this study are unbalanced in nature and are medium scale in size.
- They did not evaluate the efficiency of feature selection using SVM, whereas in this study the efficiency of SVM for feature selection is also analysed.

6.3 Finance applications analyzed

Two most important finance applications are analyzed in this research study those are *churn prediction in bank credit card customers* and *fraud detection in automobile insurance*. The churn prediction dataset is obtained from Business Intelligence Cup 2004.

The details about this dataset are provided in Section 5.5 in Chapter 5. Fraud detection in automobile insurance dataset is obtained from Pyle (1999). This dataset is also highly unbalanced with 94% legitimate and 6% fraudulent customers' data. Description and pre-processing of the insurance fraud dataset is presented below;

6.3.1 Fraud Detection in Automobile Insurance Dataset

This is the only available fraud detection dataset in automobile insurance and it is provided by Angoss Knowledge Seeker software Pyle (1999). This dataset contains 11338 records from January 1994 to December 1995, and 4083 records from January 1996 to December 1996. It has a 6% fraudulent and 94% legitimate cases, with an average of 430 claims per month. The original dataset has 6 numerical features and 25 categorical features, including the binary class label (fraud or legal). Description of original fraud detection dataset is presented in Table 6.1.

6.3.2 Pre-Processing

It is observed that the *age* feature in the dataset appeared twice (see Table 6.1, row 12 and 24) in numerical and categorical form as well. Hence, the age feature with numerical values is removed from the data to reduce the complexity caused by too many unique values it possesses. Further, the features *Year, Month, Week of the month* and *Day of week* represent the date of the accident and the features *Month claimed, Week of the month claimed* and *Day of week claimed* represent the date of the insurance claim. Thus, a new feature *Gap* is derived from seven features, those are; *Year, Month, Week of the month, Day of week, Month claimed, Week of the month claimed* and *Day of week claimed*. The feature *Gap* represents the time difference between the accident occurrence and insurance claim. Thus 24 variables which included some derived variables are selected for further study and are presented in Table 6.2. Hence, we have 15420 samples with 24 predictor variables and 1 class variable.

Table 6.1: Feature Information of the Insurance data used (Pyle 1999)

#	Feature name	Description
1	Month	Month in which accident took place
2	Week of Month	Accident week of month
3	Day of Week	Accident day of week
4	Month Claimed	Claim month
5	Week of Month Claimed	Claim week of month
6	Day of Week Claimed	Claim day of week
7	Year	1994,1995 and 1996
8	Make	Manufacturer of the car(19 companies)
9	Accident Area	Rural or Urban
10	Gender	Male or Female
11	Marital Status	Single, Married, Widow and Divorced
12	Age	Age of policy holder
13	Fault	Policy Holder or Third Party
14	Policy Type	Type of the policy (1 to 9)
15	Vehicle Category	Sedan, Sport or Utility
16	Vehicle Price	Price of the vehicle with 6 categories
17	Rep. Number	ID of the person who process the claim(16 ID's)
18	Deductible	Amount to be deducted before claim disbursement
19	Driver Rating	Driving Experience with 4 categories
20	Days: Policy Accident	Days left in policy when accident happened
21	Days: Policy Claim	Days left in policy when claim was filed
22	Past number of Claims	Past number of claims
23	Age of Vehicle	Vehicle's age with 8 categories
24	Age of Policy Holder	Policy holder's age with 9 categories
25	Policy Report Filed	Yes or no
26	Witness Presented	Yes or no
27	Agent Type	Internal or External
28	Number of Supplements	Number of supplements
29	Address Change Claim	No of times change of address requested
30	Number of Cars	Number of cars
31	Base Policy(BP)	All perils, Collision or Liability
32	Class	Fraud found (yes or no)

Table 6.2: Feature Information of the *pre-processed* Insurance data

#	Feature name	Description
1	*Gap*	Time difference of accident and insurance claim
2	*Make*	Manufacturer of the car(19 companies)
3	*Accident Area*	Rural or Urban
4	*Gender*	Male or Female
5	*Marital Status*	Single, Married, Widow and Divorced
6	*Fault*	Policy Holder or Third Party
7	*Policy Type*	Type of the policy (1 to 9)
8	*Vehicle Category*	Sedan, Sport or Utility
9	*Vehicle Price*	Price of the vehicle with 6 categories
10	*Rep. Number*	ID of the person who process the claim(16 ID's)
11	*Deductible*	Amount to be deducted before claim disbursement
12	*Driver Rating*	Driving Experience with 4 categories
13	*Days: Policy Accident*	Days left in policy when accident happened
14	*Days: Policy Claim*	Days left in policy when claim was filed
15	*Past number of Claims*	Past number of claims
16	*Age of Vehicle*	Vehicle's age with 8 categories
17	*Age of Policy Holder*	Policy holder's age with 9 categories
18	*Policy Report Filed*	Yes or no
19	*Witness Presented*	Yes or no
20	*Agent Type*	Internal or External
21	*Number of Supplements*	Number of supplements
22	*Address Change Claim*	No of times change of address requested
23	*Number of Cars*	Number of cars
24	*Base Policy(BP)*	All perils, Collision or Liability
25	*Class*	Fraud found (yes or no)

6.3.3 Literature survey of fraud detection problem

Fawcett and Provost (1997) proposed automatic design of user profiling methods for the purpose of fraud detection. They employed data mining techniques and concluded that the automatic approach performs better than hand-crafted methods for detecting frauds. A meta classifier system is proposed for fraud detection (Stolfo et al., 1997a, 1997b). This merges the results obtained from local fraud detection tools at different corporate sites to yield a more accurate global tool.

Chan et al. (1999) and Stolfo et al. (2000) proposed an extension to this approach i.e. *scalable distributed data mining model* to evaluate classification techniques using a

realistic cost model. They reported that, it significantly improved the performance of data mining algorithms in stand-alone mode and in combination as well.

Many intelligent techniques were proposed and applied for fraud detection problem. such as, Association Rule Mining and Radial Basis Function Network (Brause et al., 1999), Fuzzy Logic Control (FLC) (Stefano and Giesella, 2001), Principal Component Analysis (Brockett et al., 2002), AdaBoosted naïve Bayes (Viaene et al., 2004) and so on. Later, Phua et al. (2004) proposed a fraud detection method using stacking-bagging approach which involves Backpropagation Neural Network (BPNN) with naïve Bayesian (NB) and C4.5 algorithms. They demonstrated that stacking-bagging performs slightly better than the best performing bagged algorithm i.e. C4.5. Wheeler and Aitken (2000) and Phua et al., (2005) have explored multiple classification techniques for fraud detection.

It is observed that using only support vectors set would do the rule extraction, but generating synthetic instances near support vectors without changing the decision boundary of SVM provides more samples for rule generation algorithm to learn from. Hence, more data is fed to rule generating algorithm and it is observed that the more generalized rules are generated.

To evaluate and compare the efficiency of the proposed approach, five different methods leading to creation of five different datasets are prepared and used for rule generation purpose as follows;

1. Original ALBA (range [-0.5 to 0.5], Training set)
2. ALBA (SVs) (range [-0.5 to 0.5], Support Vectors set)
3. MALBA (range [-1 to 1], Support Vectors set)
4. MALBA (Norm) (Normal Distribution, Support Vectors set)
5. MALBA (Logistic) (Logistic Distribution, Support Vectors set)

Dataset 1 in the above list corresponds to the original ALBA proposed by Martens et al. (2009), where extra instances are appended to training set. ALBA works well with smaller datasets and an extensive study is presented by Martens et al. (2009). The main drawback of their approach is that real world problems are unbalanced and medium sized, where

appending the extra instances to training set would make the rules complex and there is a possibility that it generates incomprehensible rules. In this study a modified ALBA is proposed where extra instances are appended to support vectors set only instead of training set. Two different financial problems are solved using the proposed approach and the datasets are highly imbalance and medium scale. Datasets 2-5 in the above list represents the experiments where generated data is appended to support vector set only. It is observed from the empirical study that appending the generated data to support vectors set and using it for rule generation purpose produces more generalized and comprehensible rules.

6.4 Results and Discussions

Identifying potential churners correctly is the basic intension of many business decision makers. Hence, they place high emphasis on sensitivity alone which contributes towards the bottom-line of the fundamental customer relationship management (CRM). Consequently in this chapter also, sensitivity is accorded top priority ahead of specificity and accuracy. We used the SVM library viz., LibSVM (Chan and Lin, 2001) for SVM model building and support vector extraction. LibSVM is an integrated software for support vector classification and is developed in MATLAB. Data generation using MALBA is implemented in MATLAB. RapidMiner4.5 community edition (Mierswa et al., 2006) is used for generating NBTree and DT (J48).

Average results of the experiments under 10-fold cross validation, against validation set, t-test values based on sensitivity, time taken and the number of rules extracted are presented in results tables. Using sensitivity, the classifiers are compared using t-test at $n_1+n_2-2=10+10-2=18$ degrees of freedom at 10% level of significance. The tabulated value of t-test for 18 degrees of freedom at 10% level of significance is 1.73. That means, if the computed t-test value between two different classifiers is more than 1.73, then we can say that the difference between techniques is statistically significant and otherwise not significant. Hyphen in the Tables for t-test values represents the best classifier. Tables 6.3 through 6.16 show the average results obtained using rules extracted from NBTree. Results obtained from DT are presented in Tables 6.17 through 6.28. The tree obtained using

NBTree has naïve bayes classifiers at leaf nodes that indicates the probability of each class available in the dataset used instead of prediction of any single class. For better understanding of the tree we modified the rules and the class with higher probability assigned by the naïve bayes classifier at leaf node is considered the consequent of the rule.

6.4.1 Churn Prediction using SVM+NBTree

Table 6.3 and 6.4 present the average results obtained using 500 and 1000 extra instances for Churn Prediction problem using all the features of the data, respectively. It is observed that the proposed approach MALBA with 500 extra instances yielded highest sensitivity of 79.35%, which is statistically significant and better compared to SVM and original ALBA (Martens et al., 2009). Time taken and the number of rules extracted using NBTree is 5.2 seconds and 11.1, respectively.

Table 6.3: Average Results of Churn Prediction SVM+NBTree (500 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	64.65	80.63	79.55	4.786	7.9	
ALBA	67.7	84.52	83.38	3.125	39.1	31.8
ALBA(SVs)	76.55	82.74	82.32	0.755	14.7	13.3
MALBA	**79.35**	**79.16**	**79.17**	**-**	**5.2**	**11.1**
MALBA(Norm)	78.45	81.63	81.4	0.288	14.5	14.3
MALBA(Logistic)	74.05	80.18	79.75	1.445	12.8	19

Table 6.4: Average Results of Churn Prediction SVM+NBTree (1000 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	68.05	84.75	83.62	2.407	52	27.6
ALBA(SVs)	73.25	82.93	82.27	0.978	17.5	13.9
MALBA	75.9	83.3	82.87	0.239	**6.9**	12.9
MALBA(Norm)	**76.63**	**82.72**	**82.31**	**-**	13.7	**12.7**
MALBA(Logistic)	75.2	79.62	79.53	0.519	12.3	17.6

Using SVM-RFE feature selection is carried out and 6 most important features are selected. The selected features are; *CRED_T (Credit in month T), CRED_T-1 (Credit in month T-1), CRED_T-2 (Credit in month T-2), NCC_T (Number of Credit Cards in month*

T), NCC_T-2 (Number of Credit Cards in month T-2) and *T_WEB_T (Number of web transactions in month T)* (refer Table C.10 in Appendix C). Tables 6.5 and 6.6 present the average results yielded using 500 and 1000 extra generated instances with reduced features for Churn prediction problem, respectively. It is observed that our proposed approach MALBA with 1000 extra instances outperforms other classifiers tested and yielded highest sensitivity of 84.6%. t-test values show that the hybrids are not statistically significant i.e. all the hybrids using reduced features and extra instances behave similarly. It is observed from the results that feature selection prior to rule generation improves the performance of the rules. Rules obtained by MALBA using NBTree with reduced feature set is presented in Table 6.7.

Table 6.5: Average Results of Churn Prediction Feature Selection + SVM+NBTree (500 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	82.85	74.25	74.79	0.849	3.8	
ALBA	**84.00**	**74.79**	**75.41**	-	14	46.8
ALBA(SVs)	82.55	75.13	75.63	1.574	6.3	19.6
MALBA	83.18	75.77	76.43	0.527	6.3	22.8
MALBA(Norm)	82.30	75.08	75.57	1.156	6.1	20
MALBA(Logistic)	83.35	74.56	75.15	0.443	**5.8**	**14.9**

Table 6.6: Average Results of Churn Prediction Feature Selection + SVM+NBTree (1000 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	84.40	74.16	74.85	0.121	16.8	48.1
ALBA(SVs)	83.30	74.34	74.95	0.972	6.6	24.6
MALBA	**84.60**	**73.32**	**74.14**	**-**	**4.9**	21.2
MALBA(Norm)	82.15	75.36	75.82	1.799	6.3	21.4
MALBA(Logistic)	82.95	75.36	75.87	1.267	6	**16**

Table 6.7: Rule Extracted for Churn Prediction using NBTree (Reduced Features)

Rule #	Antecedents	Consequent
1	If *CRED-T* <= 594.94 and *CRED-T-2* <= 96.005 and *CRED-T-1* <= 100.12	Churn
2	If *CRED-T* <= 594.94 and *CRED-T-2* <= 96.005 and *CRED-T-1* > 100.12 and *CRED-T-1* <= 147.9	Churn
3	If *CRED-T* <= 594.94 and *CRED-T-2* <= 96.005 and *CRED-T-1* > 100.12 and *CRED-T-1* > 147.9	Non-Churn
4	If *CRED-T* is [579 to 594.94] and *CRED-T-2* > 96.005 and *T-WEB-T* <= 8.5 and *CRED-T-1* <= 122.5	Churn
5	If *CRED-T* is [579 to 594.94] and *CRED-T-2* > 96.005 and *T-WEB-T* <= 8.5 and *CRED-T-1* > 122.5	Non-Churn
6	If *CRED-T* is [579 to 594.94] and *CRED-T-2* > 96.005 and *T-WEB-T* <= 8.5	Churn
7	If *CRED-T* <= 594.94 and *CRED-T-2* is [96.005 to 105.2] and *T-WEB-T* > 8.5	Non-Churn
8	If *CRED-T* > 594.94	Non-Churn

It is also observed that rules using full features of the data considers *customer's margin* as the most important element to decide about *about-to-churn* customers. Feature selection using SVM-RFE selects *customers' credit* information as important and is an important factor to know *about-to-churn* customers. 4 rules are extracted for churner class and 4 rules are extracted for non-churner class. It is also observed that web transaction done by a customer during current and previous month also plays very important role in churn prediction in credit card customers.

The fidelity obtained using original ALBA and various MALBA hybrids for Churn Prediction problem with full and reduced features is presented in Table 6.8. It is observed that with full features, proposed hybrid rule extraction techniques behave more than 80% like SVM and with reduced features the fidelity obtained is more than 93%. In other words, hybrids with reduced features behave more like SVM compared to that of the behaviour of the hybrid using all the features of the data.

Table 6.8: Average Fidelity for Churn Prediction SVM+NBTree

Model	Full Features		Feature Selection	
	500	1000	500	1000
ALBA	83.28	81.64	95.89	95.89
ALBA(SVs)	80.88	79.03	93.9	93.47
MALBA	82.65	79.1	93.58	91.92
MALBA(Norm)	82.16	82.42	93.63	93.59
MALBA(Logistic)	80.91	80.36	93.65	94.2

6.4.2 Insurance fraud detection using SVM+NBTree

Tables 6.9 and 6.10 present the average results obtained using NBTree for automobile insurance fraud detection problem with 500 and 1000 extra instances using full features of the data, respectively. It is observed that using 500 extra instances our proposed approach MALBA yielded best sensitivity of 75.73%. It has taken 6.2 seconds time to generate rules and average number of rules extracted are 10. Based on the t-test values it is observed that using full features of the data MALBA stands best and its performance is statistically significant as well.

Top seven features selected for Results and discussions are; *Make, Accident Area, Marital Status, Fault, Vehicle Category, Age of Vehicle* and *Base Policy*. Average results obtained using 500 and 1000 instances with reduced features are presented in Table 6.11 and 6.12, respectively. It is observed that our proposed approach MALBA yielded best sensitivity of 88.48% with 1000 extra instances and consumed least time of 3.5 seconds for extracting on average 17 rules. It is observed that again hybrids' performance is better with reduced features than that of the corresponding hybrids' performance using full features of the data. Rules extracted by MALBA using NBTree with reduced feature set is presented in Table 6.13. It is observed from the rules that *marital status, vehicle* and *base policy* are the feature playing vital role in deciding about the fraud.

Table 6.9: Average Results of Insurance Fraud Detection
using SVM+NBTree (500 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	70.76	63.20	63.65	3.554	6.8	
ALBA	70.65	62.68	63.16	3.082	51.3	29
ALBA(SVs)	74.70	60.46	61.34	0.537	13.5	10.9
MALBA	**75.73**	**59.76**	**60.62**	**-**	**6.2**	**10**
MALBA(Norm)	73.41	59.91	60.72	1.793	16.1	18
MALBA(Logistic)	74.35	59.24	60.15	0.873	21.1	16

Table 6.10: Average Results of Insurance Fraud Detection
using SVM+NBTree (1000 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	70.22	62.74	63.19	1.575	62.3	44
ALBA(SVs)	74.11	58.78	59.42	0.236	18	16.1
MALBA	**75.21**	**54.69**	**55.97**	**-**	**13.2**	**15**
MALBA(Norm)	74.29	59.15	60.06	0.271	27	28
MALBA(Logistic)	75.05	58.33	59.84	0.04	23.9	17

Table 6.11: Average Results of Insurance Fraud Detection Feature Selection
SVM+NBTree (500 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	87.68	56.36	58.21	0.209	3.7	
ALBA	88.00	56.27	58.17	0.118	12.6	30
ALBA(SVs)	**88.43**	**55.13**	**57.19**	**-**	5.8	12.2
MALBA	88.22	55.64	57.59	0.059	2.5	13
MALBA(Norm)	88.00	55.74	57.67	0.119	8.6	16
MALBA(Logistic)	88.16	54.77	56.73	0.074	9.8	15

Table 6.12: Average Results of Insurance Fraud Detection Feature Selection
SVM+NBTree (1000 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	87.84	56.56	58.44	0.177	12.9	26
ALBA(SVs)	88.05	55.33	57.29	0.115	6.3	15.3
MALBA	**88.48**	**55.60**	**57.57**	**-**	3.5	17
MALBA(Norm)	88.11	55.88	57.82	0.103	9.8	18
MALBA(Logistic)	88.38	55.34	57.34	0.028	10.6	16

It is observed from the results that the proposed hybrid MALBA using reduced feature data and 1000 extra instances yielded the best sensitivity of 88.48%. Once again it is observed that the hybrids using reduced feature data perform better than that of their corresponding hybrids with full feature data. Based on t-test values it is observed that all the hybrids using reduced feature data behave similarly i.e. hybrids using reduced feature are statistically insignificant.

Table 6.13: Rules Extracted for Insurance Fraud Detection using NBTree (reduced features)

Rule #	**Antecedents**	**Consequent**
1	If *Marital Status* is Single and *Base Policy* is All Perils and *Age of Vehicle* is Less than 6 years then	non-Fraud
2	If *Marital Status* is Single and *Base Policy* is Collision/Liability then	Fraud
3	If *Marital Status* is Single and *Base Policy* is All Perils and *Age of Vehicle* is More than 6 years then	non-Fraud
4	If *Marital Status* is Married/Widow/Divorced and *Manufacturer* is Top 9 from manufacturers list and *Vehicle Category* is Sedan and *Fault* is of Policy Holder	Fraud
5	If *Marital Status* is Married/Widow/Divorced and *Manufacturer* is Top 9 from manufacturers list and *Vehicle Category* is Sedan and *Fault* is of Third Party	non-Fraud
6	If *Marital Status* is Married/Widow/Divorced and *Manufacturer* is Top 9 from manufacturers list and *Vehicle Category* is Sports/Utility and *Age of Vehicle* is More than 3 years	Fraud
7	If *Marital Status* is Married/Widow/Divorced and *Manufacturer* is After 9th in the list	Fraud

Fidelity obtained using Insurance Fraud detection data with and without feature selection is presented in Table 6.14. It is observed based on fidelity that using reduced features the hybrids perform more than 97% exactly like SVM and with full features slightly more than 87% only.

Table 6.14: Average Fidelity for Insurance Fraud Detection SVM+NBTree

Model	Full Features		Feature Selection	
	500	**1000**	**500**	**1000**
ALBA	92.78	92.81	99.47	99.5
ALBA(SVs)	89.22	88.25	96.93	97.09
MALBA	88.82	88.97	97.08	96.69
MALBA(Norm)	88.13	87.78	97.27	97.48
MALBA(Logistic)	87.87	88.4	97.04	96.77

6.4.3 Churn Prediction using SVM+DT

Rules are also extracted using decision tree algorithm as well. Tables 6.15 and 6.16 present the results obtained using 500 and 1000 extra instances with full feature data for Churn prediction problem, respectively. It is observed that with 500 extra instances the proposed approach MALBA (Norm) yielded best sensitivity of 77.25%. The time taken to generate rules using 500 extra instances using MALBA (Norm) is 10.3 seconds and the average number of rules extracted are 46.8. based on t-test value it is observe that using 500 extra instances MALBA (Norm) outperformed stand-alone SVM, original ALBA and ALBA (SVs). Whereas using 1000 extra instances it is observed hat all the hybrids perform similarly and MALBA (Logistic) stands best with 77.2% sensitivity.

Table 6.15: Average Results of Churn Prediction
using SVM+DT (500 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	64.65	80.63	79.55	5.414	7.9	
ALBA	65.2	82.65	81.48	4.541	37.6	117.3
ALBA(SVs)	72.7	80.38	79.25	1.725	10.7	92.9
MALBA	75.05	75.99	76.43	0.914	10.4	85.5
MALBA(Norm)	**77.25**	**82.64**	**82.28**	**-**	10.3	46.8
MALBA(Logistic)	75.2	83.19	82.71	0.531	**9.9**	**29**

Table 6.16: Average Results of Churn Prediction
using SVM+DT (1000 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	64.05	83	81.72	4.123	40.3	127
ALBA(SVs)	71.65	79.71	79.17	1.679	10.4	109.5
MALBA	72.85	81.3	80.7	1.432	11.1	96.5
MALBA(Norm)	76.1	82.91	82.45	0.337	11	54.9
MALBA(Logistic)	**77.2**	**80.58**	**80.35**	**-**	**10.2**	**32.9**

Tables 6.17 and 6.18 present the average results obtained using 500 and 1000 extra instances with reduced features for Churn prediction problem, respectively. It is observed that the proposed hybrid MALBA (SVs) with 1000 extra instances yielded best sensitivity of 83.4%, whereas MALBA (SVs) with 500 extra instances yielded sensitivity of 83.05%. It is observed that the hybrid perform better with reduced features compared to corresponding hybrids' performance with full features. Based on t-test values it is observed that all the classifiers using reduced features perform similarly only. Table 6.19 presents the sample rules extracted using DT with reduced feature data for Churn prediction problem. Similar kind of results observed when rules are generated using NBTree and rules generated using DT. Hybrids using reduced feature data perform better than their corresponding hybrids using full feature data.

Table 6.17: Average Results of Churn Prediction Feature Selection
SVM+DT (500 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	82.85	74.25	74.79	0.206	3.8	
ALBA	81.90	74.59	75.09	1.167	7.3	77.8
ALBA(SVs)	**83.05**	**75.46**	**75.97**	-	5.8	25
MALBA	82.35	76.44	76.84	0.655	5.8	30.1
MALBA(Norm)	82.30	76.36	76.76	0.794	5.8	26.2
MALBA(Logistic)	82.85	75.50	76.00	0.211	5.8	18.4

Table 6.18: Average Results of Churn Prediction Feature Selection
SVM+DT (1000 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	81.75	74.50	74.99	2.031	7.6	93.1
ALBA(SVs)	**83.40**	**71.47**	**72.26**	-	5.8	36.1
MALBA	82.89	75.32	75.83	0.061	5.78	40.78
MALBA(Norm)	82.00	75.41	75.85	1.46	5.8	39.1
MALBA(Logistic)	82.55	75.41	75.89	1.078	5.8	19.5

Rules extracted using DT also imply that among 6 selected features it is *Credit value* and *Web transactions in current and previous* month plays an important role in predicting churn in bank credit card customers.

Table 6.19: sample Rules Generated for Churn Prediction using DT with reduced features

Rule #	**Antecedents**	**Consequent**
1	If *CRED-T* <= 594.78 and *CRED-T-2* <= 97.81 and *T-WEB-T* <=5	Churner
2	If *CRED-T* <= 594.78 and *CRED-T-2* <= 97.81 and *T-WEB-T* > 5	non-Churner
3	If *CRED-T* <= 594.78 and *CRED-T-2* > 97.81 and *CRED-T-1* <= 92.81	Churner
4	If *CRED-T* <= 594.78 and *CRED-T-2* > 97.81 and *CRED-T-1* > 92.81	Churner
5	If *CRED-T* >= 580.6 and *CRED-T* <= 594.78 and *CRED-T-2* >= 97.81 and *CRED-T-2* <=102.7 and *CRED-T-1* >= 92.81 and *CRED-T-1* <= 95.31 and *T-WEB-T* <=1	Churner
6	If *CRED-T* >= 580.6 and *CRED-T* <= 594.78 and *CRED-T-2* >= 97.81 and *CRED-T-2* <= 102.7 and *CRED-T-1* > 92.81	non-Churner

The fidelity obtained using full features and reduced features are presented in Table 6.20. It is observed based on fidelity that reduced feature data yielded better fidelity i.e. more than 92% compared to full feature data i.e. more than 83% only. Considering fidelity also it is observed that hybrids with reduced features behave more like SVM than that of hybrids with all features.

Table 6.20: Average Fidelity for Churn Prediction SVM+DT

Model	Full Features		Feature Selection	
	500	**1000**	**500**	**1000**
ALBA	84.46	85.66	96.78	96.9
ALBA(SVs)	82.49	83.01	92.95	91.39
MALBA	83.46	83.7	94.03	93.19
MALBA(Norm)	83.73	84.37	93.94	93.87
MALBA(Logistic)	82.24	83.15	93.967	94.11

6.4.4 Insurance fraud detection using SVM+DT

Tables 6.21 and 6.22 present the average results obtained using 500 and 1000 extra instances using DT, respectively. It is observed that our proposed hybrid MALBA with 500 extra instances yielded best sensitivity of 74.27%. It consumed 8.8 seconds to extract rules and it extracted 48 rules on average. t-test value indicate that the classifiers analyzed perform similar only i.e. all the classifiers are statistically insignificant.

Table 6.21: Average Results of Insurance Fraud Detection using SVM+DT (500 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	70.76	63.20	63.65	2.033	6.8	
ALBA	71.24	63.19	63.67	1.565	40.8	187
ALBA(SVs)	72.76	61.57	62.25	0.752	8.8	50.2
MALBA	**74.27**	**61.00**	**62.08**	-	8.8	48
MALBA(Norm)	71.35	62.87	63.38	1.281	8.8	49
MALBA(Logistic)	72.49	61.39	62.06	0.801	8.8	48

Table 6.22: Average Results of Insurance Fraud Detection using SVM+ DT (1000 Extra Instances)

Model	Validation					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	69.62	63.65	64.00	1.578	45	196
ALBA(SVs)	70.97	62.25	62.77	0.974	8.8	63.6
MALBA	**73.68**	**59.13**	**60.00**	-	8.8	67
MALBA(Norm)	70.70	62.29	62.80	1.192	8.8	70
MALBA(Logistic)	71.08	61.80	62.36	0.99	8.8	64

Tables 6.23 and 6.24 present the average results obtained using 500 and 1000 instances with reduced features for insurance fraud detection problem, respectively. It is observed that our proposed hybrid MALBA (Logistic) with 500 extra instances yielded best sensitivity of 87.99%, whereas the proposed MALBA with 1000 extra instances stand second in the list with 87.89% sensitivity. The average time taken for MALBA (Logistic) to generated rules is 4.7 seconds and the average number of rules extracted are 17. Table 6.25 presents the rules extracted using MALBA (Logistic) with reduced features. Once again it is observed that hybrid with reduced feature yielded better sensitivity compared to their corresponding hybrid with full feature.

Table 6.23: Average Results of Insurance Fraud Detection
Feature Selection+SVM+DT (500 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
SVM	87.68	56.36	58.21	0.086	3.7	
ALBA	87.78	56.48	58.36	0.056	5.7	37
ALBA(SVs)	87.84	55.96	57.87	0.042	4.7	16.3
MALBA	87.67	56.03	57.92	0.089	4.7	17
MALBA(Norm)	87.62	56.58	58.44	0.101	4.7	21
MALBA(Logistic)	**87.99**	**56.58**	**58.46**	-	4.7	17

Table 6.24: Average Results of Insurance Fraud Detection
Feature Selection SVM+DT (1000 Extra Instances)

Model	**Validation**					
	Sensitivity	**Specificity**	**Accuracy**	**t-test**	**Time**	**Rules**
ALBA	87.57	56.48	58.34	0.09	5.7	41
ALBA(SVs)	87.62	56.37	58.25	0.074	4.7	23.3
MALBA	**87.89**	**55.81**	**57.74**	-	4.7	24
MALBA(Norm)	87.77	57.22	59.02	0.032	4.7	26
MALBA(Logistic)	87.84	56.31	58.20	0.015	4.7	22

Table 6.25: Rules Extracted for Insurance Fraud Detection using Decision Tree

Rule #	**Antecedents**	**Consequent**
1	If Policy Holder is at *Fault* and *Base Policy* is All Perils and *Vehicle Category* is Sedan	non-Fraud
2	If Policy Holder is at *Fault* and *Base Policy* is All Perils and *Vehicle Category* is Sports/Utility and *Age of Vehicle* is Less than 7	Fraud
3	If Policy Holder is at *Fault* and *Base Policy* is All Perils and *Vehicle Category* is Sports/Utility and *Age of Vehicle* is More than 7 and *Accident Area* is Rural and *Manufacturer* is Top 4 in the list	non-Fraud
4	If Policy Holder is at *Fault* and *Base Policy* is All Perils and *Vehicle Category* is Sports/Utility and *Age of Vehicle* is More than 7 and *Accident Area* is Rural and *Manufacturer* is After Top 4 in the list	Fraud
5	If Policy Holder is at *Fault* and *Base Policy* is All Perils and *Vehicle Category* is Sports/Utility and *Age of Vehicle* is More than 7 and *Accident Area* is Urban	Fraud
6	If Third Party is at *Fault*	non-Fraud
7	If Policy Holder is at *Fault* and *Base Policy* is Collision/Liability and *Vehicle Category* is Sedan and *Base Policy* is All Perils/Collision and *Marital Status* is Single	non-Fraud
8	If Policy Holder is at *Fault* and *Vehicle Category* is Sedan and *Marital Status* is Married/Widowed/Divorced and *Manufacturer* is Top 3 in the list and *Age of Vehicle* is Less than 6	Fraud
9	If Policy Holder is at *Fault* and *Vehicle Category* is Sedan and *Marital Status* is Married/Widowed/Divorced and *Manufacturer* is Top 3 in the list and *Age of Vehicle* is More than 6	non-Fraud
10	If Policy Holder is at *Fault* and *Vehicle Category* is Sedan and *Marital Status* is Married/Widowed/Divorced and *Manufacturer* is After Top 3 in the list	Fraud
11	If Policy Holder is at *Fault* and *Base Policy* is Collision/Liability and *Vehicle Category* is Spots/Utility	Fraud
12	If Policy Holder is at *Fault* and *Base Policy* is Collision/Liability and *Vehicle Category* is Sedan and *Base Policy* is Liability	Fraud

Average fidelity obtained is presented in Table 7.26. It is observed that the fidelity obtained using reduced feature is much better with more than 97.7% than that of the fidelity yielded using full feature data and it is more than 88.15% only. It is observed from the results that hybrids perform best with reduced features than that of the hybrids using full features.

Table 6.26: Average Fidelity for Insurance Fraud Detection SVM+DT

Model	Full Features		Feature Selection	
	500	1000	500	1000
ALBA	91.66	91.68	99.57	99.56
ALBA(SVs)	88.37	88.92	97.71	97.72
MALBA	88.15	85.39	97.81	97.81
MALBA(Norm)	88.43	85.88	98.19	98.08
MALBA(Logistic)	88.63	88.08	97.89	97.93

6.4.5 Overall Observations

It is observed that the data generated using MALBA with Normal distribution perform better than the original ALBA (Martens et al., 2009), where generated instances were combined with training set. Generated instances with training set results in increase in the number of instances in training set and add to the complexity of the rule extraction algorithm.

It is observed that the time consumed and the number of rules extracted using proposed hybrids ALBA with support vector set, MALBA, MALBA using Normal distribution and MALBA using Logistic distribution are very much less compared to the original ALBA (Martens et al., 2009).

It is observed that the hybrids perform better with reduced feature data as compared to full feature set with respect to sensitivity, number of rules, size of the rules and the time taken for rule extraction.

It is observed from the empirical results that rules generated using NBTree perform better compared to that of the rules extracted using DT in the following ways;

- The accuracy obtained by NBTree rules is better than that of DT.
- The time taken for NBTree generation is less than that of DT.
- The number of rules extracted using NBTree are less than that of DT.

- The length of the rules extracted using NBTree is smaller compared to that of DT generated rules.
- Comprehensibility of the rules is increased and the knowledge representation of trained SVM is improved with small length and less number of rules.

It is observed that rules can provide better comprehensibility to the management and management can consider these rules as early warnings and can make proper and in time policies to avoid huge losses because of the churn.

- It is observed from the rules that for Churn prediction problem, the *Credit in the account during current month and previous two months* are the main driving elements to determine customer churn.
- It is also observed that *web transactions* during current month and previous month also sometimes helps in deciding customer churning, likewise management can take precautionary steps to avoid such churns.
- *Credit card transactions* in current month also indicate churn propensity of customers.

For Fraud detection problems also it is the comprehensibility of the system which matters a lot for the service providing industry like Banks and Insurance Agency. Because of the feature selection in first step, the rules generated are small and less in number resulting in improved comprehensibility of the system.

- It is observed based on the rules extracted that, *Marital Status* and *Base policy* are the main driving elements for frauds.
- It is observed that fraud is more likely with *unmarried* or *single* customers.
- It is also observed that most of the accidents, claims and frauds are about *Sports* or *Utility* vehicles only, whereas claims for *Sedan* category vehicles tend to be non-frauds.

- It is also observed that when the accident took place because of the third party instead of the policy-holder, it tends to be a non-fraud case, whereas if the *fault* is of the policy-holder then there may be chance of fraud.

6.5 Conclusions

In this chapter, we presented a research study about the modified active learning based approach (MALBA) for rule extraction from SVM to solve customer Churn prediction in Bank Credit Cards and Insurance Fraud detection. In this study we employed feature selection using SVM-RFE algorithm during first phase, then extra instances are generated during active learning phase and during final phase rule extraction is employed. For Results and discussions in this chapter, we implemented original ALBA (Martens et al. 2009), MALBA, MALBA using Normal distribution and MALBA using Logistic distribution. Rules are extracted using NBTree and DT separately. These rules can also be considered as early warning system by the management. Churn prediction dataset is highly unbalanced data with 93% good customers and 7% churned customers. Insurance fraud detection dataset is also highly unbalanced with 94% legitimate cases and 6% fraudulent cases. While solving the problems like churn prediction in bank credit card customers and insurance fraud detection, sensitivity is accorded high priority by the experts. Accordingly, by considering sensitivity alone, it is observed that the proposed rule extraction approach using reduced features yielded better sensitivity compared to the hybrids with full features. The number of rules and the size of the rules is small because of the reduced number of features, resulting in the improvement of comprehensibility of the system.

Chapter 7

Rule Extraction from SVR for Solving Regression Problems

This Chapter presents an *eclectic rule extraction approach* proposed for solving regression problems. First section presents the motivation behind the proposed approach. Proposed approach is then presented in detail in second section. Brief description about the datasets used for empirical study is presented in the next section. Fifth section discusses the results and the implications. Final section concludes the chapter.

7.1 Motivation

It is observed that SVM is evolved to solve regression problems as well and is called as SVR (Support Vector Regression). SVR solves regression problems based on the concept of SVM introduced by Vapnik (1995) as described in Appendix A. Over the last decade, different algorithms for extracting rules from SVM for solving classification problems have been developed. However, rule extraction from SVR for solving regression problems is never reported earlier. Further, the efficiency of Classification And Regression Tree (CART), Adaptive Network based Fuzzy Inference System (ANFIS) and Dynamic Evolving Fuzzy Inference System (DENFIS) to deal with regression problems and to produce human comprehensible model was never analyzed in the literature of rule extraction from SVM. In this chapter a hybrid approach for extracting rules from SVR to solve regression problems is presented. CART, ANFIS and DENFIS are employed for generating rules to solve regression problems.

7.2 Proposed Eclectic Rule Extraction Technique

Despite superior performance of SVR for forecasting problems, it develops a black box model. In this chapter we present a novel *eclectic approach* for extracting rules from SVR. The proposed rule extraction procedure is carried out in two phases;

1. *Extraction of support vectors and SVR predictions*. Later, actual target values of the support vectors are replaced by the predictions obtained using the developed SVR model.
2. *Rule generation* using machine leaning techniques.

7.2.1 Extraction of Support Vectors and SVR Predictions

Support vectors are extracted from the given training set as an outcome of SVR training. The predictive accuracy measured in terms of Root Mean Squared Error (RMSE) obtained by SVR is computed and the set of support vectors corresponding to the experiments that yielded the lowest RMSE are considered for the next phase of the hybrid. The dataflow for phase 1 is depicted in Figure 7.1.

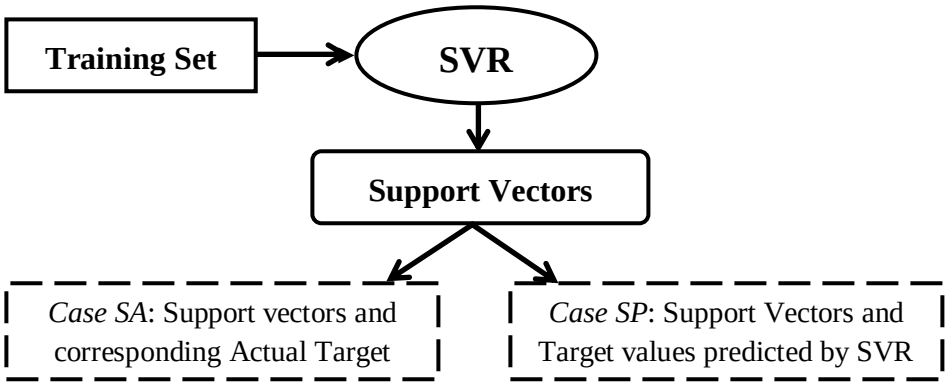


Figure 7.1: Phase 1 of the proposed hybrid
(*Extraction of Support Vectors and SVR Predictions*)

Two different datasets are constructed using the support vectors. Dataset *Case-SA dataset* consists of the support vectors and their corresponding actual target values given in the dataset. Dataset *Case-SP dataset* consists of the support vectors and the corresponding

predicted target values of SVR. Using *Case-SP* dataset it is ensured that the rules generated during phase 2 are indeed extracted from SVR and represents the knowledge learnt by SVR during training.

7.2.2 Rule Generation

It is observed from the literature that researchers depend on decision tree alone for rule generation purpose and for solving classification problems only. During this study techniques *viz.*, CART, ANFIS and DENFIS are explored and employed for rule generation purpose to solve regression problems. Rules are generated using both *Case-SA dataset* and *Case-SP* datasets. Figure 7.2 depicts phase 2 (rule generation) of the proposed hybrid rule extraction method. Prediction accuracy of the rules is determined in terms of Root Mean Squared Error (RMSE), the idea of error is that the lower the RMSE value, higher the prediction rate is.

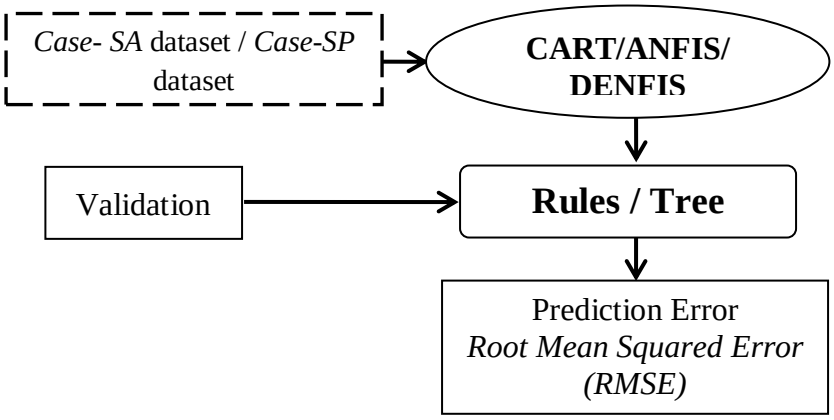


Figure 7.2: Phase 2 of the proposed hybrid (*Rule Generation*)

7.3 Problem Analyzed

We chose publicly available benchmark datasets for regression analysis from UCI machine learning repository and StatLib (Data, Software and News from the Statistics Community) repository. The datasets *viz.*, *Auto MPG*, *Body Fat*, *Boston Housing*, *Forest Fires* and *Pollution* are used to evaluate the proposed hybrid rule extraction procedure for solving regression problems. Table 7.1 presents the feature and instance information about the

datasets analysed in this research study. Table 7.2 presents the division of datasets analysed into training and validation set as explained in Section 1.7 of Chapter 1.

Table 7.1: Dataset Information

Dataset	Total instances	Features	Target Variable
Auto MPG	398	8	Miles Per Gallon
Body Fat	252	15	Body Mass Index
Boston Housing	506	14	MEDV
Forest Fires	517	13	Area effected
Pollution	60	16	Air pollution

Table 7.2: Division of the datasets into Training and Validation

Dataset	Total instances	Training (80%)	Validation (20%)
Auto MPG	398	320	78
Body Fat	252	200	52
Boston Housing	506	500	106
Forest Fires	517	410	107
Pollution	60	50	10

7.4 Results and Discussion

Data mining tool RapidMiner is used for employing SVR algorithm. For rule generation purpose, Salford Systems' CART software is employed. ANFIS is employed in MATLAB and DENFIS is employed in NeuCom software. For developing SVR model all the kernels such as *Linear*, *Polynomial*, *RBF* and *Sigmoid* were employed and SVR yielding the least error is selected for extraction of support vectors. Table 7.3 presents the average RMSE values obtained by SVR. The kernel which yielded best accuracy is then selected to extract support vectors set and is represented in boldface font in the table.

The proposed approach first extracts support vectors and the target values of these support vectors are replaced by the predictions of SVR resulting in *Case-SP* dataset. Support vectors with actual corresponding target values are called *Case-SA* dataset. The efficiency of the proposed hybrid rule extraction procedure i.e. SVR+CART, SVR+ANFIS and

SVR+DENFIS using *Case-SP* dataset is compared with stand-alone CART, ANFIS and DENFIS and the hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS using *Case-SA* dataset. Extensive experiments are conducted on five benchmark datasets *viz.*, *Auto MPG*, *Body Fat*, *Boston Housing*, *Forest Fires* and *Pollution*, to demonstrate the effectiveness of the proposed approach in generating accurate regression rules. Results tables presents the results yielded on validation set only. As CART, ANFIS and DENFIS are also employed in stand-alone fashion their respective RMSE for *Case-SP* dataset are not available.

Table 7.3: Average RMSE values by SVR for UCI benchmark datasets

Dataset	**Linear**		**Polynomial**		**RBF**		**Sigmoid**	
	RMSE	**SVs**	**RMSE**	**SVs**	**RMSE**	**SVs**	**RMSE**	**SVs**
Auto MPG	0.0814	144	**0.0808**	**150**	0.098	141	0.0815	144
Body Fat	**0.0189**	**106**	0.0441	106	0.1124	90	0.019	105
Boston Housing	0.1115	188	**0.0857**	**199**	0.1468	183	0.1115	188
Forest Fires	0.0515	227	0.0518	242	**0.0515**	**228**	0.0515	228
Pollution	0.1187	28	0.1352	33	0.1416	25	**0.1185**	**28**

Table 7.4 presents the average RMSE values for *Auto MPG* dataset using various classifiers. As Polynomial kernel yielded the lowest RMSE of 0.0808 and hence the support vectors extracted using polynomial kernel are used for rule generation purpose. Our proposed hybrid SVR+CART using *Case-SP* dataset yielded highest prediction accuracy against validation set with RMSE of 0.0274. It is observed that our propose hybrid approach perform better than that of SVR algorithm from where the support vectors are extracted and rules are generated. This implies that more generalized rules are extracted. Further, it is also observed that rules extracted using *Case-SP* dataset perform better than their corresponding *Case-SA* datasets. Furthermore, the *Case-SP* dataset represents the knowledge of SVR in the form of predictions. The hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS with *Case-SA* dataset obtained the poorer RMSE values. Table 7.5 presents sample rules extracted using SVR+CART and all the 26 rules are presented in Table D.5 of Appendix D. It is observed from the rules that *weight of the vehicle, horse power* and *year of making* are the most important features and play important role in calculating mile per gallon feature.

Table 7.4: Average RMSE values using Auto MPG dataset

Model	***Case-SA***	***Case-SP***
CART	0.0484	NA
SVR + CART	0.0642	**0.0274**
DENFIS	0.3134	NA
SVR + DENFIS	0.1781	0.1604
ANFIS	0.1607	NA
SVR + ANFIS	0.937	0.1185

Table 7.5: Sample Rules Set using SVR + CART for Auto MPG dataset

Rule #	**Antecedents**	**Consequent**
01	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN <= 0.75 and MODEL_YEAR <= 0.541667	0.473147
02	if WEIGHT <= 0.415226 and ORIGIN <= 0.75 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.875 and HORSEPOWER <= 0.0380435	0.636408
03	if WEIGHT <= 0.415226 and ORIGIN <= 0.75 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.875 and HORSEPOWER > 0.0380435 and HORSEPOWER <= 0.14673	0.529263
04	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN <= 0.75 and MODEL_YEAR > 0.875	0.619481

For *Body fat* dataset Linear kernel yielded the best prediction accuracy with RMSE of 0.0189 (see Table 7.3) and the extracted set of support vectors using linear kernel are then used in the *rule generation phase*. Our proposed hybrid SVR+DENFIS with *Case-SP* dataset yielded highest prediction accuracy on validation data with RMSE of 0.0048. It is observed that SVR+DENFIS yielded the least RMSE compared to SVR itself. This also implies that the more generalised rules are extracted using the proposed hybrid approach. It is also observed that proposed hybrid with *Case-SP* dataset perform better than the hybrids using *Case-SA* datasets. The rules extracted using SVR+DENFIS show GMF(*x*, *y*) i.e. Gaussian Membership Function (*mean*, *variance*) in the antecedent part of the rule with all the features. The prediction function does not change but the *mean* and *variance* values of the features in the antecedent part changes. The rules extracted using the proposed hybrid SVR+DENFIS with *Case-SP* dataset are 46. Sample rule set is presented in Table 7.7 below and full rule set is presented in Table D.6 of Appendix D.

Table 7.6: Average RMSE values using Body Fat dataset

Model	*Case-SA*	*Case-SP*
CART	0.0134	NA
SVR + CART	0.0314	0.0195
DENFIS	0.0311	NA
SVR + DENFIS	0.0201	**0.0048**
ANFIS	0.057	NA
SVR + ANFIS	0.052	0.047

Table 7.7: Sample Rules Set using SVR + CART for Body Fat dataset

Rule #	Antecedents	Prediction
01	if X1 is GMF(0.50,0.34) and X2 is GMF(0.50,0.71) and X3 is GMF(0.50,0.49) and X4 is GMF(0.50,0.73) and X5 is GMF(0.50,0.68) and X6 is GMF(0.50,0.55) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.39) and X9 is GMF(0.50,0.45) and X10 is GMF(0.50,0.44) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.73) and X13 is GMF(0.50,0.67) and X14 is GMF(0.50,0.53)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
02	if X1 is GMF(0.50,0.35) and X2 is GMF(0.50,0.23) and X3 is GMF(0.50,0.53) and X4 is GMF(0.50,0.83) and X5 is GMF(0.50,0.50) and X6 is GMF(0.50,0.46) and X7 is GMF(0.50,0.48) and X8 is GMF(0.50,0.50) and X9 is GMF(0.50,0.63) and X10 is GMF(0.50,0.67) and X11 is GMF(0.50,0.24) and X12 is GMF(0.50,0.75) and X13 is GMF(0.50,0.69) and X14 is GMF(0.50,0.19)	
03	if X1 is GMF(0.50,0.78) and X2 is GMF(0.50,0.55) and X3 is GMF(0.50,0.20) and X4 is GMF(0.50,0.76) and X5 is GMF(0.50,0.17) and X6 is GMF(0.50,0.29) and X7 is GMF(0.50,0.26) and X8 is GMF(0.50,0.20) and X9 is GMF(0.50,0.29) and X10 is GMF(0.50,0.19) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.28) and X13 is GMF(0.50,0.42) and X14 is GMF(0.50,0.14)	

GMF(x, y); Gaussian Membership Function with mean *x* and variance *y*

Table 7.8 presents the average results obtained using *Boston Housing* dataset. As Polynomial kernels yielded lowest RMSE of 0.0857 (See Table 7.3) and hence the support vectors extracted using polynomial kernel are used in *rule generation phase*. The proposed hybrid SVR+CART with *Case-SP* dataset yielded highest prediction accuracy with RMSE of 0.0568. It is observed that SVR+CART using *Case-SP dataset* performed best compared to that of SVR from where support vectors and rules are extracted. It is also observed that the proposed hybrid using *Case-SP dataset* perform better than their

corresponding hybrids using *Case-SA dataset*. Further it is observed that stand-alone CART, ANFIS and DENFIS perform better than the hybrids using *Case-SA dataset*. Using Boston Housing data also it is observed that our proposed hybrid approach yielded better and generalized rules. Table 7.9 presents sample rules extracted using the proposed hybrid SVR+CART using *Case-SP* dataset. Full rule set is presented in Table D.7 in Appendix D.

Table 7.8: Average RMSE values using Boston Housing dataset

Model	***Case-SA***	***Case-SP***
CART	0.0657	NA
SVR + CART	0.0784	**0.0568**
DENFIS	1.1477	NA
SVR + DENFIS	0.4263	0.3101
ANFIS	0.5635	NA
SVR + ANFIS	0.7204	0.1509

Table 7.9: Sample Rules Set using SVR + CART for Boston Housing dataset

Rule #	**Antecedents**	**Prediction**
01	if CRIM <= 0.059321 and LSTAT <= 0.104305 and RM <= 0.620617	0.522726
02	if CRIM <= 0.059321 and LSTAT <= 0.104305 and RM > 0.620617 and RM <= 0.661142	0.580579
03	if CRIM <= 0.059321 and LSTAT > 0.104305 and LSTAT <= 0.164045 and RM <= 0.573864	0.45106
04	if CRIM <= 0.059321 and LSTAT > 0.104305 and LSTAT <= 0.164045 and RM > 0.573864 and RM <= 0.661142	0.505173

Table 7.10 presents the average RMSE values obtained using *Forest Fires* dataset. Using this data RBF kernel yielded the lowest RMSE 0.0515 (See Table 7.3) and the support vectors extracted using RBF kernel are selected for *rule generation phase* of the proposed hybrid. The hybrid SVR + CART with *Case SP* obtained best prediction accuracy with RMSE of 0.00031. The rules extracted using the hybrid SVR+CART with *Case SP* are presented in Table 7.11. It is observed that the hybrid SVR+CART using *Case-SP* dataset yielded the least RMSE compared to that of the stand-alone CART, ANFIS, DENFIS and hybrids using *Case-SA* datasets. The rules extracted using SVR+CART show that *month*

and *rain* are the two most important factors to predict the forest fires. It is also observed once again that the rules generalizes well and proposed approach extract least number of rules resulting in improved comprehensibility of the system.

Table 7.10: Average RMSE using Forest Fires dataset

Model	***Case-SA***	***Case-SP***
CART	0.042	NA
SVR + CART	0.0442	**0.00031**
DENFIS	0.2516	NA
SVR + DENFIS	0.252	0.0099
ANFIS	0.3462	NA
SVR + ANFIS	0.099	0.0098

Table 7.11: Complete set of Rules using SVR+CART for Forest Fires dataset

Rule #	**Antecedents**	**Prediction**
1	if MONTH <= 0.863636 and RAIN <= 0.578125	0.00346333
2	if MONTH <= 0.863636 and RAIN > 0.578125	0.00667844
3	if MONTH > 0.863636	0.00532111

Table 7.12 presents the results obtained using *Pollution* data. As the sigmoid kernel yielded the lowest prediction error using pollution data (See Table 7.3) the support vectors extracted using the sigmoid kernel are used for rule generation. The hybrid SVR+DENFIS with *Case-SP* dataset outperforms other approaches with RMSE value of 0.0765. Table 7.13 presents sample rules and total 26 rules extracted using the proposed hybrid SVR+DENFIS with *Case-SP* dataset are presented in Table D.8 of Appendix D. It is observed from the empirical results that once again the proposed hybrid approach SVR+DENFIS using *Case-SP* dataset yielded best prediction accuracy compared to stand-alone approaches, hybrid using *Case-SA* dataset and SVR as well. The rules extracted rules DENFIS carry GMF(x, y) in antecedents part of the rules which is Gaussian Membership Function with mean x and variance y.

Table 7.12: Average RMSE values using Pollution dataset

Model	***Case-SA***	***Case-SP***
CART	0.1127	NA
SVR + CART	0.1512	0.0854
DENFIS	0.1395	NA
SVR + DENFIS	0.2499	**0.0765**
ANFIS	0.1034	NA
SVR + ANFIS	0.1145	0.0956

Table 7.13: Sample set of Rules using SVR+CART for Pollution dataset

Rule #	**Antecedents**	**Prediction**
01	if X1 is GMF(0.50,0.82) and X2 is GMF(0.50,0.59) and X3 is GMF(0.50,0.69) and X4 is GMF(0.50,0.35) and X5 is GMF(0.50,0.88) and X6 is GMF(0.50,0.39) and X7 is GMF(0.50,0.05) and X8 is GMF(0.50,0.25) and X9 is GMF(0.50,0.95) and X10 is GMF(0.50,0.34) and X11 is GMF(0.50,0.91) and X12 is GMF(0.50,0.09) and X13 is GMF(0.50,0.13) and X14 is GMF(0.50,0.55) and X15 is GMF(0.50,0.31)	Y = 1.30 + 0.38 * X1 - 0.20 * X2 - 0.11 * X3 - 0.10 * X4 - 0.12 * X6 - 0.09 * X7 + 0.29 * X8 + 0.51 * X9 - 0.05 * X10 - 0.03 * X11 - 0.03 * X12 + 0.02 * X13 + 0.15 * X14 + 0.01 * X15
02	if X1 is GMF(0.50,0.64) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.37) and X4 is GMF(0.50,0.82) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.93) and X7 is GMF(0.50,0.70) and X8 is GMF(0.50,0.40) and X9 is GMF(0.50,0.11) and X10 is GMF(0.50,0.77) and X11 is GMF(0.50,0.13) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.13) and X14 is GMF(0.50,0.48) and X15 is GMF(0.50,0.39)	
03	if X1 is GMF(0.50,0.68)and X2 is GMF(0.50,0.24) and X3 is GMF(0.50,0.15) and X4 is GMF(0.50,0.95) and X5 is GMF(0.50,0.53) and X6 is GMF(0.50,0.64) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.29) and X9 is GMF(0.50,0.05) and X10 is GMF(0.50,0.46) and X11 is GMF(0.50,0.28) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.09) and X15 is GMF(0.50,0.39)	

GMF(x, y); Gaussian Membership Function with mean *x* and variance *y*

It is observed from the experiments that the proposed hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS with *Case-SP* dataset obtained better prediction accuracies, compared to other hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS with *Case-SA* dataset and stand-alone CART, ANFIS and DENFIS. It is also observed from the experiments that the hybrid SVR+CART with *Case-SP* dataset obtained the best prediction accuracy in case of

Auto MPG, *Boston Housing* and *Forest Fires* datasets and SVR+DENFIS with *Case-SP* dataset obtained best results for *Body Fat* and *Pollution* datasets. Stand-alone CART, ANFIS and DENFIS performed better compared to the hybrid SVR+CART, SVR+ANFIS and SVR+DENFIS with *Case-SA* dataset, because stand-alone approaches use all the training samples to learn where as hybrids using Case-SA dataset utilize only support vector set with corresponding actual target values. Further, it is observed that more generalized rules are generated using the proposed approach i.e. Case-SP dataset and rules extracted using Case-SP dataset yielded less error than that of SVR using which support vectors are extracted.

7.5 Conclusions

In this chapter, we propose a new hybrid rule extraction approach for solving regression problems using SVR. The proposed approach involves two major phases. During first phase, support vectors are extracted from SVR and the predictions are obtained for those support vectors using trained SVR. The dataset of extracted support vectors with actual corresponding actual target values available is referred to as *Case-SA* dataset, whereas in *Case-SP* dataset the corresponding target values are replaced by the predictions given by SVR. Later these modified support vectors are fed to one of CART, ANFIS and DENFIS separately in second phase i.e. rule generation phase. Using *Case-SP* dataset we are ensuring that the rules generated during second phase are indeed extracted from SVR. From the analysis it is concluded that the proposed hybrid SVR+CART, SVR+ANFIS and SVR+DENFIS with *Case-SP* dataset achieved higher accuracy than the accuracies obtained using stand-alone CART, ANFIS, DENFIS, hybrids SVR+CART, SVR+ANIFS and SVR+DENFIS with *Case-SA* dataset. Stand-alone CART, ANFIS and DENFIS achieved better accuracy compared to the hybrids SVR+CART, SVR+ANFIS and SVR+DENFIS using *Case-SA* dataset. It is concluded that, using the proposed hybrid approach more generalized rules are extracted and these rules perform better than that of SVR. We conclude that the proposed hybrid is a viable alternative to generating rules from SVR for solving regression problems.

Chapter 8

Overall Conclusions and Future Directions

Although Support Vector Machines have been used to develop highly accurate classification and regression models in various real-world problem domains, the most significant barrier is that, they generate models that are *difficult to understand*. The procedure to convert these opaque models into transparent models is called *rule extraction*. This thesis investigates the task of extracting comprehensible models from trained SVMs for solving classification and regression problems, thereby alleviating this limitation.

This thesis investigates various ways to extract the knowledge learnt by trained SVM. In the research work presented in this thesis, we proposed various rule extraction approaches to represent the knowledge learnt by SVM and we solved problems in banking and finance using the rules extracted. There are two dimensions to the rule extraction approach (i) rule extraction using SVM and (ii) rule extraction from SVM. During rule extraction using SVM, SVM is used as a pre-processor only, where only support vectors are extracted resulting in *Case-SA* dataset. During rule extraction from SVM, the trained SVM is used for prediction purpose as well, resulting in two variants of the datasets i.e. *Case-P* and *Case-SP*, where *Case-P* represents the training data with SVM predictions and *Case-SP* represents the support vector set with SVM predictions, respectively. However, feature selection using SVM-RFE algorithm spans both the dimensions. The thesis addresses both the dimensions equally. The modified data is the replica of the knowledge learnt by SVM during training. Later, modifications to active learning based approach are proposed for solving data mining problems in finance.

This thesis also investigates the efficiency of our proposed rule extraction approach in solving the problem of Bankruptcy Prediction in Banks. Bankruptcy prediction in banks and corporate firms is the most researched area in the field of finance. Bank management would be interested in the comprehensibility of the algorithms used for predictions. We

extracted fuzzy rules for bankruptcy prediction problems using fuzzy rule based systems and the efficiency of the fuzzy rules is then compared with the rules extracted using Decision Tree. It is observed from the results that fuzzy rules perform better than decision tree rules and are more comprehensible in nature.

In this thesis an eclectic approach is proposed to extract rules from SVR for solving regression problems. No rule extraction approach is proposed for solving regression problems using SVR in earlier research work in this field. We proposed a rule extraction from SVR approach for solving regression problem for the first time in literature. We employed CART, ANFIS and DENFIS for generating rules for regression problems. It is observed that rules extracted using our proposed approach to solve regression problems increases the generalization ability of the SVR and produces accurate comprehensible models.

In real world data mining applications the datasets are highly unbalanced. It is observed from the literature that standard machine learning techniques tend to learn better about majority class, thus producing poor prediction accuracy over the minority class and usually the objective of the study is predicting minority class. We proposed a rule extraction approach to extract rules for solving such unbalanced and medium scale problems. Churn prediction in credit card customers' problems is solved during this study and the dataset analyzed during this study is unbalanced in nature and medium scale in size. Various balancing techniques are employed to bring the balance in dataset before extracting rules. NBTree algorithm is employed to extract rules, which is capable of learning better and faster for large datasets. It is observed that our proposed rule extraction approach with SMOTE data produced the most accurate results.

Later, modifications to Active Learning Based Approach (Martens et al., 2009) are proposed, where extra instances are generated near decision boundary of SVM using various distributions such as Normal and Logistic. Data mining problems such as Churn prediction in bank credit card customers' and fraud detection in Insurance are solved using mALBA. NBTree algorithm is employed for extracting rules. During this study, we highlighted the importance of support vectors with newly generated data. Despite using

unbalanced datasets for analysis, it is observed that our proposed approach yielded better results than that of the best results reported in literature for these problems. It is also observed that, when we used artificially generated instances with support vectors instead of training set, the complexity of the rule extraction procedure is reduced and the extracted rules are also less in number resulting in high comprehensibility.

Overall, in this thesis various approaches to extract the knowledge of SVM in the form of *if-then* rules are presented in all the categories of the framework of decompositional, pedagogical and eclectic methods of rule extraction. We have analyzed the efficiency of various rule generating algorithms for extracting rules such as; FRBS, CART, DT, ANFIS, DENFIS, NBTree. As the problems solved are from banking and finance domain, management would be interested to have transparent model to understand how a customer is behaving etc. or the financial health of a bank. Rules extracted can later be used as an early warning system by the management of the bank and they can take right decisions at right times to avoid heavy losses to the organization.

Future Directions

A very important topic in this area requiring further investigation is how to extend rule extraction methods to the case of SVM incremental and active learning, and how to embed a rule extraction method into an online data mining applications. In these cases, more attention should be paid to the time complexity of the proposed algorithms.

Performing feature selection using SVM-RFE prior to extract fuzzy rules from SVM could be another direction for future work.

Time series problems can also be addressed with respect to rule extraction from SVM.

Appendix A:

Overview of SVM/SVR and SVM-RFE

This section provides the detailed overview of the machine learning techniques used for rule generation purpose. The techniques discussed in this section are Support Vector Classification, Support Vector Regression and SVM-Recursive Feature Elimination.

A.1 Support Vector Machine

The theoretical foundation of support vector machine (SVM) is given by statistical learning theory (Vapnik 1995). SVM was largely developed at AT&T Bell Labs by Vapnik and co-workers (Boser et al. 1992; Guyon et al. 1993; Cortes and Vapnik 1995; Scholkopf et al. 1995; Scholkopf et al. 1996; Vapnik et al. 1997). Due to this industrial context, SVM research has up-to-date had a sound orientation towards real world applications. A comprehensive tutorial for support vector classifiers has been published by Burges (1998). Support vector regression (SVR) has shown superior performance in many real time applications (Muller et al. 1997; Ducker et al. 1997; Stitson et al. 1999; Mattera and haykin 1999; Tay and Cao 2002; Eads et al. 2002; Cao and Tay 2003; Chen and Ho 2005; Lin et al. 2006). SVM transforms non-linear problems into ones in multidimensional space using kernel functions and tries to find linear separating hyperplane margin. The problem of empirical data modelling is germane to many engineering applications. In empirical data modelling a process of induction is used to build up a model, from which it is hoped to deduce responses of the system that have yet to be observed. Ultimately the quantity and quality of the observations govern the performance of the empirical model. By its observational nature data obtained is finite and sampled, typically this sampling is non-uniform and due to the high dimensional nature of the problem the data will form only a sparse distribution in the input space. Consequently the problem is nearly always ill posed (Poggio et al., 1985) in the sense of Hadamard (1923). Traditional neural network

approaches have suffered difficulties with generalization, producing models that can overfit the data. This is a consequence of the optimisation algorithms used for parameter selection and the statistical measures used to select the "best" model. The foundation of Support Vector Machines (SVM) have been developed by Vapnik (1995) and are gaining popularity due to many attractive features, and promising empirical performance. The formulation embodies the Structural Risk Minimisation (SRM) principle, which has been shown to be superior (Gunn et al., 1997), to the traditional Empirical Risk minimisation (ERM) principle that is employed by conventional neural networks. SRM minimises an upper bound on the expected risk, as opposed to ERM that minimises the error on the training data. It is this difference which equips SVM with a greater ability to generalise, which is the goal in statistical learning. SVMs were developed to solve the classification problem, but recently they have been extended to the domain of regression problems (Vapnik et al., 1997).

A.1.1 VC Dimension

The VC dimension is a scalar value that measures the capacity of a set of functions. The VC dimension of a set of functions is p if and only if there exists a set of points $\{x_i\}_{i=1}^{p}$ such that these points can be separated in all $2p$ possible configurations, and that no set $\{x_i\}_{i=1}^{q}$ exists where $q > p$ satisfying this property. Figure A.1 illustrates how three points in the plane can be shattered by the set of linear indicator functions whereas four points cannot. In this case the VC dimension is equal to the number of free parameters, but in general that is not the case; e.g. the function $A\sin(b_x)$ has an infinite VC dimension (Vapnik, 1995). The set of linear indicator functions in n dimensional space has a VC dimension equal to $n+1$.

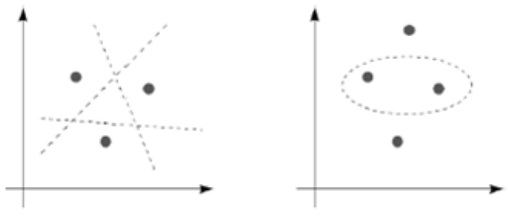


Figure A.1: VC Dimension Illustration

A.1.2 Structural Risk Minimisation

Create a structure such that S_h is a hypothesis space of VC dimension h then,

$$S_1 \subset S_2 \subset ... \subset S_\infty \qquad \text{a.1}$$

SRM consists in solving the following problem

$$\min_{S_h} R_{emp}[f] + \sqrt{\frac{h \ln\left(\frac{2l}{h}+1\right) - \ln\left(\frac{\delta}{4}\right)}{l}} \qquad \text{a.2}$$

If the underlying process being modelled is not deterministic the modelling problem becomes more exacting. Multiple output problems can usually be reduced to a set of single output problems that may be considered independent. Hence it is appropriate to consider processes with multiple inputs from which it is desired to predict a single output.

A.1.3 Support Vector Classification

The goal is to produce a classifier that will work well on unseen examples, i.e. it generalises well. Consider the example in Figure A.2. Here there are many possible linear classifiers that can separate the data, but there is only one that maximises the margin. This linear classifier is termed the optimal separating hyperplane.

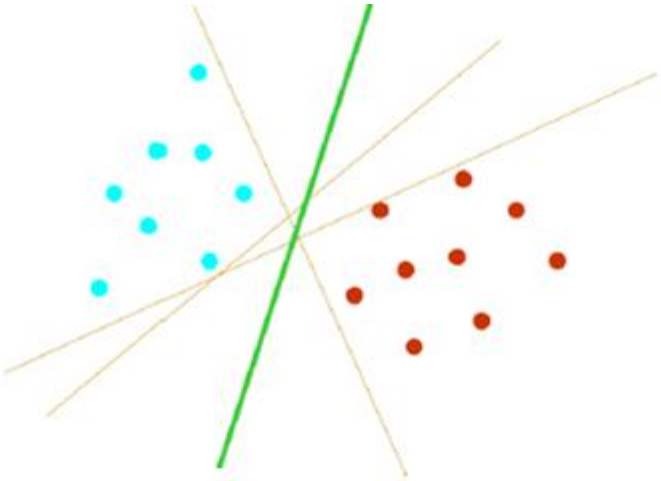


Figure A.2: Optimal Separating Hyperplane

A.1.4 Linearly Separable Case

Consider the problem of separating the set of training vectors belonging to two separate classes,

$$D=\{(x_1,y_1),...,(x_l,y_l)\},\quad x\in\Re_n, y\in\{-1,1\}, \qquad \text{a.3}$$

with a hyperplane,

$$\langle w,x\rangle+b=0. \qquad \text{a.4}$$

The set of vectors is said to be optimally separated by the hyperplane if it is separated without error and the distance between the closest vector to the hyperplane is maximal. There is some redundancy in Equation a.4, and without loss of generality it is appropriate to consider a canonical hyperplane (Vapnik, 1995), where the parameters *w*, *b* are constrained by,

$$\min_i \left|\langle w,x_i\rangle+b\right|=1. \qquad \text{a.5}$$

A separating hyperplane in canonical form must satisfy the following constraints,

$$y_i\left[\langle w,x_i\rangle+b\right]\geq 1,\quad i=1,...,l. \qquad \text{a.6}$$

The distance *d(w, b; x)* of a point *x* from the hyperplane (*w*, *b*) is,

$$d(w,b;x)=\frac{\left|\langle w,x_i\rangle+b\right|}{\|w\|}. \qquad \text{a.7}$$

The optimal hyperplane is given by maximising the margin, *p*, subject to the constraints of Equation a.6. The margin is given by, as shown in Figure A.3 below and the learning of the optimal separating hyperplane is shown in Figure A.4.

$$\rho(w,b) = \min_{x_i;\, y_i=-1} d(w,b;x_i) + \min_{x_i;\, y_i=1} d(w,b;x_i)$$
$$= \min_{x_i;\, y_i=-1} \frac{|\langle w, x_i\rangle + b|}{\|w\|} + \min_{x_i;\, y_i=1} \frac{|\langle w, x_i\rangle + b|}{\|w\|}$$
$$= \frac{1}{\|w\|}\left(\min_{x_i;\, y_i=-1} |\langle w, x_i\rangle + b| + \min_{x_i;\, y_i=1} |\langle w, x_i\rangle + b| \right)$$
$$= \frac{2}{\|w\|} \qquad \text{a.8}$$

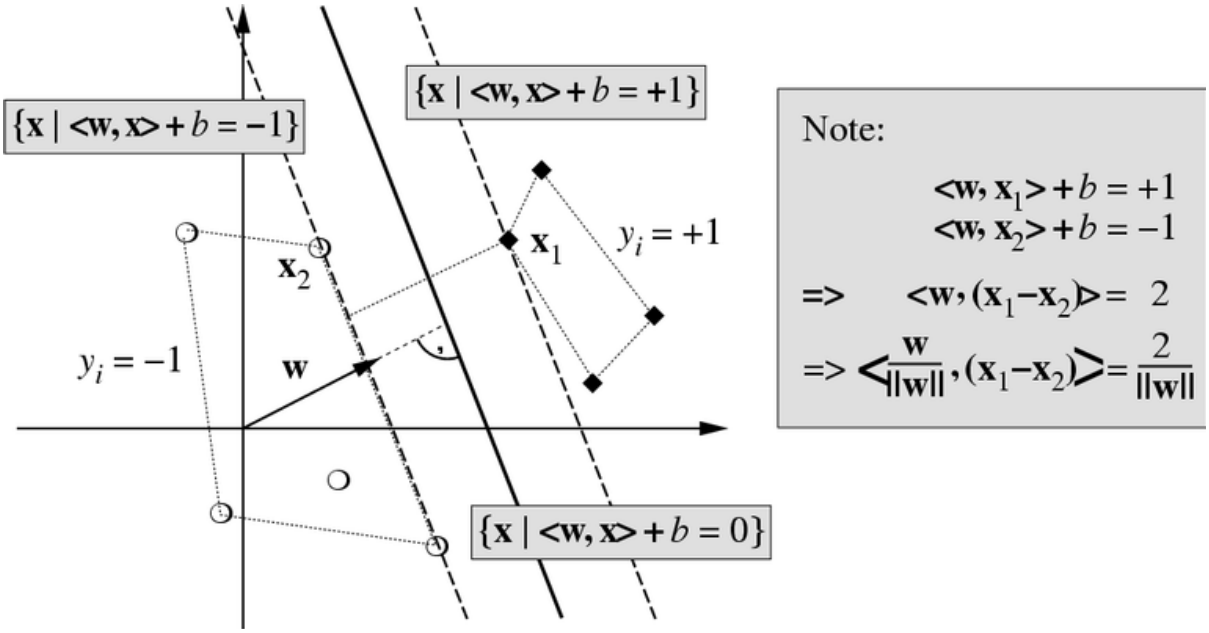


Figure A.3: Margin for the Hyperplane.

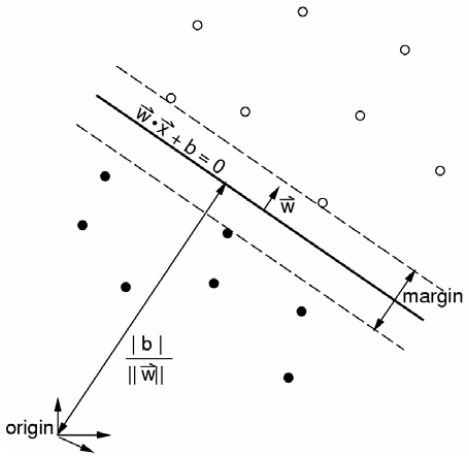


Figure A.4: How do SVM choose the margin?

Hence the hyperplane that optimally separates the data is the one that minimises

$$\Phi(w) = \frac{1}{2}\|w\|^2 \qquad \text{a.9}$$

The VC dimension, h, of the set of canonical hyperplanes in n dimensional space is bounded by,

$$h \le \min\left[R^2 A^2, n\right] + 1 \qquad \text{a.10}$$

where *R* is the radius of a hypersphere enclosing all the data points.

The solution to the optimisation problem of equation a.9 under the constraints of equation a.6 is given by the saddle point of the Lagrange Functional (Lagrangian) (Minoux, 1986),

$$\Phi(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i \left(y_i \left[\langle w, x_i \rangle + b\right] - 1\right), \qquad \text{a.11}$$

where alpha are the Lagrange multipliers. The Lagrangian has to be minimised with respect to w, b and maximised with respect to alpha>=0. Classical Lagrangian duality enables the primal problem, equation a.11, to be transformed to its dual problem, which is easier to solve. The dual problem is given by,

$$\max_{\alpha} W(\alpha) = \max_{\alpha}\left(\min_{w,b} \Phi(w, b, \alpha)\right) \qquad \text{a.12}$$

The minimum with respect to w and b of the Lagrangian, Φ, is given by,

$$\begin{aligned} \frac{\partial \Phi}{\partial b} = 0 &\Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0 \\ \frac{\partial \Phi}{\partial w} = 0 &\Rightarrow w = \sum_{i=1}^{l} \alpha_i y_i x_i \end{aligned} \qquad \text{a.13}$$

Hence from equation a.11, a.12 and a.13, the dual problem is,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\frac{1}{2}\sum_{i-1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \left\langle x_i, x_j \right\rangle + \sum_{k=1}^{l} \alpha_k, \qquad \text{a.14}$$

and hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l}\alpha_k \qquad \text{a.15}$$

with constraints

$$\alpha_i \geq 0 \quad i = 1,...,l$$
$$\sum_{j=1}^{l}\alpha_j y_j = 0 \qquad \text{a.16}$$

Solving Equation a.15 with constraints Equation a.16 determines the Lagrange multipliers, and the optimal separating hyperplane is given by,

$$w^* = \sum_{i=1}^{l}\alpha_i y_i x_i$$
$$b^* = -\frac{1}{2}\langle w^*, x_r + x_s \rangle. \qquad \text{a.17}$$

where x_r and x_s are any support vector from each class satisfying,

$$\alpha_r, \alpha_s > 0, \qquad y_r = -1, \quad y_s = 1 \qquad \text{a.18}$$

The hard classifier is then,

$$f(x) = \text{sgn}(\langle w^*, x \rangle + b) \qquad \text{a.19}$$

Alternatively, a soft classifier may be used which linearly interpolates the margin,

$$f(x) = h(\langle w^*, x \rangle + b) \quad \textit{where } h(z) = \begin{cases} -1 & : \ z < -1 \\ z & : \ -1 \leq z \leq 1 \\ +1 & : \ z > 1 \end{cases} \qquad \text{a.20}$$

This may be more appropriate than the hard classifier of Equation a.19, because it produces a real valued output between -1 and 1 when the classifier is queried within the margin, where no training data resides. From the Kuhn-Tucker conditions,

$$\alpha_i \left(y_i \left[\langle w, x_i \rangle + b \right] - 1 \right) = 0, \quad i = 1, \ldots, l, \qquad \text{a.21}$$

and hence only points x_i which satisfy,

$$y_i \left[\langle w, x_i \rangle + b \right] = 1 \qquad \text{a.22}$$

will have non-zero Lagrangian multipliers. These points are termed support vectors (SVs). If the data is linearly separable all the SVs will lie on the margin and hence the number of SVs can be very small. Consequently the hyperplane is determined by a small subset of the training set. The other points could be removed from the training set and recalculating the hyperplane would produce the same answer. Hence, SVM can be used to summarise the information contained in a dataset by the SV produced. If the data is linearly separable the following equality will hold.

$$\|w\|^2 = \sum_{i=1}^{l} \alpha_i = \sum_{i \in SV_s} \alpha_i = \sum_{i \in SV_s} \sum_{j \in SV_s} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle. \qquad \text{a.23}$$

Hence from Equation a.10 the VC dimension of the classifier is bounded by,

$$h \leq \min \left[R^2 \sum_{i \in SV_s}, n \right] + 1, \qquad \text{a.24}$$

and if the training data X is normalised to lie in the unit hypersphere.

$$h \leq 1 + \min \left[\sum_{i \in SV_s}, n \right], \qquad \text{a.25}$$

A.1.5 Linearly Non-Separable Case

There are two approaches to generalising the problem, which are dependent upon prior knowledge of the problem and an estimate of the noise on the data. In the case where it is expected (or possibly even known) that a hyperplane can correctly separate the data, a method of introducing an additional cost function associated with misclassification is appropriate. Alternatively a more complex function can be used to describe the boundary.

To enable the optimal separating hyperplane method to be generalised, Cortes and Vapnik (1995) introduced non-negative variables, $\xi \geq 0$, and a penalty function,

$$F_\sigma(\xi) = \sum_i \xi_i^\sigma \qquad \sigma > 0. \qquad \text{a.26}$$

where the ξ_i are a measure of the misclassification errors. The optimisation problem is now posed so as to minimise the classification error as well as minimising the bound on the VC dimension of the classifier. The constraints of Equation a.6 are modified for the non-separable case to,

$$y_i\left[\langle w, x_i \rangle + b\right] \geq 1 - \xi_i, \quad i = 1, \ldots, l. \qquad \text{a.27}$$

where $\xi_i \geq 0$. The generalised optimal separating hyperplane is determined by the vector w, that minimises the functional,

$$\Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i, \qquad \text{a.28}$$

(where C is a given value) subject to the constraints of Equation a.27. the solution to the optimisation problem of Equation a.28 under the constraints of Equation a.27 is given by saddle point of the Lagrangian (Minoux, 1986),

$$\Phi(w, b, \alpha, \xi, \beta) = \frac{1}{2}\|w\|^2 + C\sum_i \xi_i - \sum_{i=1}^{l} \alpha_i \left(y_i\left[w^T x_i + b\right] - 1 + \xi_i\right) - \sum_{j=1}^{l} \beta_i \xi_i \qquad \text{a.29}$$

where alpha, beta are the Lagrange multipliers. The Lagrangian has to be minimised with respect to *w, b, x* and maximised with respect to alpha and beta. The dual problem is then given by,

$$\max_\alpha W(\alpha, \beta) = \max_{\alpha, \beta}\left(\min_{w, b, \xi} \Phi(w, b, \alpha, \xi, \beta)\right) \qquad \text{a.30}$$

The minimum with respect to *w, b* and ξ of the Lagrangian, Φ, is given by,

$$\frac{\partial \Phi}{\partial b} = 0 \Rightarrow \sum_{i=1}^{l} \alpha_i y_i = 0$$
$$\frac{\partial \Phi}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{l} \alpha_i y_i x_i \qquad \text{a.31}$$
$$\frac{\partial \Phi}{\partial \xi} = 0 \Rightarrow \alpha_i + \beta_i = C$$

Hence from Equation a.29, a.30 and a.31 the dual problem is,

$$\max_{\alpha} W(\alpha) = \max_{\alpha} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l} \alpha_k, \qquad \text{a.32}$$

and hence the solution to the problem is given by,

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle + \sum_{k=1}^{l} \alpha_k, \qquad \text{a.33}$$

with constraints,

$$0 \le \alpha_i \le C \quad i = 1, \ldots, l$$
$$\sum_{j=1}^{l} \alpha_j y_j = 0. \qquad \text{a.34}$$

C can be directly related to a regularisation parameter (Girosi 1997; Smola and Scholkopf, 1998). Blanz et al., (1996) uses a value of C = 5, but ultimately C must be chosen to reflect the knowledge of the noise on the data.

A.1.6 Feature Space

In the case where a linear boundary is inappropriate the SVM can map the input vector X into a high dimensional feature space *z*. By choosing a non-linear mapping a priori, the SVM constructs an optimal separating hyperplane in this higher dimensional space as shown in Figure A.5 and Figure A.6. The idea exploits the method of Aizerman et al., (1964) which, enables the curse of dimensionality (Bellman, 1961) to be addressed.

Figure A.5: Mapping Low Dimension Input Space to High Dimensional Feature Space

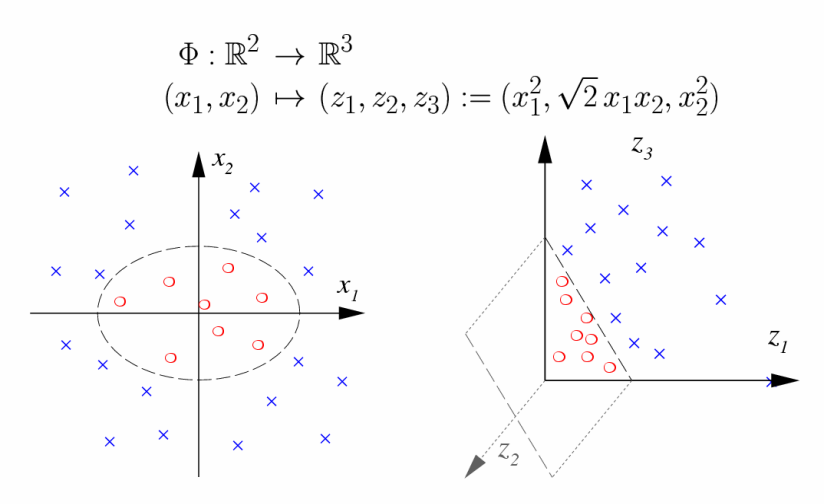


Figure A.6: Mapping into Non-Linear Feature Space

There are some restrictions on the non-linear mapping that can be employed, but it turns out, surprisingly that most commonly employed functions are acceptable. Among acceptable mappings are polynomials, radial basis functions and certain sigmoid functions. The optimisation problem of Equation a.33 becomes,

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j \left(x_i, x_j\right) - \sum_{k=1}^{l}\alpha_k, \qquad \text{a.35}$$

where *K(x, x')* is the kernel function performing the non-linear mapping into feature space, and the constraints are unchanged,

$$0 \le \alpha_i \le C \quad i = 1, \ldots, l$$
$$\sum_{j=1}^{l} \alpha_j y_j = 0. \qquad \text{a.36}$$

A hard classifier implementing the optimal separating hyperplane in the feature space is given by,

$$f(x) = \text{sgn}\left(\sum_{i \in SV_s} \alpha_i y_i K(x_i, x) + b \right) \qquad \text{a.37}$$

where

$$\langle w^*, x \rangle = \sum_{i=1}^{l} \alpha_i y_i K(x_i, x)$$
$$b^* = -\frac{1}{2} \sum_{i=1}^{l} \alpha_i y_i \left[K(x_i, x_r) + K(x_i, x_r) \right]. \qquad \text{a.38}$$

The bias is computed here using two support vectors, but can be computed using all the SVs on the margin for stability (Vapnik et al., 1997). If the Kernel contains a bias term, the bias can be accommodated within the Kernel, and hence the classifier is simply,

$$f(x) = \text{sgn}\left(\sum_{i \in SV_s} \alpha_i K(x_i, x) \right) \qquad \text{a.39}$$

Many employed kernels have a bias term and any finite Kernel can be made to have one (Girosi, 1997). This simplifies the optimisation problem by removing the equality constraint of Equation a.36.

The idea of the kernel function is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Hence, the inner product does not need to be evaluated in the feature space. This provides a way of addressing the curse of dimensionality. However, the computation is still critically dependent upon the number of training patterns and to provide a good data distribution for a high dimensional problem will generally require a large training set.

A.1.7 Kernel Functions

The following theory is based upon Reproducing Kernel Hilbert Spaces (RKHS) (Aronszajn, 1950; Girosi, 1997; Heckman, 1997; Wahba, 1990). An inner product in feature space has an equivalent kernel in input space,

$$K(x, x') = \langle \phi(x), \phi(x') \rangle, \qquad \text{a.40}$$

provided certain conditions hold. If K is a symmetric positive definite function, which satisfies Mercer's Conditions,

$$K(x, x') = \sum_{m}^{\infty} \alpha_m \phi_m(x) \phi_m(x'), \qquad \alpha_m \geq 0, \qquad \text{a.41}$$

$$\int\int K(x, x') g(x) g(x') dx dx' > 0, \qquad g \in L_2, \qquad \text{a.42}$$

then the kernel represents a legitimate inner product in feature space. Valid functions that satisfy Mercer's conditions are now given, which unless stated are valid for all real *x* and *x'*.

Polynomial

A polynomial mapping is a popular method for non-linear modelling,

$$K(x, x') = \langle x, x' \rangle^d. \qquad \text{a.43}$$

$$K(x, x') = (\langle x, x' \rangle + 1)^d. \qquad \text{a.44}$$

The second kernel is usually preferable as it avoids problems with the hessian becoming zero.

Radial Basis Function

Radial basis functions have received significant attention, most commonly with a Gaussian of the form,

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right). \qquad \text{a.45}$$

produces a piecewise linear solution which can be attractive when discontinuities are acceptable.

Multi-Layer Perceptron

The long established MLP, with a single hidden layer, also has a valid kernel representation,

$$K(x, x') = \tanh\left(\rho\langle x, x'\rangle + g\right) \qquad \text{a.46}$$

for certain values of the scale, ρ, and offset, *g*, parameters. Here the SV correspond to the first layer and the Lagrange multipliers to the weights.

However, this kernel is probably not a good choice because its regularisation capability is poor, which is evident by consideration of its Fourier transform (Smola and Scholkopf, 1997).

A.1.8 Kernel Selection

It is very much difficult to select or suggest a best mapping for a particular problem? But with the inclusion of many mappings within one framework it is easier to make a comparison. As a final caution, even if a strong theoretical method for selecting a kernel is developed, unless this can be validated using independent test sets on a large number of problems, methods such as bootstrapping and cross-validation will remain the preferred method for kernel selection.

Support Vector Machines are an attractive approach to data modelling. They combine generalisation control with a technique to address the curse of dimensionality. the formulation results in a global quadratic optimisation problem with box constraints, which is readily solved by interior point methods. The kernel mapping provides a unifying framework for most of the commonly employed model architectures, enabling comparisons to be formed. In classification problems generalisation control is obtained by

maximising the margin, which corresponds to minimisation of the weight vector in a canonical framework. The solution obtained as a set of support vectors that can be sparse. These lie on the boundary and as such summarise the information required to separate the data. the minimisation of the weight vector can be used as a criterion in regression problems, with a modified loss function.

Usually more than one kernel used in the literature to map the input space into feature space (Cristianini and Shawe-Taylor, 2000). The question is which kernel functions provide good generalization for a particular problem. One has to use more than one kernel function for a particular problem in order to resolve this issue. Because of the approximate mapping of input space to higher dimensional feature space using different kernel functions, support vectors extracted are different and the number of support vectors varies as well for each kernel.

A.2 Support Vector Regression

SVMs can also be applied to regression problems by the introduction of an alternative loss function (Smola, 1996). The loss function must be modified to include a distance measure. Various forms of loss functions are shown in Figure A.7. Figure A.8 shows the curve obtained using soft margin using a linear SVR.

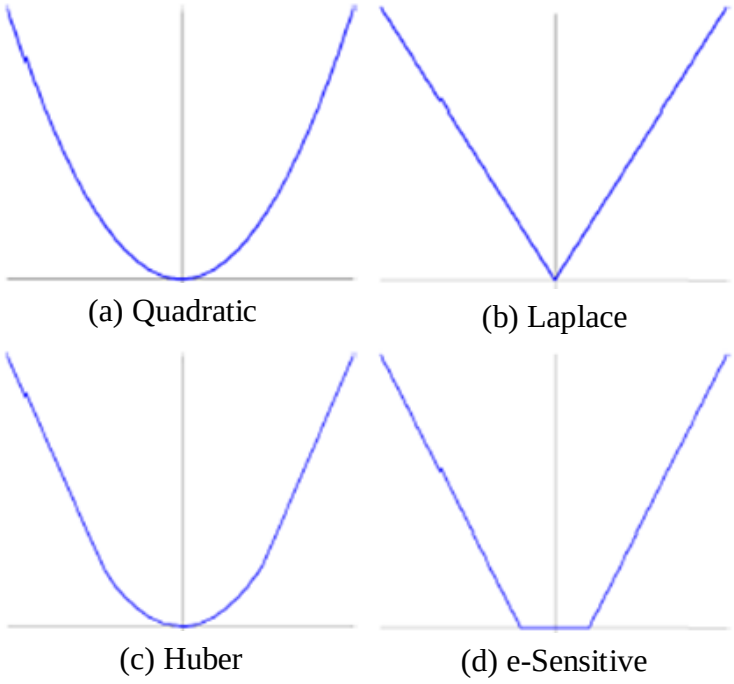


Figure A.7: (a, b and c) produce no sparseness in the support vectors, to address this issue Vapnik proposed the loss function in Figure A.7 (d) as an approximation to Huber's loss function that enables a sparse set of support vectors to be obtained.

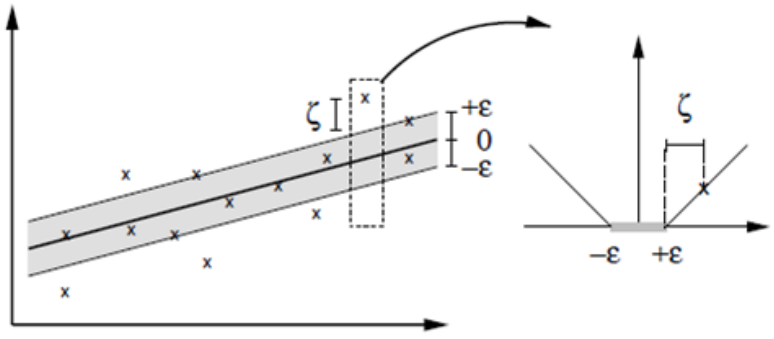


Figure A.8: The soft margin loss setting of Linear SVR.

A.2.1 Linear Regression

Consider the problem of approximating the set of data,

$$D = \{(x_1, y_1), \ldots, (x_l, y_l)\}, \quad x \in \Re^n, y \in \Re, \qquad \text{a.47}$$

with a linear function,

$$f(x) = \langle w, x \rangle + b \qquad \text{a.48}$$

The optimal regression function is given by the minimum of the functional,

$$\Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_i (\xi_i^- + \xi_i^+), \qquad \text{a.49}$$

where C is pre-specified value, and ξ_i^- and ξ_i^+ are slack variables representing upper and lower constraints on the outputs of the system.

ε-insensitive Loss Function

using an e-insensitive loss function, Figure 5.1(d),

$$L_\varepsilon(y) = \begin{cases} 0 & for \ |f(x) - y| < \varepsilon \\ |f(x) - y| - \varepsilon & otherwise \end{cases} \qquad \text{a.50}$$

the solution is given by,

$$\max_{\alpha,\alpha^*} W(\alpha, \alpha^*) = \max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)\langle x_i, x_j \rangle + \sum_{i=1}^{l}\alpha_i(y_i - \varepsilon) - \alpha_i^*(y_i + \varepsilon) \qquad \text{a.51}$$

or alternatively,

$$\alpha, \alpha^* = \arg\max_{\alpha,\alpha^*} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j \rangle - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)y_i + \sum_{i=1}^{l}(\alpha_i + \alpha_i^*)\varepsilon$$

a.52

with constraints,

$$0 \le \alpha_i,\ \alpha_i^* \le C, \quad i = 1, \ldots, l$$
$$\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0. \qquad \text{a.53}$$

Solving Equation a.51 with constraints Equation a.53 determines the Lagrange multipliers, $\alpha,\ \alpha*$, and the regression function is given by Equation a.48, where

$$\bar{w} = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i$$
$$\bar{b} = -\frac{1}{2} \langle \bar{w}, (x_r + x_s) \rangle. \qquad \text{a.54}$$

The Karush-Kuhn-Tucker (KKT) conditions that are satisfied by the solution are,

$$\bar{\alpha}_i \bar{\alpha}_i^* = 0, \qquad i = 1, \ldots, l. \qquad \text{a.55}$$

Therefore the support vectors are points where exactly one of the Lagrange multipliers is greater than zero. When epsilon=0, we get the L1 loss function and the optimisation problem is simplified,

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \beta_i \beta_j \langle x_i, x_j \rangle - \sum_{i=1}^{l} \beta_i y_i \qquad \text{a.56}$$

with constraints,

$$-C \le \beta_i \le C, \quad i = 1, \ldots, l$$
$$\sum_{i=1}^{l} \beta_i = 0, \qquad \text{a.57}$$

and the regression function is given by Equation a.48, where

$$\bar{w} = \sum_{i=1}^{l} \beta_i x_i$$
$$\bar{b} = -\frac{1}{2} \langle \bar{w}, (x_r + x_s) \rangle. \qquad \text{a.58}$$

A.2.2 Non-Linear Regression

Similarly to classification problems, a non-linear model is usually required to adequately model data. In the same manner as the non-linear SVC approach, a non-linear mapping can be used to map the data into a high dimensional feature space where linear regression is performed. The kernel approach is again employed to address the curse of dimensionality. The non-linear SVR solution, using an e-insensitive loss function, is given by,

$$\max_{\alpha,\alpha^*} W(\alpha,\alpha*) = \max_{\alpha,\alpha^*} \sum_{i=1}^{l} \alpha_i^* (y_i - \varepsilon) - \alpha_i (y_i + \varepsilon) - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) K(x_i, x_j)$$

a.59

with constraints,

$$0 \le \alpha_i,\ \alpha_i^* \le C, \quad i = 1, \ldots, l$$
$$\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0. \qquad \text{a.60}$$

This determines the Lagrange multipliers, α_i, α_i^* and regression function is given by,

$$f(x) = \sum_{SV_s} (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) + \bar{b} \qquad \text{a.61}$$

where

$$\langle \bar{w}, x \rangle = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) K(x_i, x_j)$$
$$\bar{b} = -\frac{1}{2} \sum_{i=1}^{l} (\alpha_i - \alpha_i^*)(K(x_i, x_r) + K(x_i, x_s)). \qquad \text{a.62}$$

As with the SVC the equality constraint may be dropped if the kernel constraints a bias term, *b* being accommodated within the Kernel function, and the regression function is given by,

$$f(x) = \sum_{i=1}^{l} (\bar{\alpha}_i - \bar{\alpha}_i^*) K(x_i, x) \qquad \text{a.63}$$

The optimisation criteria for the other loss functions are similarly obtained by replacing the dot product with a kernel function. The e-insensitive loss function is attractive because unlike the quadratic and Huber cost function, where all the data points will be support vectors, the SV solution can be sparse. The quadratic loss function produces a solution which is equivalent to ridge regression, or zeroth order regularisation, where the regularisation parameter $\lambda = \frac{1}{2C}$

A.3 SVM-RFE (SVM-Recursive Feature Elimination)

A good feature ranking criterion is not necessarily a good feature subset ranking criterion. The criteria *DJ(i)* (Cost Function) or $(w_i)^2$ estimate the effect of removing one feature at a time on the objective function. They become very sub-optimal when it comes to removing several features at a time, which is necessary to obtain a small feature subset. This problem can be overcome by using the following iterative procedure that we call Recursive Feature Elimination:

1. Train the classifier (optimize the weights wi with respect to J).
2. Compute the ranking criterion for all features ($DJ(i)$ or $(w_i)^2$).
3. Remove the feature with smallest ranking criterion.

This iterative procedure is an instance of backward feature elimination (Kohavi 1997). For computational reasons, it may be more efficient to remove several features at a time, at the expense of possible classification performance degradation. In such a case, the method produces a feature subset ranking, as opposed to a feature ranking. Feature subsets are nested $F_1 \subset F_2 \cdots F$.

If features are removed one at a time, there is also a corresponding feature ranking. However, the features that are top ranked (eliminated last) are not necessarily the ones that are individually most relevant. Only taken together the features of a subset Fm are optimal in some sense. It should be noted that RFE has no effect on correlation methods since the ranking criterion is computed with information about a single feature.

SVM-RFE is an application of RFE using the weight magnitude as ranking criterion.
SVM-RFE algorithm is described below;

Inputs:

Training examples $X_0 = [x_1, x_2 \cdots x_k \cdots x_l]^T$

Class labels $y = [y_1, y_2 \cdots y_k \cdots y_l]^T$

Initialize:

Subset of surviving features $s = [1,2,\cdots n]$

Feature ranked list r = [ ]

Repeat until s = [ ]

Restrict training examples to good feature indices $X = X_0(:, s)$

Train the classifier $\alpha = SVM - Train(X, y)$

Compute the weight vector of dimension length(**s**) $w = \sum_k \alpha_k y_k x_k$

Compute the ranking criteria $c_i = (w_i)^2$, for all *i*

Find the feature with smallest ranking criterion f = *argmin*(**c**)

Update feature ranked list $r = [s(f), r]$

Eliminate the feature with smallest ranking criterion $s = s(1: f-1, f+1: length(s))$

Output:

Feature ranked list *r*.

As mentioned before the algorithm can be generalized to remove more than one feature per step for speed reasons.

Appendix B

Overview of the Transparent Machine Learning Techniques

This section provides the detailed overview of the machine learning techniques used for rule generation purpose. The techniques discussed in this section are Fuzzy Rule Based Systems (FRBS), Decision Tree (DT), Classification and Regression Tree (CART), Adaptive Network-based Fuzzy Inference Systems (ANFIS), Dynamic Evolving Fuzzy Inference Systems (DENFIS) and NBTree (Naive Bayes Tree).

B.1 Fuzzy Rule Based Systems (FRBS)

Fuzzy rule based systems have been successfully applied to various control problems (Sugeno, 1985) where fuzzy rules are usually derived from human experts as linguistic *if-then* rules. If the comprehensibility of fuzzy *if-then* rules by human users is a criterion in designing a fuzzy rule-based system, a fuzzy partition by a simple fuzzy grid with pre-specified membership functions is preferable. An example of such partition is given in Figure B.1, where each axis of a two dimensional pattern space is homogeneously partitioned by five linguistic values (*S: small, MS: medium small, M: medium, ML: medium large and L: large*). It has been often claimed that grid-type fuzzy partitions such as Figure B.1 cannot handle high dimensional problems with many input variables due to the curse of dimensionality (Carse et al., 1996). That is, when we use the grid-type fuzzy partition, the number of fuzzy if-then rules exponentially increases as the number of input variables increases. Ishibuchi (1999), however, used the grid-type fuzzy partition for pattern classification problems with many continuous features, because such a fuzzy partition maintains an inherent advantage of fuzzy rule-based systems. They dealt with the curse of dimensionality by utilizing “don’t care” as an antecedent fuzzy set and generating

only a small number of promising fuzzy if-then rules by genetic operations. The antecedent fuzzy set “don’t care” is represented by an interval-type membership function whose membership value is always unity in the domain of each feature value (Ishibuchi et al., 1997). Figure B.2 shows an example fuzzy partition that incorporates with the antecedent fuzzy se “don’t care.”

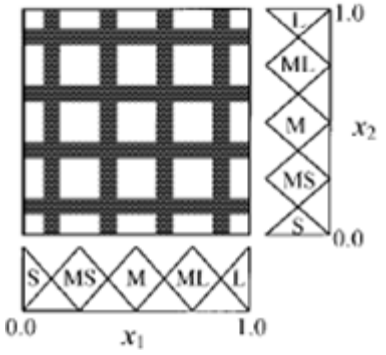


Figure B.1: Example of fuzzy partition by simple fuzzy grid with five linguistic values for each axis of the 2-D pattern space [0, 1] x [0, 1]

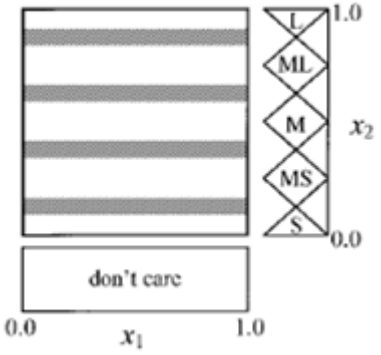


Figure B.2: Example of fuzzy partition of the 2-D pattern space [0, 1] x [0, 1] with “don’t care” as an antecedent fuzzy set.

Ishibuchi et al, (1999) presented a fuzzy rule-based system that is constructed from a set of labeled samples. The constructed fuzzy rule-based system assigns a new feature vector to one of given classes. That is, fuzzy rule based systems are application to the same dataset as non fuzzy classification methods. One advantage of fuzzy rule base systems over other

classification methods is their comprehensibility. We can easily understand them because each fuzzy if-then rule is interpreted through linguistic values such as “small” and “large.” High classification ability is another advantage of fuzzy rule-based systems.

B.1.1 Rule Generation

R_j: if x_1 is A_{j1} and ... and x_n is A_{jn}

then Class C_j with $CF = CF_j$

where R_j is the label of the j^{th} fuzzy if-then rule, A_{j1}, ..., A_{jn} is antecedent fuzzy sets on the unit interval [0, 1], C_j is consequent class and CF_j is grade of certainty of the fuzzy if-then rule R_j.

It should be noted that the grade of certainty CF_j is different from the fitness value of each rule. The fitness value is used in a selection operation of the fuzzy classification system while CF_j is used in fuzzy reasoning for classifying new patterns.

As antecedent fuzzy sets, they use five linguistic values as in Figure 1 and “don’t care” in Figure 2. Thus, for the n-dimensional pattern classification problem, the total number of fuzzy if-then rules is $(5+1)^n$. It is impossible to use all the rules in fuzzy rule based system when the number of features i.e. n is large. Ishibuchi et al. (1999) tries to search for a set of a relatively small number of fuzzy if-then rules.

Determination of Cj and CFj

Step 1: calculate the compatibility grade of each training pattern $x_p=(x_{p1}, x_{p2},..., x_{pn})$ with fuzzy if-then rule R_j by the following product operation:

$$\mu_j(x_p) = \mu_{j1}(x_{p1}) \times \cdots \times \mu_{jn}(x_{pn}) \qquad 3$$

where $\mu_{A_{ji}}(x_{p1})$ is the membership function of A_{ji}.

Step 2: For each class, calculate the sum of the compatibility grades of the training patterns with the fuzzy if-then rule R_j

$$\beta_{Classh}(R_j) = \sum_{x_p \in Classh} \mu_j(x_p), \qquad h = 1, 2, \ldots, c \qquad 4$$

where c the sum of the compatibility grades of the training patterns in Class *h* with the fuzzy if-then rule R_j.

Step 3: Find Class hj that has the maximum value of $\beta_{Classh}(R_j)$

$$\beta_{Class\hat{h}}(R_j) = Max\{\beta_{Class1}(R_j), \ldots, \beta_{Classc}(R_j)\}. \qquad 5$$

Step 4: If the consequent class C_j is 8, let the grade of certainty CF_j of the fuzzy if-then rule R_j be $CF_j = 0$. Otherwise the grade of certainty CF_j is determined as follows:

$$CF_j = \{\beta_{Classh}(R_j) - \bar{\beta}\} \Big/ \sum_{h=1}^{c} \beta_{Classh}(R_j) \qquad 6$$

where

$$\bar{\beta} = \sum_{\substack{h=1 \\ h \neq h_j}}^{c} \beta_{Classh}(R_j)/(c-1) \qquad 7$$

B.1.2 Fuzzy Reasoning

When the antecedent fuzzy sets of each fuzzy if-then rule are given, the consequent class and the grade of certainty can be determined. Let S be the fuzzy if-then rule set. An input pattern X_p to the fuzzy rule-based system with the rule set *S* is classified by a fuzzy reasoning method. Ishibuchi et al. (1999) perform the fuzzy reasoning via the single winner rule. The winner rule R_j for the input pattern X_p is determined as

$$\mu_{\hat{j}}(x_p) \cdot CF_{\hat{j}} = Max\{\mu_j(x_p) \cdot CF_j \;\; R_j \in s\}. \qquad 10$$

That is, the winner rule has the maximum product of the compatibility $\mu_j(x_p)$ and the grade of certainty CF_j. If more than one fuzzy if-then rule have the same maximum product but different consequent classes for the input pattern x_p, the classification is rejected. The classification is also rejected if no fuzzy if-then rule is compatible with the

input pattern x_p ($\mu_n(x_p)$ = 0 for R_j e S). This fuzzy reasoning method based on a single winner rule leads to simplicity in the credit assignment algorithm in our fuzzy classifier system, as only one rule is responsible for the classification result of each input pattern. It is possible to modify our fuzzy reasoning method to account for the situation when different classes have the same maximum value in (10), and for when no fuzzy if-then rule is compatible with the input pattern x_p. After this modification, classification rates of fuzzy rule-based systems are improved because rejection rates are decreased. At the same time, there is an increase in error rates. Thus, such a modification is promising when the penalty of rejection is not small.

B.1.3 Coding of Fuzzy if-then rules

Ishibuchi et al., 1999 denote five linguistic values and "don't care" by the following six symbols (i.e. 1, 2, 3, 4, 5 and #).

S: small
MS: medium small
M: medium
ML: medium large
L: large
DC: don't care

Each fuzzy if-then rule can be denoted by a string of these six symbols. For example, a string "1#3#" denotes a four-dimensional patter classification problem as follows;

If x_1 is small and x_3 is medium then class C_j with $CF = CF_j$.

B.1.4 Outline of Fuzzy Classifier System

Ishibuchi et al., 1999 proposed a fuzzy classifier system based on the heuristics rule generation procedure discussed in next section. Genetic operations such as selection, crossover and mutation are used for generating a combination of antecedents fuzzy sets of each fuzzy if-then rule. the outline of the fuzzy classifiers system is as follows;

Step 1: Generate an initial populations of fuzzy if-then rules;

Step 2: Evaluate each fuzzy if-then rule in the current population;

Step 3: Generate new fuzzy if-then rules by genetic operations.

Step 4: Replace a part of the current population with the newly generated rules.

Terminate the algorithm if a stopping criteria is satisfied, otherwise return to step 2.

B.1.5 Initial Population

Let us denote the number of fuzzy if-then rules in each population in our fuzzy classifier system by N_{pop}. To construct an initial population, N_{pop} fuzzy if-then rules are generated by randomly selecting their antecedent fuzzy sets from the six symbols corresponding to the five linguistic values and "don't care". Each symbol is randomly selected with the probability of 1/6.

B.1.6 Evaluation of Each Rule

A unit reward is assigned to the winner rule when a training pattern is correctly classified by that rule. After all the training patterns are examined, the fitness value of each fuzzy if-then rule is defined as follows by the total reward assigned to that rule

$$fitness(R_j) = NCP(R_j) \qquad 11$$

where fitness(R_j) is the fitness value of the fuzzy if-then rule R_j and $NCP(R_j)$ is the number of training patterns that are correctly classified by R_j. Fitness value of each fuzzy if-then rule is updated by 11 at each generation.

B.1.7 Genetic Operations for Generating New Rules

In order to generate new fuzzy if-then rules, first a pair of fuzzy if-then rules is selected from the current population. Each fuzzy if-then rule in the current population is selected based on the roulette wheel selection with linear scaling;

$$P(R_j) = \frac{fitness(R_j) - fitness_{\min}(S)}{\sum_{R_j \in S}\{fitness(R_j) - fitness_{\min}(S)\}} \qquad 12$$

where $fitness_{min}(S)$ is the minimum fitness value of the fuzzy if-then rules in the current population S.

From the selected pair of fuzzy if-then rules, two rules are generated by the uniform crossover for the antecedent fuzzy sets, as illustrated in Figure B.3. Only antecedent fuzzy sets of the selected pair of fuzzy if-then rules are mated. As crossover operation, mutation also applied on antecedent part of the fuzzy if-then rule as illustrated in Figure B.4. Using a pre-specified mutation probability each antecedent fuzzy set of the fuzzy if-then rules generated by the crossover operation is randomly replaced.

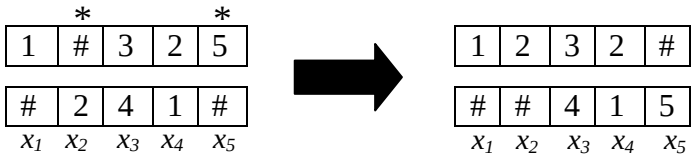


Figure B.3: Uniform crossover for antecedent fuzzy sets (* denotes a crossover position)

Figure B.4: Mutation for antecedent fuzzy sets (* denotes a mutation position)

These genetic operations (i.e. selection, crossover and mutation) are iterated until a pre-specified number of fuzzy if-then rules are newly generated.

B.1.8 Rule Replacement

A prespecified number of fuzzy if-then rules (N_{rep}) in the current population are replaced with the newly generated rules by the genetic operations. N_{rep} rules with the smallest fitness values are removed from the current population and the newly generated fuzzy if-then rules are added.

Termination Test

Ishibuchi et al., (1999) used the total number of generations as a stopping condition. The final solution obtained by this FRBS is the rule set with the maximum classification rate

for training patterns over all generations. That is, the final solution is not the final population but the best population. Software package KEEl 1.0 is used to employ FRBS in this thesis.

B.2 Decision Tree

In this section, we describe the development of decision trees for classification tasks. These trees are constructed beginning with the root of the tree and proceeding down to its leaves. Decision tree is employed using Knime data mining tool.

One approach to the induction task above would be to generate all possible decision trees that correctly classify the training set and to select the simplest of them (Pearl, 1978b; Quinlan 1983a). The number of such trees is finite but very large, so this approach would only be feasible for small induction tasks. The basic structure of ID3 is iterative. A subset of the training set called the window is chosen at random and a decision tree formed from it; this tree correctly classifies all objects in the window. All other objects in the training set are then classified using the tree. If the tree gives the correct answers for all these objects then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified objects is added to the window and the process continues. This way, correct decision trees have been found after only a few iterations. Empirical evidence suggests that a correct decision tree is usually found more quickly by this iterative method than by forming a tree directly from the entire training set. O'Keefe (1983) noted that the iterative framework cannot be guaranteed to converge on a final tree unless the window can grow to include the entire training set.

Why are decision tree classifiers so popular? The construction of decision tree classifier does not require any domain knowledge or parameter setting, and there is appropriate for exploratory knowledge discovery. Decision tree can handle high dimensional data. Their representation of acquired knowledge in the form of tree is intuitive and generally easy to assimilate by humans. The learning and classification of decision tree induction are simple and fast. Generally, decision tree classifiers have good prediction accuracy. However, successful use may depend on the data at hand.

B.2.1 Classification by Decision Tree Induction

Decision tree induction is the learning of decision trees from class-labelled training objects. A decision tree is a flow chart like tree structure, where each internal node (non

leaf node) denotes a test on an feature, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. The topmost node in a tree is the root node. An example tree is shown in Figure B.5 below;

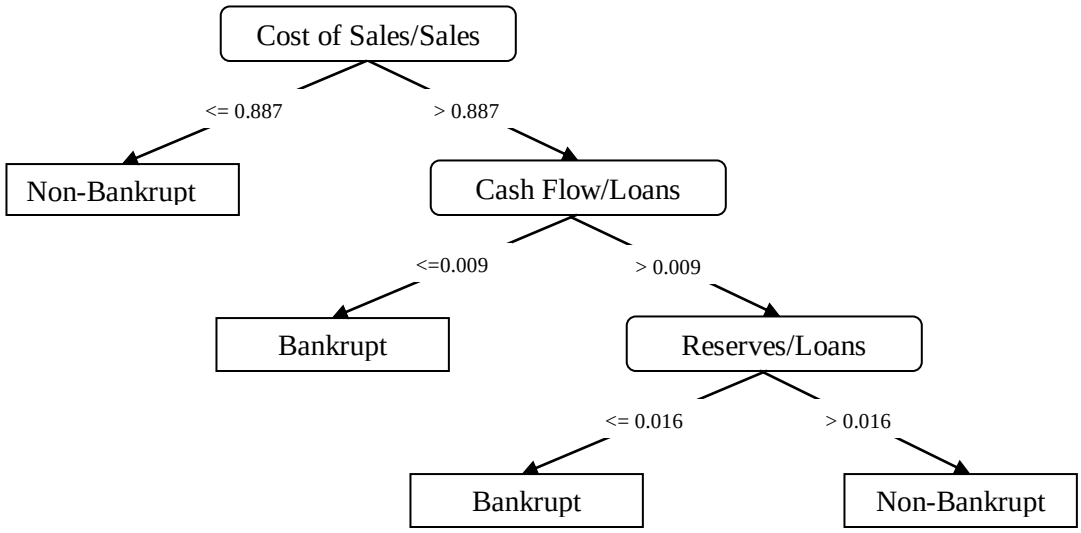


Figure B.5: Example Decision Tree for Bankruptcy prediction in Banks

How are decision trees used for classification? Given a tuple *X*, for which the associated class label is unknown, the feature values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

During the late 1970s and early 1980s, J. Ross Quinlan, a researcher in machine learning, developed a decision tree algorithm known as ID3 (Iterative Dichotomiser). Quinlan later presented C4.5 (a successor of ID3), which became a benchmark to which newer supervised learning algorithms are often compared.

C4.5 adopt a greedy (i.e. non backtracking) approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner.

The algorithm is called with three parameters: *D, attribute_list* and *Attribute_selection_method*. *D* is a data partition which is the complete set of training

objects and their associated class labels. *Attribute_selection_method* specifies a heuristic procedure for selecting the feature that "*best*" discriminates the given objects according to class. *Information gain* or *gini index* are the feature selection measures usually employed.

B.2.2 Algorithm:

Generate_Decision_Tree. Generate a decision tree from the training objects of data partition *D*.

Input:

- ✓ *Data partition, D*
- ✓ *attribute_list*
- ✓ *Attribute_selection_method*

Output: A *decision tree*.

Method:

1. Create a node *N*;
2. If objects in *D* are all of the same class, *C* then
3. Return *N* as a leaf node labelled with the class *C*;
4. If *attribute_list* is empty then
5. Return *N* as a leaf node labelled with the majority class in *D*; (majority voting)
6. Apply *attribute_selection_method*(*D*, *attribute_list*) to find the "*best*" *splitting_criterion*;
7. Label node *N* with *splitting_criterion*;
8. If *splitting_attribute* is discrete-valued and Multiway splits allowed then // not restricted to binary trees
9. *attribute_list* ---- attribute_list – splitting_attribute; // remove splitting_attribute
10. For each outcome *j* of *splittitng_criterion* // partition the objects and grow subtrees for each partition

11. Let D_j be the set of data objects in D satisfying outcome j; // a partition
12. If D_j is empty then
13. Attach a leaf labelled with majority class in D to node N;
14. Else attach the node returned by *Generate_decision_tree*(D_j, *attribute_list*) to node N;

    End for
15. Return N.

Basic algorithm for inducing a decision tree from training examples

The tree starts as a single node, N, representing the training objects in D (step 1).

If the objects in D are all of the same class, then node N becomes a leaf and is labelled with that class (step 2 and 3).

Otherwise, the algorithm calls *Attribute_selection_method* to determine the splitting criterion. The splitting criterion decides feature to test at node N by determining the "*best*" way to separate or partition the objects in D into individual classes (step 6).

The node N is labelled with the splitting criterion, which serves as a test at the node (step 7). A branch is grown from node N for each of the outcomes of the splitting criterion. The objects in D are partitioned accordingly (step 10 to 11). Based on the nature of the data there are three possible scenarios. Let A be the splitting feature. A has v distinct values, $\{a_1, a_2, ..., a_v\}$, based on the training data.

1. A is discrete-valued: in this case, the outcomes of the test at node N corresponds to the known values of A. A branch is created for each known value, a_j of A and labelled with that value (Figure B.6 a). Partition D_j is the subset of class-labelled objects in D having value a_j of A. Because all of the objects in a given partition have the same value for A, then A need not be considered in any future partitioning of the objects. Therefore, it is removed from *attribute_list* (step 8 and 9).

2. *A* is continuous-valued: in this case, the test at node *N* has two possible outcomes corresponding to the conditions *A*≤ A <= *split_point* and A>*split_point*, respectively. Two branches are grown from *N* and labelled according to the above outcomes (Figure B.6 b). The objects are partitioned such that D_1 holds the subset of class-labelled objects in *D* for which A <= *split_point*, while D_2 holds the rest.

3. *A* is discrete-valued and a binary tree must be produced: the test at node *N* is of the form "*A belongs SA*?". *SA* is the splitting subset for *A*, returned by *Atribute_selection_method* as part of the splitting criterion. It is a subset of the known values of *A*. If a given object has value a_j of A and if a_j belongs to *SA*, then the test at node *N* is satisfied. Two branches are grown from *N* (Figure B.6 c). By convention, the left branch out of *N* is labelled *yes* so that D_1 corresponds to the subset of class labelled objects in *D* that satisfy the test. The right branch out of *N* is labelled *no* so that D_2 corresponds to the subset of class-labelled objects from *D* that do not satisfy the test.

The algorithm uses the same process recursively to from a decision tree for the objects at each resulting partition, D_j of *D* (step 14).

The recursive partitioning stops only when any one of the following terminating condition is true:

1. All of the objects in partition *D* belong to the same class (step 2 and 3).

2. There are no remaining features on which the objects may be further partitioned (step 4). In this case majority voting is employed (step 5).

3. There are no objects for a given branch, that is, a partition D_j is empty (step 12). In this case a leaf is created with majority class in *D* (step 13)

The resulting decision tree is returned (step 15)

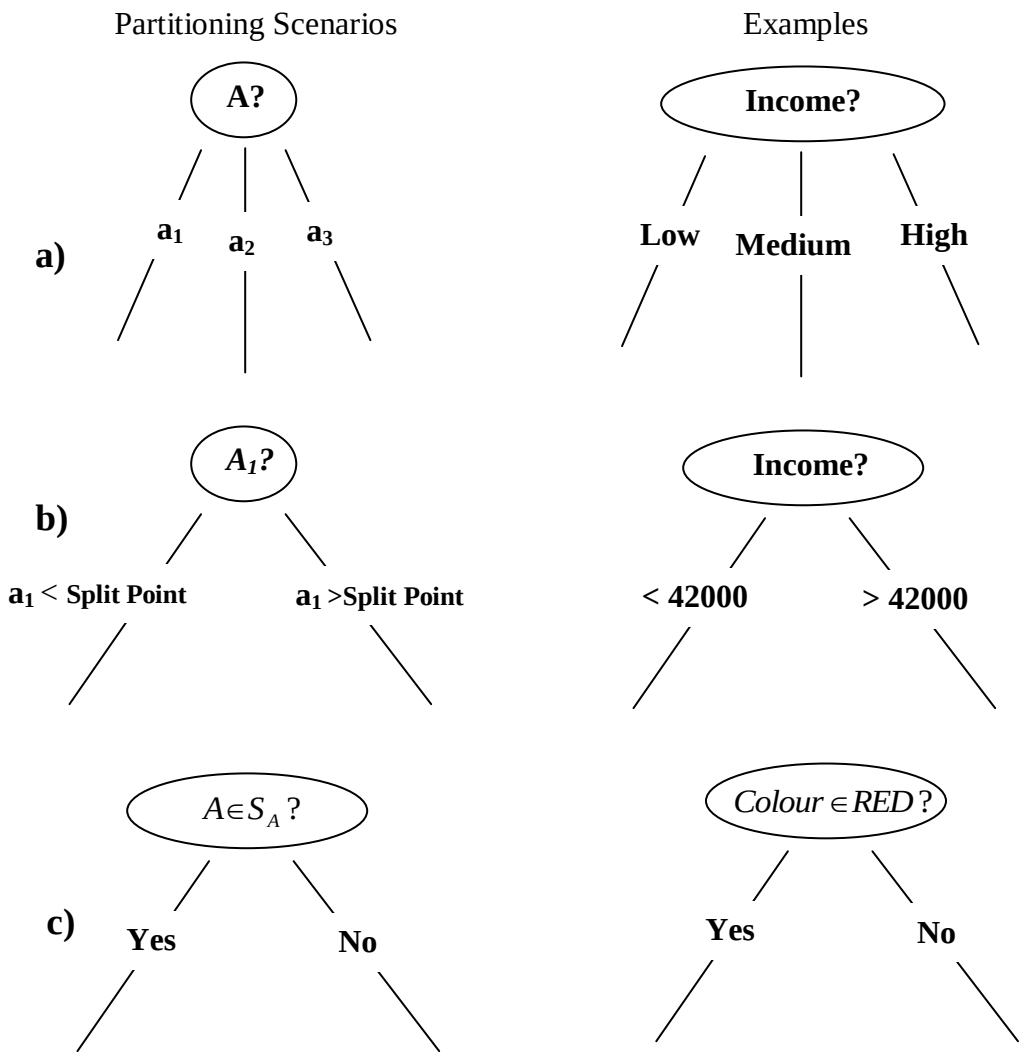


Figure B.6: Three possibilities for partitioning objects based on the splitting criterion, shown with examples.

B.2.3 Splitting Rule

In theory there are several impurity functions, but only two of them are widely used in practice: Gini splitting rule and Twoing splitting rule.

Gini Splitting Rule

Gini splitting rule (or Gini index) is most broadly used rule.

$$i(t)=\sum_{k\neq l} p(k\mid t)\,p(l\mid t)$$

where *k, l 1, ..., K* – index of the class; *p(k|t)* – conditional probability of class *k* provided we are in node *t*.

$$\Delta i(t) = -\sum_{k=1}^{K} p^2(k\mid t_p) + P_l\sum_{k=1}^{K} p^2(k\mid t_l) + P_r\sum_{k=1}^{K} p^2(k\mid t_r)$$

Therefore, Gini algorithm solves the following problem:

$$\arg\max_{x_j \le x_j^R,\ j=1,\ldots,M}\left[-\sum_{k=1}^{K} p^2(k\mid t_p) + P_l\sum_{k=1}^{K} p^2(k\mid t_l) + P_r\sum_{k=1}^{K} p^2(k\mid t_r)\right]$$

Gini algorithm searches for the largest class and isolate it from the rest of the data. it works well for noisy data.

B.3 CART

Classification and Regression Tree is a classification method which uses historical data to construct so-called decision tree. CART methodology was developed in 80s by Breiman, Freidman, Olshen and Stone (1984). For building decision trees CART uses learning sample which is a set of historical data with pre-assigned classes for all observations. For example, learning samples for credit scoring system would be fundamental information about previous borrows (variables) matched with actual payoff results (classes). Decision trees are represented by a set of questions which splits the learning sample into smaller and smaller parts. CART asks only yes/no questions and it will search for all possible variables and all possible values in order to find the best split – the question that splits the data into two parts with maximum homogeneity. The process is then repeated for each of the resulting data fragments. An example rule set obtained using CART for predicting forest fires is presented in Figure B.7.

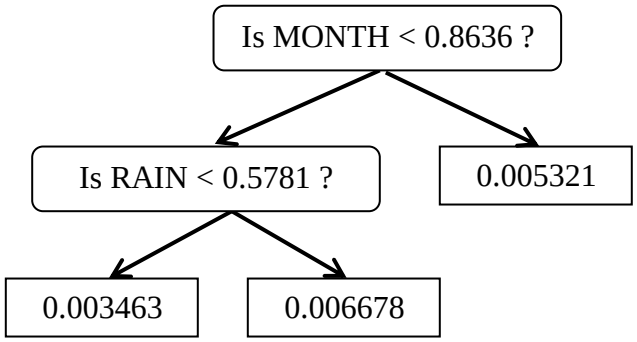


Figure B.7: Example rule set for forest fires data

CART methodology

- ✓ Construction of maximum tree
- ✓ Choice of the right tree size
- ✓ Classification of new data using constructed tree

B.3.1 Construction of maximum tree

Building the maximum tree implies splitting the learning sample up to last observations, i.e. when terminal nodes contain observations only of one class. Splitting algorithms are different for classification and regression trees.

Let t_p be a parent node and t_l, t_r – respectively left and right child nodes of parent node. Consider the learning sample with variable matrix X with M number of variables x_j and N observations. Let class vector Y consist of N observations with total amount of K classes. Classification tree is built in accordance with splitting rule, described in the previous section 2.4.2.–the rule that performs the splitting of learning sample into smaller parts i.e. each time, data have to be divided into two parts with maximum homogeneity as illustrated in Figure B.8.

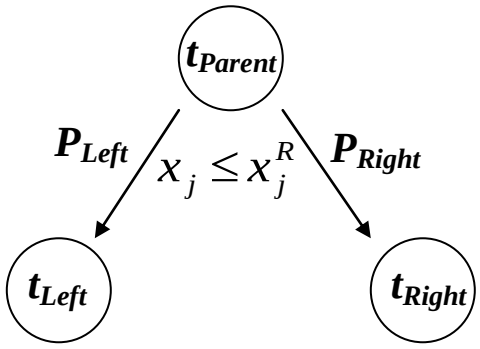


Figure B.8: Splitting Algorithm of CART

where t_p, t_l, t_r – parent, left and right nodes; x_j – variable j; x_j^R – best splitting value of variable x_j.

Maximum homogeneity of child nodes is defined by so-called impurity function *i(t)*. Since the impurity of parent node t_p is constant for any of the possible splits $x_j <= x_j^R$, $j = i, ..., M$, the maximum homogeneity of left and right child nodes will be equivalent to the maximization of change of impurity function *i(t)*:

$$\Delta i(t) = i(t_p) - E[i(t_c)]$$

Assuming that P_l, P_r – probabilities of left and right nodes, we get:

$$\Delta i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r)$$

Therefore, at each node CART solves the following maximization problem:

$$\underset{x_j \le x_j^R,\ j=1,\ldots,M}{\arg\max} \left[i(t_p) - P_l i(t_l) - P_r i(t_r) \right]$$

Above equation implies that CART will search through all possible values of all variables in matrix X for the best split question $x_j < x_j^R$ which will maximize the change of impurity measure $i(t)$.

Regression trees do not have classes, instead there are response vector Y which represents the response values for each observation in variable matrix X. Since regression trees do not have pre-assigned classes, classification splitting rule like Gini or Twoing cannot be applied.

Splitting in regression trees is made in accordance with squared residuals minimization algorithm which implies that expected sum variances for two resulting nodes should be minimized.

$$\underset{x_j \le x_j^R,\ j=1,\ldots,M}{\arg\max} \left[P_l\, Var(Y_l) + P_r\, Var(Y_r) \right]$$

where $Var(Y_l)$, $Var(Y_r)$ – response vectors for corresponding left and right child node; $x_j < x_j^R$, $j = 1, \ldots, M$ – optimal splitting question satisfying above condition. Squared residuals minimization algorithm is identical to Gini splitting rule.

B.3.2 Choice of the right tree size

Maximum trees may turn out to be very high complexity and consists of hundreds of levels. Therefore, they have to be optimized before being used for classification of new data. Two pruning algorithms can be used in practice: optimization by number of points in each node and cross-validation.

In the case of optimization by number of points, splitting is stopped when number of observations in the node is less than predefined required minimum N_{min}. Bigger N_{min} parameter, smaller will be the tree. This approach works very fast, it is easy to use and it has consistent results. But, it requires the calibration of new parameter N_{min}. In practice N_{min} is usually set to 10% of the learning sample size.

The procedure of cross-validation is based on optimal proportion between the complexity of the tree and misclassification error. With the increase in size of the tree, misclassification error is decreasing and in case of maximum tree, misclassification error is equal to 0. It is observed that complex decision trees poorly perform on independent data. Performance of decision tree on independent data is called true predictive power of the tree. Therefore – the primary task is to find the optimal proportion between the tree complexity and misclassification error.

$$R_\alpha(T) = R(T) + \alpha(\tilde{T}) \to \min_T$$

where *R(T)* – misclassification error of the tree *T*; $\alpha(\tilde{T})$ – complexity measure which depends on $\tilde{T}$ (total number of terminal nodes). α – parameter is found through the sequence of in-sample testing when a part of learning sample is used to build the tree, the other part of the data is taken as a testing sample. The process is repeated several times for randomly selected learning and testing samples.

B.3.3 Classification of new data using constructed tree

As the classification or regression tree is constructed, it can be used for classification of new data. the output of this stage is an assigned class or response value to each of the new observations. By set of questions in the tree, each of the new observations will get to one of the terminal nodes of the tree. A new observation is assigned with the dominating class / response value of terminal node, where this observation belongs to. Salford systems' CART software is used to employ CART algorithm.

B.3.4 Advantages and Disadvantages of CART

- It can handle both numerical and categorical variables.
- It is robust to outliers,
- It is the only tree capable of generating regression trees.
- The structure of its classification or regression trees is invariant with respect to monotone transformations of independent variables. One can replace any variable with its logarithm or square root value, the structure of the tree will not change.
- CART is a nonparametric approach.
- It does not require variables to be selected in advance.
- CART results are invariant to monotone transformations of its independent variables.
- CART has no assumptions and it is computationally fast.
- CART is flexible and has an ability to adjust in time.

Disadvantages of CART

- CART may have unstable decision trees.
- CART splits only by one variable.

B.4 Adaptive Network-based Fuzzy Inference System (ANFIS)

ANFIS is fuzzy inference system implemented in the framework of adaptive networks. Conventional (mathematical) system modelling is not well suited when dealing with ill-defined and uncertain systems. As the fuzzy inference system employing fuzzy if-then rules can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analyses. Fuzzy modelling or fuzzy identification was first explored systematically by Takagi and Sugeno (1985) and that has found various practical applications in control (Pedrycz, 1989; Sugeno, 1985), prediction and inference (Kandel, 1988; 1992). However, no standard methods exist for transforming human knowledge or experience into the rule base and there is no need for effective methods for tuning the membership functions so as to minimize the output error or maximize performance index. ANFIS serves as a basis for constructing a set of fuzzy if-then rules with appropriate membership functions to generate the stipulated input-output pairs. ANFIS is employed in MATLAB for experiments in this thesis.

B.4.1 Fuzzy if-then Rules

Fuzzy if-then rules are expressions of the form IF A THEN B, where A and B are labels of fuzzy sets (Zadeh, 1965) characterized by appropriate membership functions. Fuzzy if-then rules have the ability to capture the imprecise modes of reasoning that play an essential role in the human ability to make decisions in an environment of uncertainty and imprecision.

For example,

If a person is a "*heavy smoker*"

Then the risk of cancer is "*high*"

where person and risk are linguistic variables (Zadeh, 1973), heavy smoker and high are linguistic values that are characterized by membership functions.

Another form of fuzzy if-then rule (Takagi and Sugeno, 1983) involves fuzzy sets only in the premise part.

If velocity is high, the force = $k*(velocity)^2$.

B.4.2 Fuzzy Inference Systems

Basically FIS is composed of five functional blocks as illustrated in Figure B.9.

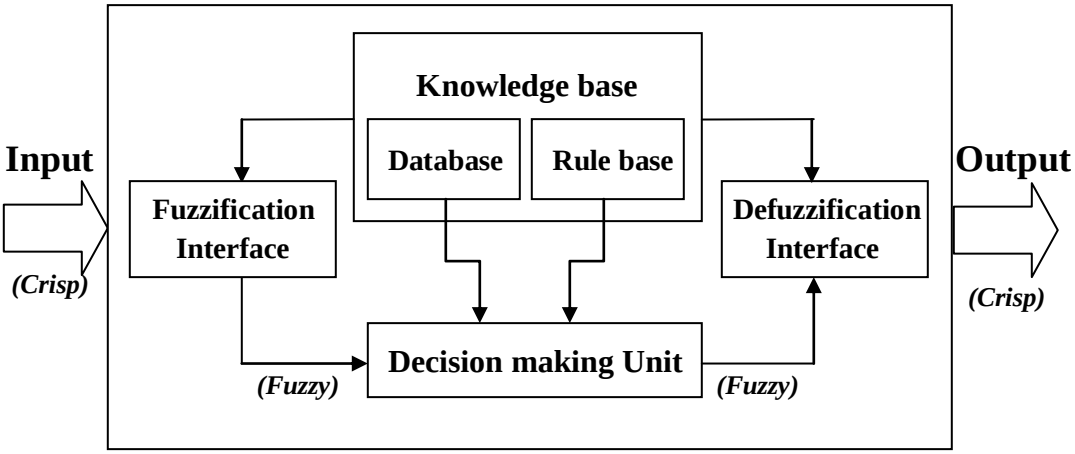


Figure B.9: Fuzzy Inference System

- A rule base consists of fuzzy if-then rules.
- A database defining membership functions of the fuzzy sets.
- A decision-making unit to perform inference operations.
- A fuzzification interface which transform the crisp input into degrees of match with linguistic values.
- A defuzzification interface which transforms the fuzzy results into a crisp output.

B.4.3 Adaptive networks

An adaptive network is a network structure consisting of nodes and directional links through which the nodes are connected (See Figure B.10). Moreover, part or all of the nodes are adaptive i.e. their outputs depend on the parameters pertaining to these nodes and the learning rule (i.e. gradient decent and the chain rule (Werbos, 1974)) which specifies how these parameters should be changed to minimize a prescribed error measure. Because

of the local minima problem of gradient method Jang et al., (1993) proposed a hybrid learning rule which can speed up the learning process substantially.

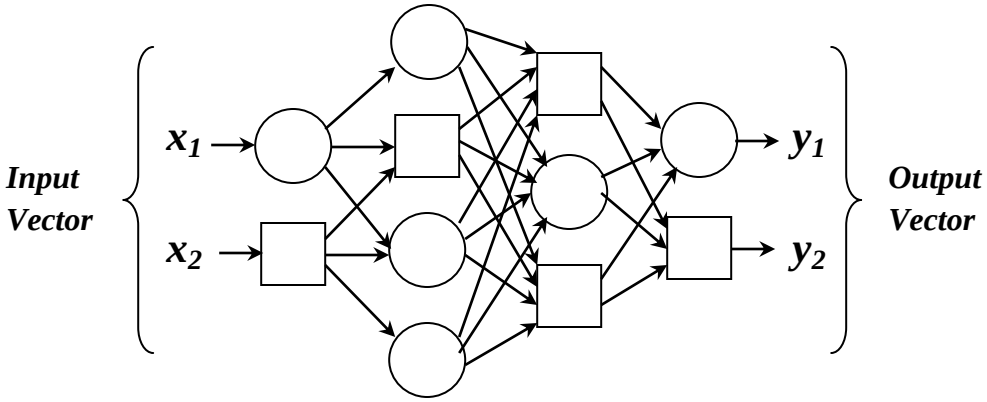


Figure B.10: Adaptive Networks

A square node (adaptive node) has parameters while a circle node (fixed node) does not. The parameter set of an adaptive network is the union of the parameter set of an adaptive network is the union of the parameter sets of each adaptive node. These parameters are updated according to given training data and a gradient-based learning procedure. Because the adaptive network used in ANFIS architecture needs to be feedforward type, the adaptive network's applications are immediate and immense in various areas.

B.4.4 ANFIS Architecture

For simplicity, the fuzzy inference system under consideration has two inputs x and y and one output *z*. Suppose the rule base contains two fuzzy if-then rules of Takagi and Sugeno's type (1983).

Rule 1: if x is A_1 and y is B_1, then $f_1=p_1x+q_1y+r_1$.
Rule 2: if x is A_2 and y is B_2, then $f_2=p_2x+q_2y+r_2$.

Figure B.11 *a* and *b* illustrates the type-3 fuzzy reasoning and the corresponding equivalent ANFIS architecture.

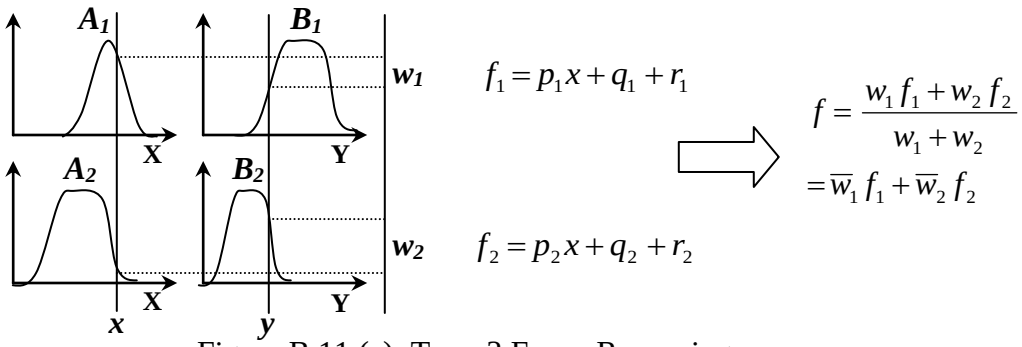


Figure B.11 (a): Type-3 Fuzzy Reasoning

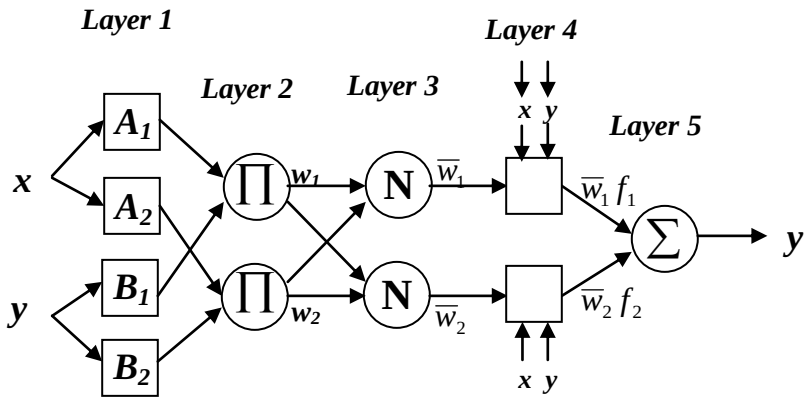


Figure B.11 (b): Equivalent ANFIS (Type-3 ANFIS)

Layer 1: every node i in this layer is a square node with a node function

$$O_i^1 = \mu_{A_i}(x)$$

where x is the input to node i, and Ai is the linguistic label associated with this node function. Usually membership function chosen is bell-shaped with maximum equal to 1 and minimum equal to 0, such as

$$\mu_{A_i}(x) = \frac{1}{1 + \left[\left(\frac{x - c_i}{\alpha_i}\right)^2\right]^{b_i}}$$

or

$$\mu_{A_i}(x) = \exp\left\{ -\left(\frac{x - c_i}{\alpha_i} \right)^2 \right\}$$

Where $\{a_i, b_i, c_i\}$ is the parameter set. Bell-shaped functions vary according to these parameters.

Layer 2: Every node in this layer is a circle node Labeled II, which multiplies the incoming signals and sends the product out, for instance,

$$w_i = \mu_{A_i}(x) \times \mu_{B_i}(y), \; i = 1, 2.$$

Each node output represents the firing strength of a rule.

Layer 3: Every node in this layer is a circle node labelled N. The ith node calculates the ratio of the ith rule's firing strength to the sum of all rules' firing strengths:

$$\overline{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2.$$

The output of this layer will be called normalized firing strengths.

Layer 4: Every node i in this layer is a square node with a node function

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i + r_i)$$

where wi is the output of layer 3, and {pi, qi, ri} is the parameter set. parameters in this layer are referred to as consequent parameters.

Layer 5: The single node in this layer is a circle node labelle SUM that computes the overall output as the summation of all incoming signals, i.e.,

$$O_1^5 = overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

Thus we have an adaptive network which is functionally equivalent to a type-3 fuzzy inference system.

Figure B.12 shows a 2-input, type-3 ANFIS with nine rules. Three membership functions are associated with each input, so the input space is partitioned into nine fuzzy subspaces, each of which is governed by a fuzzy if-then rule.

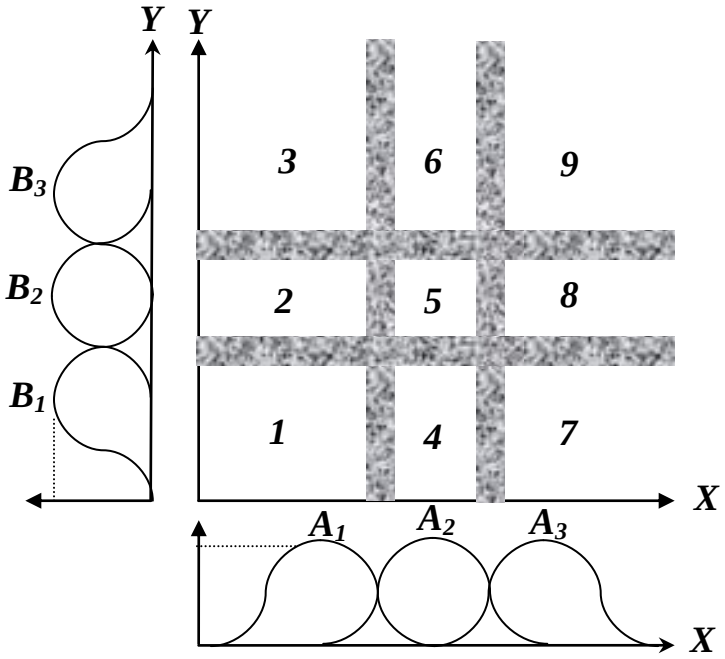


Figure B.12: Two input Type-3 ANFIS with nine rules

In fact, there are four methods to update the parameters, Gradient Descent Only, Gradient Descent and One Pass of LSE, Gradient Descent and LSE and Sequential (Approximate) LSE only. The choice of above mentioned methods should be based on the trade-off between computation complexity and resulting performance.

B.5 Dynamic Evolving Fuzzy Inference Systems (DENFIS)

The complexity and dynamics of real-world problems, especially in engineering and manufacturing, require sophisticated methods and tools for building online, adaptive intelligent systems. Such systems should be able to grow as they operate, to update their knowledge and refine the model through interaction with the environment (Amari and Kasabov, 1997, Kasabov, 1998a, Kasabov, 1996). Fast learning, online incremental adaptive learning, open structure organization, memorising information, active interaction, knowledge acquisition and self-improvement and spatial and temporal learning are some of the major requirements of the intelligent systems (Kasabov, 2001, Kasabov, 1998b, Kasabov 1998a, Kasabov and Woodford, 1999). Neucomn_Student software package is used for employing DENFIS in this thesis.

B.5.1 Evolving clustering method (ECM)

ECM is an evolving online maximum distance-based clustering method. ECM is carried out in two modes, first one is usually applied to online learning systems and the second one is more suitable for offline learning system. DENFIS's online model works based on online ECM.

B.5.2 Online ECM

Without any optimization, the online ECM is a fast, one-pass algorithm for a dynamic estimation of the number of clusters in a dataset, and for finding their current centers in the input data space. It is distance based connectionist clustering method. With this method, cluster centers are represented by evolved nodes. In any cluster the maximum distance, *MaxDist*, between an example point and the cluster center is less than a threshold value, *Dthr*, that has been set as a clustering parameter and would affect the number of clusters to be estimated.

In the clustering process, the data examples come from a data stream and this process starts with an empty set of clusters. When a new cluster is created, the cluster center *Cc* is defined and its cluster radius *Ru* is initially set to zero. With more examples presented one after another, some created clusters will be updated through changing their centers'

positions and increasing their cluster radius. A cluster will not be updated any more when its cluster radius, *Ru* reaches the value that is equal to threshold values, *Dthr*.

The ECM algorithm is described as follows.

Step 0: Create the first cluster C_1^0 by simply taking the position of the first example from the input stream as the first cluster Cc_1^0 and setting a value *0* for its cluster radius Ru_1.

Step 1: If all examples of the data stream have been processed, the algorithm is finished. Else, the current input example, x_i, is taken and the distance between this example and all *n* already created cluster centers Cc_j, $D_{ij} = ||x_i - Cc_j||$, $j = 1, 2, ..., n$, are calculated.

Step 2: If there is any distance value, $D_{ij} = ||x_i - Cc_j|$, equal to or less than at least one of the radii, Ru_j, $j = 1, 2, ..., n$, it means that the current example x_i belongs to a cluster C_m with minimum distance. $D_{im} = ||x_i - Cc_m|| = \min(||x_i - Cc_j||)$ subject to the constraint $D_{ij} <= Ru_j$, $j=1, 2, ..., n$.

In this case, neither a new cluster is created nor any existing cluster is updated (in cases of x_4 and x_6 in Figure B.13) the algorithm returns to step 1 else go to next step.

Step 3: Find cluster C_a (with center Cc_a and radius Ru_a) from all *n* existing cluster centers through calculating the values $S_{ij} = D_{ij} + Ru_j$, $j = 1, 2, ..., n$, and then choosing the cluster center Cc_a with minimum value S_{ia}: $S_{ia} = D_{ia} + Ru_a = min(S_{ij})$, $j = 1, 2, ..., n$.

Step 4: If S_{ia} is greater than 2 * *Dthr*, the example x_i does not belong to any existing cluster. A new cluster is created in the same way as described in step 0 (the cases of x_3 and x_8 in Figure B.13) and the algorithm returns to step 1.

Step 5: If S_{ia} is not greater than 2 * *Dthr*, the cluster C_a is updated by moving its center, Cc_a and increasing the value of its radius, Ru_a. The updated radius Ru_a^{new} is set to be equal to $S_{ia}/2$ and the new center C_a^{new} is located at the point on the line connecting x_i and Cc_a and the distance from the new center Cc_a^{new} to the point x_i is equal to Ru_a^{new} (the cases of x_2, x_5, x_7 and x_9 in Figure B.13). The algorithm returns to Step 1.

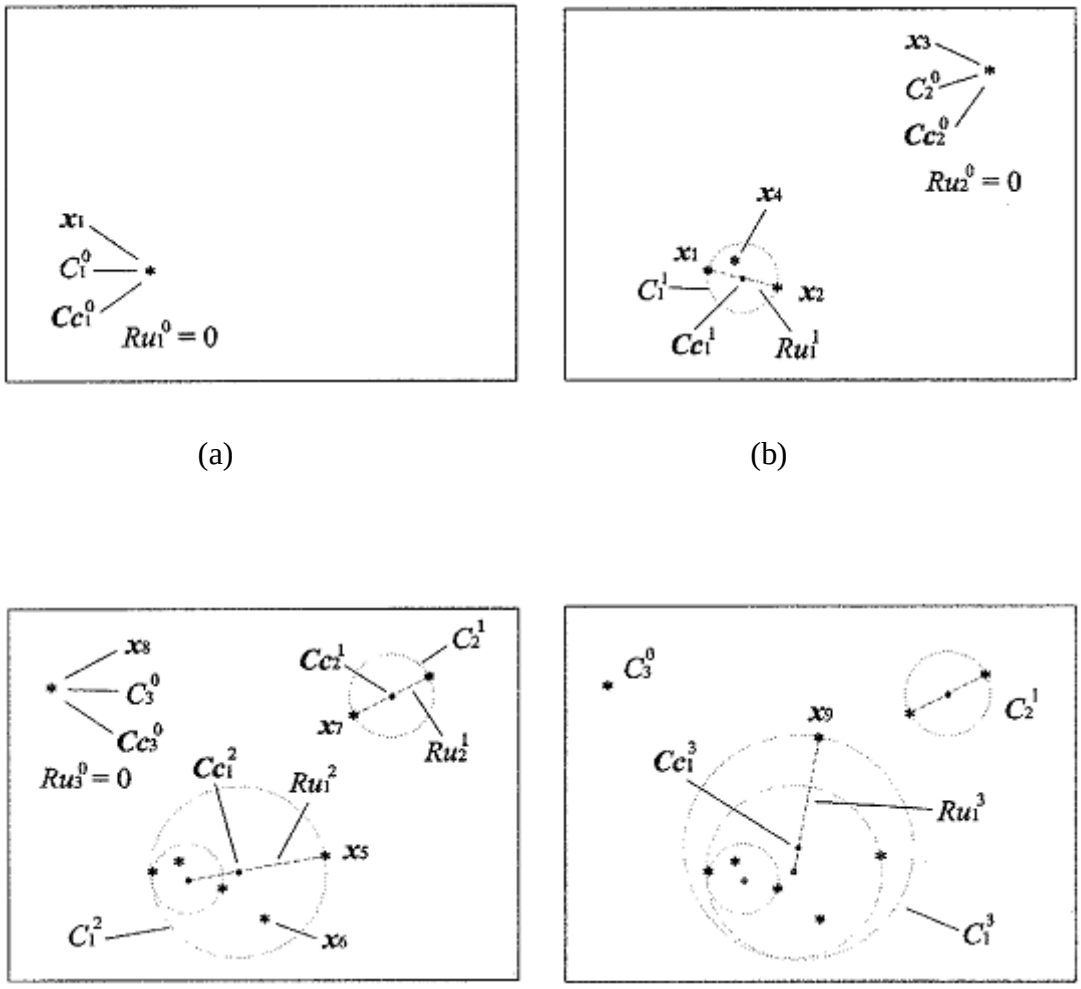


Figure B.13: A brief clustering process using ECM with samples x_1 to x_9 in a 2-D space. (a) The example x_1 causes the ECM to create a new Cluster C_1^0. (b) x_2: update cluster $C_1^0 \rightarrow C_1^1$; x_3: create a new cluster C_2^0; x_4: do nothing. (c) x_5: update cluster $C_1^1 \rightarrow C_1^2$; x_6: do nothing, x_7: update cluster $C_2^0 \rightarrow C_2^1$; x_8: create a new cluster C_3^0. (d) x_9: update cluster $C_1^2 \rightarrow C_1^3$.

B.5.3 DENFIS: Dynamic Evolving Neural-Fuzzy Inference Systems

Online and offline models of DENFIS use Takagi-Sugeno type fuzzy inference system (Takagi and Sugeno 1985). In first layer pre-processing using online ECM is done and using the clusters obtained FIS is generated. Such FIS is composed of *m* fuzzy rules indicated as shown below, where "x_j is R_{ij}" i = 1, 2, ..., m, j = 1, 2, ..., q, are *m*q* fuzzy proportions as *m* antecedents from *m* fuzzy rules respectively. X_j, *j = 1, 2, ..., q,* are fuzzy sets defined by their fuzzy membership functions $\mu_{R_{ij}} : X_j \rightarrow [0,1]$, *i = 1, 2, ..., m; j = 1, 2, ..., q*. In the consequent part *y* is a consequent variable, and polynomial functions f_i, *i = 1, 2, ..., m*, are employed.

if x_1 is R_{11} and x_2 is R_{12} and ... and x_q is R_{1q}, then *y* is $f_1(x_1, x_2,..., x_q)$
if x_1 is R_{21} and x_2 is R_{22} and ... and x_q is R_{2q}, then *y* is $f_2(x_1, x_2,..., x_q)$
...
if x_1 is R_{m1} and x_2 is R_{m2} and ... and x_q is R_{mq}, then *y* is $f_m(x_1, x_2,..., x_q)$

In both DENFIS online and offline models, all fuzzy membership functions are triangular type functions which depend on three parameters as given in the following equation:

$$\mu(x) = mf(x,a,b,c) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x \le c \\ 0. & c \le x \end{cases}$$

Where: *b* is the value of the cluster center on the *x* dimension, *a=b-d*Dthr* and *c=b+d*Dthr, d=1.2-2,* the threshold value, *Dthr* is clustering parameter.

If the consequent functions are crisp constants then such systems are called zero-order Takagi-Sugeno type FIS. FIS is called first-order if consequent part is a linear function and

if the consequent part is non-linear function then it is called high-order Takagi-Sugeno FIS.

For an input vector $x_0 = [x_{10}, x_{20}, ..., x_{q0}]$, the result of inference, y_0 (the output of the system) is the weighted average of each rule's output indicated as follows:

$$y_0 = \frac{\sum_{i=1}^{m} \omega_i f_i(x_{10}, x_{20}, ..., x_{q0})}{\sum_{i=1}^{m} \omega_i}$$

where $\omega_i = \prod_{j=1}^{q} \mu_{R_{ij}}(x_{j0})$; $i = 1, 2, ..., m; j = 1, 2, ..., q$.

B.6 Naive-Bayes Tree (NBTree)

A classifier provides a function that maps (classifies) a data item (instance) into one of several predefined classes (Fayyad et al., 1996). The automatic induction of classifiers from data not only provides a classifier that can be used to map new instances into their classes, but may also provide a human-comprehensible characterization of the classes. In many applications, interpretability of the induction algorithm is very crucial. Some classifiers are naturally easier to interpret than others; for example decision trees (Quinlan, 1993) are easy to visualize, while neural networks are much harder. NBTree algorithm is employed using RapidMiner software.

Previous section in this chapter gave away the details about the decision trees. Naive-Bayes (Good 1965; Langley et al., 1992) uses Bayes rule to compute the probability of each class given the instance, assuming the features are conditionally independent given the label.

B.6.1 NBTree: The Hybrid Algorithm

The algorithm is similar to the classical recursive partitioning schemes, except that the leaf nodes created are Naive-Bayes categorizers instead of nodes predicting a single class. A threshold for continuous features is chosen using the standard entropy minimization technique. The utility of a node is computed by discretizing the data and computing the 5-fold cross validation accuracy estimate of using Naive-Bayes at the node.

Intuitively, it is attempted to approximate whether the generalization accuracy for a Naive-Bayes classifier at each leaf is higher than a single Navie-Bayes classifier at the current node. To avoid splits with little value, it is defined that a split to be significant if the relative reduction in error is greater than 5% and there are at least 30 instances in the node.

Input: a set *T* of labelled instanes.

Output: a decision tree with navie-bayes categorizers at the leaves.

1. For each feature X_i, evaluate the utility, $u(X_i)$, of a split on feature X_i, for continuous features, a threshold is also found at this stage.
2. Let j = $argmax_i(u_i)$, i.e., the feature with the highest utility.
3. If u_j is not significantly better than the utility of the current node, create a Naive-Bayes classifier for the current node and return.
4. Partition T according to the test on X_j. If X_j is continuous, a threshold split is used; if X_j is discrete, a multi-way split is made for all possible values.

For each child, call the algorithm recursively on the portion of T that matches the test leading to the child.

Naive-Bayes classifiers (Langley and Simon 1995) are generally easy to understand and the induction of these classifiers is extremely fast, requiring only a single pass through the data if all features are discrete. Naive-Bayes classifiers are also very simple and easy to understand (Kononenko 2001). Naive-Bayes classifiers are very robust to irrelevant features and classification takes into account evidence from many features to make the final prediction, a property that is useful in many cases where there is no "main effect". The limitation of Naive-Bayes classifier is that it requires strong independence assumptions and when these are violated, the achievable accuracy may asymptote early and will not improve much as the database size increases.

Decision tree classifiers are also fast and comprehensible, but current induction methods based on recursive partitioning suffer from the fragmentation problem: as each split is made, the data is split based on the test and after two dozen levels there is usually very little data on which to base decisions.

A hybrid approach is attempted to utilize the advantages of both decision tree (i.e. segmentation) and Naive-Bayes (evidence accumulation from multiple features). A decision tree is built with univariate split at each node, but with Naive-Bayes classifiers at the leaves. The final classifier resembles Utgoff's perceptron trees (Utgoff, 1988). The resulting classifier is as easy to interpret as decision trees and Naive-Bayes. The decision tree segments the data, a task that is considered an essential part of the data mining process

in large databases. Each segment of the data, represented by a leaf, is described through a Naive-Bayes classifier.

Appendix C:

Datasets Descriptions

This section provides the details about the feature information about the datasets analyzed during the research study presented in this thesis.

Table C.1: Financial Ratios of Spanish Banks Dataset

#	***Predictor Variable Name***	
1	***C**urrent **A**ssets/**T**otal **A**ssets*	*CA/TA*
2	***C**urrent **A**ssets-**C**ash/**T**otal **A**ssets*	*CAC/TA*
3	***C**urrent **A**ssets/**L**oans*	*CA/L*
4	***R**eserves/**L**oans*	*R/L*
5	***N**et **I**ncome/**T**otal **A**ssets*	*NI/TA*
6	***N**et **I**ncome/**T**otal **E**quity **C**apital*	*NI/TEC*
7	***N**et **I**ncome/**L**oans*	*NI/L*
8	***C**ost Of **S**ales/**S**ales*	*CS/S*
9	***C**ash **F**low/**L**oans*	*CF/L*

Table C.2: Financial Ratios of Turkish Banks Dataset

#	***Predictor Variable Name***	
1	***I**nterest **E**xpenses/**A**verage **P**rofitable **A**ssets*	*IE/APA*
2	***I**nterest **E**xpenses/**A**verage **N**on-Profitable **A**ssets*	*IE/ANA*
3	*(**S**hare Holders' **E**quity + **T**otal **I**ncome)/(**D**eposits + **N**on-Deposit **F**unds)*	*(SE+TI)/(D+NF)*
4	***I**nterest **I**ncome/**I**nterest **E**xpenses*	*II+IE*
5	*(**S**hare Holders' **E**quity + **T**otal **I**ncome)/**T**otal **A**ssets*	*(SE+TI)/TA*
6	*(**S**hare Holders' **E**quity + **T**otal **I**ncome)/(**T**otal **A**ssets + **C**ontingencies and **C**ommitments)*	*(SE+TI)/(TA+CC)*
7	***N**etworking **C**apital/**T**otal **A**ssets*	*NC/TA*
8	*(**S**alary And **E**mployees' **B**enefits + **R**eserve For **R**etirement)/No. Of **P**ersonnel*	*(SEB+RR)/P*
9	***L**iquid **A**ssets/(**D**eposits + **N**on-Deposit **F**unds)*	*LA/(D+NF)*
10	***I**nterest **E**xpenses/**T**otal **E**xpenses*	*IE/TE*
11	***L**iquid **A**ssets/**T**otal **A**ssets*	*LA/TA*
12	***S**tandard **C**apital **R**atio*	*SCR*

Table C.3: Financial Ratios of US Banks Dataset

#	***Predictor Variable Name***	
1	***W**orking **C**apital/**T**otal **A**ssets*	*WC/TA*
2	***R**etained **E**arnings/ **T**otal **A**ssets*	*RE/TA*
3	***E**arnings Before **I**nterest And **T**axes/ **T**otal **A**ssets*	*EIT/TA*
4	***M**arket Value Of **E**quity/ **T**otal **A**ssets*	*ME/TA*
5	***S**ales/ **T**otal **A**ssets*	*S/TA*

Table C.4: Financial Ratios of UK banks dataset

#	***Predictor Variable Name***	
1	***S**ales*	*Sales*
2	***P**rofit **B**efore **T**ax/**C**apital **E**mployed (%)*	*PBT/CE*
3	***F**unds **F**low/**T**otal **L**iabilities*	*FF/TL*
4	*(**C**urrent **L**iabilities + **L**ong **T**erm **D**ebits)/**T**otal **A**ssets*	*(CL+LTD)/TA*
5	***C**urrent **L**iabilities/**T**otal **A**ssets*	*CL/TA*
6	***C**urrent **A**ssets/**C**urrent **L**iabilities*	*CA/CL*
7	***C**urrent **A**ssets-**S**tock/**C**urrent **L**iabilities*	*CA-S/CL*
8	***C**urrent **A**ssets-**C**urrent **L**iabilities/**T**otal **A**ssets*	*CA-CL/TA*
9	***LAG**(Number of days between account year end and the date of annual report)*	*LAG*
10	***Age***	*Age*

Table C.5: Feature description of Auto MPG dataset

#	**Feature Name**	**Feature Type**
1	Cylinders	Multi-valued Discrete
2	Displacement	Continuous
3	Horsepower	Continuous
4	Weight	Continuous
5	Acceleration	Continuous
6	Model year	Multi-valued Discrete
7	Origin	Multi-valued Discrete
8	**Miles Per Gallon**	**(TARGET)**

Table C.6: Feature description of Body Fat dataset

#	Feature Name	Feature Type
1	Density determined from underwater weighing	Continuous
2	Age (years)	Multi Valued Discrete
3	Weight (lbs)	Continuous
4	Height (inches)	Continuous
5	Neck circumference (cm)	Continuous
6	Chest circumference (cm)	Continuous
7	Abdomen 2 circumference (cm)	Continuous
8	Hip circumference (cm)	Continuous
9	Thigh circumference (cm)	Continuous
10	Knee circumference (cm)	Continuous
11	Ankle circumference (cm)	Continuous
12	Biceps (extended) circumference (cm)	Continuous
13	Forearm circumference (cm)	Continuous
14	Wrist circumference (cm)	Continuous
15	**Percent body fat from Siri's (1956) equation**	**(TARGET)**

Table C.7: Feature description of Boston Housing dataset

#	Feature Name	Feature Type
1	CRIM: per capita crime rate by town	Continuous
2	ZN: proportion of residential land zoned for lots over 25,000 sq.ft.	Continuous
3	INDUS: proportion of non-retail business acres per town	Continuous
4	CHAS: Charles River dummy variable	Binary
5	NOX: nitric oxides concentration (parts per 10 million)	Continuous
6	RM: average number of rooms per dwelling	Continuous
7	AGE: proportion of owner-occupied units built prior to 1940	Continuous
8	DIS: weighted distances to five Boston employment centres	Continuous
9	RAD: index of accessibility to radial highways	Continuous
10	TAX: full-value property-tax rate per $10,000	Continuous
11	PTRATIO: pupil-teacher ratio by town	Continuous
12	B: 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town	Continuous
13	LSTAT: % lower status of the population	Continuous
14	**MEDV: Median value of owner-occupied homes in $1000's**	**(TARGET)**

Table C.8: Feature description of Forest Fires dataset

#	Feature Name	Feature Type
1	X - x-axis spatial coordinate within the Montesinho park map: 1 to 9	Multi Valued Discrete
2	Y - y-axis spatial coordinate within the Montesinho park map: 2 to 9	Multi Valued Discrete
3	Month	Multi Valued Discrete
4	Day	Multi Valued Discrete
5	FFMC - Fine Fuel Moisture Code	Continuous
6	DMC - Duff Moisture Code	Continuous
7	DC - Drought Code	Continuous
8	ISI - Initial Spread Index	Continuous
9	Temperature	Continuous
10	RH - relative humidity	Continuous
11	wind - wind speed in km/h	Continuous
12	rain - outside rain in mm/m2	Continuous
13	**area - the burned area of the forest (in ha): 0.00 to 1090.84**	**Continuous (TARGET)**

Table C.9: Feature description of Pollution dataset

#	Feature Name	Feature Type
1	PREC Average annual precipitation in inches	Continuous
2	JANT Average January temperature in degrees F	Continuous
3	JULT Average July temperature in degrees F	Continuous
4	OVR65 % of 1960 SMSA population aged 65 or older	Continuous
5	POPN Average household size	Continuous
6	EDUC Median school years completed by those over 22	Continuous
7	HOUS % of housing units which are sound and with all facilities	Continuous
8	DENS Population per sq. mile in urbanized areas, 1960	Continuous
9	NONW % non-white population in urbanized areas, 1960	Continuous
10	WWDRK % employed in white collar occupations	Continuous
11	POOR % of families with income < $3000	Continuous
12	HC Relative hydrocarbon pollution potential	Continuous
13	NOX Same for nitric oxides	Continuous
14	SO@ Same for sulphur dioxide	Continuous
15	HUMID Annual average % relative humidity at 1pm	Continuous
16	**MORT Total age-adjusted mortality rate per 100,000**	**(TARGET)**

Table C.10: Feature description of churn prediction dataset

#	Feature	Description	Value
	Target	Target Variable	0-NonChurner 1-Churner
1	CRED_T	Credit in month T	Positive real number
2	CRED_T-1	Credit in month T-1	Positive real number
3	CRED_T-2	Credit in month T-2	Positive real number
4	NCC_T	Number of credit cards in months T	Positive integer value
5	NCC_T-1	Number of credit cards in months T-1	Positive integer value
6	NCC_T-2	Number of credit cards in months T-2	Positive integer value
7	INCOME	Customer's Income	Positive real number
8	N_EDUC	Customer's educational level	1 - University student 2 - Medium degree 3 - Technical degree 4 - University degree
9	AGE	Customer's age	Positive integer
10	SX	Customers sex	1 – male 0 - Female
11	E_CIV	Civilian status	1-Single 2-Married 3-Widow 4-Divorced
12	T_WEB_T	Number of web transaction in months T	Positive integer
13	T_WEB_T-1	Number of web transaction in months T-1	Positive integer
14	T_WEB_T-2	Number of web transaction in months T-2	Positive integer
15	MAR_T	Customer's margin for the company in months T	Real Number
16	MAR_T-1	Customer's margin for the company in months T-1	Real Number
17	MAR_T-2	Customer's margin for the company in months T-2	Real Number
18	MAR_T-3	Customer's margin for the company in months T-3	Real Number
19	MAR_T-4	Customer's margin for the company in months T-4	Real Number
20	MAR_T-5	Customer's margin for the company in months T-5	Real Number
21	MAR_T-6	Customer's margin for the company in months T-6	Real Number

Appendix D:

Rules Tables

In this section larger rule sets extraction during the Results and discussions are shown.

Table D.1: Fuzzy Rules Extracted using Wine Dataset

#	Antecedents	Consequent
1	If "Flavanoids is High"	A
2	If "Alcohol is Low" and "Flavanoids is Medium"	B
3	If "Alcohol is Low"	B
4	If "Flavanoids is High" and " Hue is Medium"	A
5	If "Color Intensity is Low"	B
6	If "Color Intensity is High"	C
7	If "Alcohol is Low" and " Color Intensity is Low"	B
8	If "Alcohol is Medium" and "Color Intensity is High"	C
9	If "Flavanoids is Medium" and "Color Intensity is Low"	B
10	If "Total Phenols is Low" and "Proanthocyaninsis Low" and "Color Intensity is Medium"	C
11	If "Malic Acid is Medium" and "Total Phenols is Low" and "Proanthocyanins is Low"	C
12	If "Alcohol is Low" and "Hue is High"	B
13	If "Magnesium is High" and "Od280/Od315 Of Diluted Wines is High"	B
14	If "Alcohol is Low" and "Od280/Od315 Of Diluted Wines is Medium"	B
15	If "Flavanoids is High" and "Od280/Od315 Of Diluted Wines is High"	A
16	If "Malic Acid is Low" and "Magnesium is Low" and "Flavanoids is Medium"	B
17	If "Alcalinity Of Ash is Low" and "Flavanoids is High"	A
18	If "Alcalinity Of Ash is High" and "Color Intensity is Low"	B

Table D.2: Rule set extracted using SVR+CART (*Case-P*)
for *Auto MPG* dataset (3 features)

#	Antecedents	Prediction
1	If Displacement <= 0.114987 and Model Layer <= 0.625 and Weight <= 0.12617	0.55459
2	If Displacement <= 0.114987 and Model Layer <= 0.625 and Weight > 0.12617	0.486769
3	If Displacement <= 0.114987 and Model Layer > 0.625 and Weight <= 0.147292	0.64483
4	If Displacement <= 0.114987 and Model Layer > 0.625 and Weight > 0.147292	0.595032
5	If Displacement > 0.114987 and Displacement <= 0.373385 and Weight <= 0.367734 and Model Layer <= 0.458333	0.418954
6	If Displacement > 0.114987 and Displacement <= 0.373385 and Weight <= 0.367734 and Model Layer > 0.458333 and Model Year <= 0.791667	0.467772
7	If Displacement > 0.114987 and Displacement <= 0.373385 and Model Layer <= 0.791667 and Weight > 0.367734	0.357201
8	If Displacement > 0.114987 and Displacement <= 0.373385 and Model Layer > 0.791667 and Weight <= 0.345052	0.555221
9	If Displacement > 0.114987 and Displacement <= 0.373385 and Model Layer > 0.791667 and Weight > 0.345052	0.462763
10	If Displacement > 0.373385 and Weight <= 0.515453 and Model Layer <= 0.958333	0.303012
11	If Displacement > 0.373385 and Weight <= 0.515453 and Model Layer > 0.958333	0.459805
12	If Displacement > 0.373385 and Weight > 0.515453 and Weight <= 0.665012 and Model Layer <= 0.333333	0.180468
13	If Displacement > 0.373385 and Weight > 0.515453 and Weight <= 0.665012 and Model Layer > 0.333333	0.240821
14	If Displacement > 0.373385 and Weight > 0.665012 and Model Layer <= 0.458333	0.096366
15	If Displacement > 0.373385 and Weight > 0.665012 and Model Layer > 0.458333	0.168647

Table D.3: Rule set extracted using SVR+CART (*Case-P*)
for *Boston Housing* dataset (all features)

#	Antecedents	Predictions
1	If PTRATIO <= 0.75 and RM <= 0.551638 and LSTAT <= 0.158527	0.460798
2	If PTRATIO <= 0.75 and RM <= 0.551638 and LSTAT > 0.158527 and LSTAT <= 0.269316	0.407351
3	If LSTAT <= 0.269316 and PTRATIO <= 0.75 and RM > 0.551638 and RM <= 0.617072	0.494458
4	If LSTAT <= 0.269316 and RM <= 0.617072 and PTRATIO > 0.75 and B <= 0.908026	0.224296
5	If LSTAT <= 0.269316 and RM <= 0.617072 and PTRATIO > 0.75 and B > 0.908026	0.367576
6	If LSTAT <= 0.269316 and RM > 0.617072 and RM <= 0.662771	0.541152
7	If LSTAT <= 0.269316 and RM > 0.662771 and RM <= 0.742671	0.594396
8	If LSTAT <= 0.269316 and RM > 0.742671	0.677624
9	If LSTAT > 0.269316 and LSTAT <= 0.499448 and PTRATIO <= 0.771276 and RM <= 0.591205	0.340127
10	If LSTAT > 0.269316 and LSTAT <= 0.499448 and PTRATIO <= 0.771276 and RM > 0.591205	0.552586
11	If LSTAT > 0.269316 and LSTAT <= 0.499448 and PTRATIO > 0.771276 and B <= 0.936482	0.222108
12	If PTRATIO > 0.771276 and B > 0.936482 and LSTAT > 0.269316 and LSTAT <= 0.355685	0.319963
13	If PTRATIO > 0.771276 and B > 0.936482 and LSTAT > 0.355685 and LSTAT <= 0.499448	0.264505
14	If LSTAT > 0.499448 and LSTAT <= 0.769454 and CRIM <= 0.269919 and CHAS <= 0.5	0.193346
15	If LSTAT > 0.499448 and LSTAT <= 0.769454 and CRIM <= 0.269919 and CHAS > 0.5	0.286191
16	If LSTAT > 0.499448 and LSTAT <= 0.769454 and CRIM > 0.269919	0.077681
17	If LSTAT > 0.769454 and RM <= 0.228588	-0.0048
18	If LSTAT > 0.769454 and RM > 0.228588	0.08882

Table D.4: Rule set extracted using SVR+CART (*Case-P*)
for *Forest Fires* dataset (7 features)

#	Antecedents	Prediction
1	If X-Axis <= 0.6875 and Month <= 0.681818 and Wind <= 0.077778	0.00328504
2	If Month <= 0.681818 and X-Axis <= 0.1875 and RH <= 0.705882 and Wind > 0.077778 and Wind <= 0.677778 and DC <= 0.111528	0.00267747
3	If Month <= 0.681818 and X-Axis <= 0.1875 and Wind > 0.077778 and Wind <= 0.677778 and DC > 0.111528 and DC <= 0.755365 and RH <= 0.329412 and Day <= 0.0833335	0.00300294
4	If Month <= 0.681818 and X-Axis <= 0.1875 and Wind > 0.077778 and Wind <= 0.677778 and RH <= 0.329412 and Day > 0.0833335 and DC > 0.111528 and DC <= 0.69532	0.00287374
5	If Month <= 0.681818 and X-Axis <= 0.1875 and Wind > 0.077778 and Wind <= 0.677778 and RH <= 0.329412 and Day > 0.0833335 and DC > 0.69532 and DC <= 0.755365	0.0027755
6	If Month <= 0.681818 and X-Axis <= 0.1875 and DC > 0.111528 and DC <= 0.755365 and RH > 0.329412 and RH <= 0.705882 and Wind > 0.077778 and Wind <= 0.277778	0.00304313
7	If Month <= 0.681818 and X-Axis <= 0.1875 and DC > 0.111528 and DC <= 0.755365 and RH > 0.329412 and RH <= 0.705882 and Wind > 0.277778 and Wind <= 0.677778	0.00287807
8	If Month <= 0.681818 and X-Axis <= 0.1875 and DC <= 0.755365 and RH <= 0.705882 and Wind > 0.677778 and Wind <= 0.95	0.00321315
9	If Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.95 and X-Axis <= 0.1875 and DC <= 0.755365 and RH > 0.705882	0.00326583
10	If Month <= 0.681818 and X-Axis <= 0.1875 and DC > 0.755365 and Day <= 0.75 and Wind > 0.077778 and Wind <= 0.7	0.0027067
11	If Month <= 0.681818 and X-Axis <= 0.1875 and DC > 0.755365 and Day <= 0.75 and Wind > 0.7 and Wind <= 0.95	0.0028798
12	If Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.95 and X-Axis <= 0.1875 and DC > 0.755365 and Day > 0.75	0.00258712
13	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month <= 0.590909 and Wind > 0.077778 and Wind <= 0.205556	0.002902
14	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month <= 0.590909 and Wind > 0.205556 and Wind <= 0.427778 and DC <= 0.544213	0.0026695
15	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month <= 0.590909 and Wind > 0.205556 and Wind <= 0.427778 and DC > 0.544213	0.00249425
16	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Wind > 0.427778 and Wind <= 0.95 and Month <= 0.318181 and Day <= 0.666667	0.00258895
17	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Wind > 0.427778 and Wind <= 0.95 and Month <= 0.318181 and Day > 0.666667	0.00246797
18	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Wind > 0.427778 and Wind <= 0.95 and Month > 0.318181 and Month <= 0.590909	0.00266863

19	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and RH <= 0.770589 and DC <= 0.702591	0.00285978
20	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and RH <= 0.770589 and DC > 0.702591 and DC <= 0.706813	0.0027838
21	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and DC > 0.706813 and DC <= 0.83734 and RH <= 0.347059	0.00273606
22	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and DC > 0.706813 and DC <= 0.83734 and RH > 0.347059 and RH <= 0.770589	0.0026857
23	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and DC <= 0.83734 and RH > 0.770589	0.0025275
24	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.077778 and Wind <= 0.577778 and DC > 0.83734	0.0024753
25	If X-Axis > 0.1875 and X-Axis <= 0.4375 and Month > 0.590909 and Month <= 0.681818 and Wind > 0.577778 and Wind <= 0.95	0.00306913

Table D.5: Rules Set using SVR + CART for *Auto MPG* dataset

#	**Antecedents**	**Prediction**
01	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN <= 0.75 and MODEL_YEAR <= 0.541667	0.473147
02	if WEIGHT <= 0.415226 and ORIGIN <= 0.75 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.875 and HORSEPOWER <= 0.0380435	0.636408
03	if WEIGHT <= 0.415226 and ORIGIN <= 0.75 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.875 and HORSEPOWER > 0.0380435 and HORSEPOWER <= 0.14673	0.529263
04	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN <= 0.75 and MODEL_YEAR > 0.875	0.619481
05	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN > 0.75 and MODEL_YEAR <= 0.875	0.618746
06	if WEIGHT <= 0.415226 and HORSEPOWER <= 0.146739 and ORIGIN > 0.75 and MODEL_YEAR > 0.875	0.760563
07	if HORSEPOWER > 0.146739 and WEIGHT <= 0.274171 and MODEL_YEAR <= 0.541667 and DISPLACEMENT <= 0.130491	0.442948
08	if HORSEPOWER > 0.146739 and WEIGHT <= 0.274171 and MODEL_YEAR <= 0.541667 and DISPLACEMENT > 0.130491	0.406843

09	if HORSEPOWER > 0.146739 and WEIGHT <= 0.274171 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.791667 and ACCELERATION <= 0.538691	0.472736
10	if HORSEPOWER > 0.146739 and WEIGHT <= 0.274171 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.791667 and ACCELERATION > 0.538691	0.547625
11	if HORSEPOWER > 0.146739 and WEIGHT > 0.274171 and WEIGHT <= 0.415226 and MODEL_YEAR <= 0.541667	0.35487
12	if HORSEPOWER > 0.146739 and WEIGHT > 0.274171 and WEIGHT <= 0.415226 and MODEL_YEAR > 0.541667 and MODEL_YEAR <= 0.791667	0.409262
13	if HORSEPOWER > 0.146739 and MODEL_YEAR > 0.791667 and WEIGHT <= 0.200312	0.632958
14	if HORSEPOWER > 0.146739 and MODEL_YEAR > 0.791667 and WEIGHT > 0.200312 and WEIGHT <= 0.296569	0.536736
15	if HORSEPOWER > 0.146739 and MODEL_YEAR > 0.791667 and WEIGHT > 0.296569 and WEIGHT <= 0.415226	0.488202
16	if MODEL_YEAR <= 0.875 and WEIGHT > 0.415226 and WEIGHT <= 0.505387 and HORSEPOWER <= 0.138587	0.419966
17	if MODEL_YEAR <= 0.875 and WEIGHT > 0.415226 and WEIGHT <= 0.505387 and HORSEPOWER > 0.138587 and DISPLACEMENT <= 0.447028	0.331893
18	if MODEL_YEAR <= 0.875 and WEIGHT > 0.415226 and WEIGHT <= 0.505387 and HORSEPOWER > 0.138587 and DISPLACEMENT > 0.447028	0.298214
19	if MODEL_YEAR <= 0.875 and WEIGHT > 0.505387 and WEIGHT <= 0.604479 and ACCELERATION <= 0.747024 and HORSEPOWER <= 0.453804	0.304063
20	if MODEL_YEAR <= 0.875 and WEIGHT > 0.505387 and WEIGHT <= 0.604479 and ACCELERATION <= 0.747024 and HORSEPOWER > 0.453804	0.236424
21	if MODEL_YEAR <= 0.875 and WEIGHT > 0.505387 and WEIGHT <= 0.604479 and ACCELERATION > 0.747024	0.196165
22	if WEIGHT > 0.415226 and WEIGHT <= 0.604479 and MODEL_YEAR > 0.875	0.477826
23	if WEIGHT > 0.604479 and WEIGHT <= 0.693082 and ACCELERATION <= 0.494047	0.183192
24	if WEIGHT > 0.604479 and WEIGHT <= 0.693082 and ACCELERATION > 0.494047	0.104375
25	if WEIGHT > 0.693082 and WEIGHT <= 0.870286	0.117609
26	if WEIGHT > 0.870286	-0.0874944

Table D.6: Rules Set using SVR+DENFIS for *Body Fat* dataset

#	Antecedents	Predictions
01	if X1 is GMF(0.50,0.34) and X2 is GMF(0.50,0.71) and X3 is GMF(0.50,0.49) and X4 is GMF(0.50,0.73) and X5 is GMF(0.50,0.68) and X6 is GMF(0.50,0.55) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.39) and X9 is GMF(0.50,0.45) and X10 is GMF(0.50,0.44) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.73) and X13 is GMF(0.50,0.67) and X14 is GMF(0.50,0.53)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
02	if X1 is GMF(0.50,0.35) and X2 is GMF(0.50,0.23) and X3 is GMF(0.50,0.53) and X4 is GMF(0.50,0.83) and X5 is GMF(0.50,0.50) and X6 is GMF(0.50,0.46) and X7 is GMF(0.50,0.48) and X8 is GMF(0.50,0.50) and X9 is GMF(0.50,0.63) and X10 is GMF(0.50,0.67) and X11 is GMF(0.50,0.24) and X12 is GMF(0.50,0.75) and X13 is GMF(0.50,0.69) and X14 is GMF(0.50,0.19)	
03	if X1 is GMF(0.50,0.78) and X2 is GMF(0.50,0.55) and X3 is GMF(0.50,0.20) and X4 is GMF(0.50,0.76) and X5 is GMF(0.50,0.17) and X6 is GMF(0.50,0.29) and X7 is GMF(0.50,0.26) and X8 is GMF(0.50,0.20) and X9 is GMF(0.50,0.29) and X10 is GMF(0.50,0.19) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.28) and X13 is GMF(0.50,0.42) and X14 is GMF(0.50,0.14)	
04	if X1 is GMF(0.50,0.70) and X2 is GMF(0.50,0.18) and X3 is GMF(0.50,0.25) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.15) and X6 is GMF(0.50,0.32) and X7 is GMF(0.50,0.23) and X8 is GMF(0.50,0.19) and X9 is GMF(0.50,0.22) and X10 is GMF(0.50,0.31) and X11 is GMF(0.50,0.10) and X12 is GMF(0.50,0.11) and X13 is GMF(0.50,0.39) and X14 is GMF(0.50,0.05)	
05	if X1 is GMF(0.50,0.54) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.48) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.42) and X6 is GMF(0.50,0.46) and X7 is GMF(0.50,0.34) and X8 is GMF(0.50,0.40) and X9 is GMF(0.50,0.42) and X10 is GMF(0.50,0.53) and X11 is GMF(0.50,0.33) and X12 is GMF(0.50,0.47) and X13 is GMF(0.50,0.54) and X14 is GMF(0.50,0.32)	
06	if X1 is GMF(0.50,0.40) and X2 is GMF(0.50,0.48) and X3 is GMF(0.50,0.29) and X4 is GMF(0.50,0.75) and X5 is GMF(0.50,0.41) and X6 is GMF(0.50,0.44) and X7 is GMF(0.50,0.37) and X8 is GMF(0.50,0.21) and X9 is GMF(0.50,0.34) and X10 is GMF(0.50,0.22) and X11 is GMF(0.50,0.07) and X12 is GMF(0.50,0.57) and X13 is GMF(0.50,0.59) and X14 is GMF(0.50,0.19)	
07	if X1 is GMF(0.50,0.50) and X2 is GMF(0.50,0.25) and X3 is GMF(0.50,0.72) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.95) and X6 is GMF(0.50,0.58) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.53) and X9 is GMF(0.50,0.58) and X10 is GMF(0.50,0.74) and X11 is GMF(0.50,0.34) and X12 is GMF(0.50,0.82) and X13 is GMF(0.50,0.84) and X14 is GMF(0.50,0.66)	

08	if X1 is GMF(0.50,0.49) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.68) and X4 is GMF(0.50,0.83) and X5 is GMF(0.50,0.59) and X6 is GMF(0.50,0.65) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.57) and X9 is GMF(0.50,0.71) and X10 is GMF(0.50,0.95) and X11 is GMF(0.50,0.38) and X12 is GMF(0.50,0.84) and X13 is GMF(0.50,0.73) and X14 is GMF(0.50,0.45)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
09	if X1 is GMF(0.50,0.88) and X2 is GMF(0.50,0.52) and X3 is GMF(0.50,0.71) and X4 is GMF(0.50,0.95) and X5 is GMF(0.50,0.70) and X6 is GMF(0.50,0.67) and X7 is GMF(0.50,0.52) and X8 is GMF(0.50,0.55) and X9 is GMF(0.50,0.53) and X10 is GMF(0.50,0.79) and X11 is GMF(0.50,0.24) and X12 is GMF(0.50,0.52) and X13 is GMF(0.50,0.69) and X14 is GMF(0.50,0.77)	
10	if X1 is GMF(0.50,0.77) and X2 is GMF(0.50,0.05) and X3 is GMF(0.50,0.39) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.31) and X7 is GMF(0.50,0.26) and X8 is GMF(0.50,0.35) and X9 is GMF(0.50,0.43) and X10 is GMF(0.50,0.37) and X11 is GMF(0.50,0.26) and X12 is GMF(0.50,0.36) and X13 is GMF(0.50,0.56) and X14 is GMF(0.50,0.36	
11	if X1 is GMF(0.50,0.52) and X2 is GMF(0.50,0.08) and X3 is GMF(0.50,0.61) and X4 is GMF(0.50,0.86) and X5 is GMF(0.50,0.53) and X6 is GMF(0.50,0.47) and X7 is GMF(0.50,0.52) and X8 is GMF(0.50,0.61) and X9 is GMF(0.50,0.85) and X10 is GMF(0.50,0.90) and X11 is GMF(0.50,0.38) and X12 is GMF(0.50,0.75) and X13 is GMF(0.50,0.65) and X14 is GMF(0.50,0.45)	
12	if X1 is GMF(0.50,0.49) and X2 is GMF(0.50,0.25) and X3 is GMF(0.50,0.32) and X4 is GMF(0.50,0.73) and X5 is GMF(0.50,0.28) and X6 is GMF(0.50,0.41) and X7 is GMF(0.50,0.42) and X8 is GMF(0.50,0.36) and X9 is GMF(0.50,0.58) and X10 is GMF(0.50,0.49) and X11 is GMF(0.50,0.14) and X12 is GMF(0.50,0.40) and X13 is GMF(0.50,0.40) and X14 is GMF(0.50,0.12)	
13	if X1 is GMF(0.50,0.41) and X2 is GMF(0.50,0.32) and X3 is GMF(0.50,0.50) and X4 is GMF(0.50,0.88) and X5 is GMF(0.50,0.45) and X6 is GMF(0.50,0.34) and X7 is GMF(0.50,0.41) and X8 is GMF(0.50,0.48) and X9 is GMF(0.50,0.63) and X10 is GMF(0.50,0.69) and X11 is GMF(0.50,0.35) and X12 is GMF(0.50,0.57) and X13 is GMF(0.50,0.60) and X14 is GMF(0.50,0.42)	
14	if X1 is GMF(0.50,0.34) and X2 is GMF(0.50,0.28) and X3 is GMF(0.50,0.82) and X4 is GMF(0.50,0.84) and X5 is GMF(0.50,0.79) and X6 is GMF(0.50,0.79) and X7 is GMF(0.50,0.70) and X8 is GMF(0.50,0.69) and X9 is GMF(0.50,0.80) and X10 is GMF(0.50,0.82) and X11 is GMF(0.50,0.35) and X12 is GMF(0.50,0.63) and X13 is GMF(0.50,0.60) and X14 is GMF(0.50,0.40)	

15	if X1 is GMF(0.50,0.44) and X2 is GMF(0.50,0.66) and X3 is GMF(0.50,0.35) and X4 is GMF(0.50,0.84) and X5 is GMF(0.50,0.20) and X6 is GMF(0.50,0.38) and X7 is GMF(0.50,0.40) and X8 is GMF(0.50,0.35) and X9 is GMF(0.50,0.36) and X10 is GMF(0.50,0.48) and X11 is GMF(0.50,0.19) and X12 is GMF(0.50,0.43) and X13 is GMF(0.50,0.46) and X14 is GMF(0.50,0.44)	
16	if X1 is GMF(0.50,0.68) and X2 is GMF(0.50,0.55) and X3 is GMF(0.50,0.43) and X4 is GMF(0.50,0.78) and X5 is GMF(0.50,0.70) and X6 is GMF(0.50,0.56) and X7 is GMF(0.50,0.46) and X8 is GMF(0.50,0.34) and X9 is GMF(0.50,0.39) and X10 is GMF(0.50,0.35) and X11 is GMF(0.50,0.15) and X12 is GMF(0.50,0.61) and X13 is GMF(0.50,0.71) and X14 is GMF(0.50,0.55)	
17	if X1 is GMF(0.50,0.17) and X2 is GMF(0.50,0.46) and X3 is GMF(0.50,0.51) and X4 is GMF(0.50,0.71) and X5 is GMF(0.50,0.46) and X6 is GMF(0.50,0.77) and X7 is GMF(0.50,0.74) and X8 is GMF(0.50,0.69) and X9 is GMF(0.50,0.54) and X10 is GMF(0.50,0.45) and X11 is GMF(0.50,0.16) and X12 is GMF(0.50,0.46) and X13 is GMF(0.50,0.62) and X14 is GMF(0.50,0.14)	
18	if X1 is GMF(0.50,0.50) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.63) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.57) and X7 is GMF(0.50,0.52) and X8 is GMF(0.50,0.47) and X9 is GMF(0.50,0.59) and X10 is GMF(0.50,0.58) and X11 is GMF(0.50,0.26) and X12 is GMF(0.50,0.77) and X13 is GMF(0.50,0.66) and X14 is GMF(0.50,0.53)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
19	if X1 is GMF(0.50,0.39) and X2 is GMF(0.50,0.69) and X3 is GMF(0.50,0.25) and X4 is GMF(0.50,0.76) and X5 is GMF(0.50,0.43) and X6 is GMF(0.50,0.38) and X7 is GMF(0.50,0.36) and X8 is GMF(0.50,0.31) and X9 is GMF(0.50,0.30) and X10 is GMF(0.50,0.43) and X11 is GMF(0.50,0.16) and X12 is GMF(0.50,0.32) and X13 is GMF(0.50,0.32) and X14 is GMF(0.50,0.27)	
20	if X1 is GMF(0.50,0.31) and X2 is GMF(0.50,0.14) and X3 is GMF(0.50,0.60) and X4 is GMF(0.50,0.79) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.53) and X7 is GMF(0.50,0.62) and X8 is GMF(0.50,0.52) and X9 is GMF(0.50,0.76) and X10 is GMF(0.50,0.60) and X11 is GMF(0.50,0.38) and X12 is GMF(0.50,0.68) and X13 is GMF(0.50,0.68) and X14 is GMF(0.50,0.53)	
21	if X1 is GMF(0.50,0.23) and X2 is GMF(0.50,0.25) and X3 is GMF(0.50,0.74) and X4 is GMF(0.50,0.80) and X5 is GMF(0.50,0.64) and X6 is GMF(0.50,0.71) and X7 is GMF(0.50,0.79) and X8 is GMF(0.50,0.65) and X9 is GMF(0.50,0.95) and X10 is GMF(0.50,0.65) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.75) and X13 is GMF(0.50,0.75) and X14 is GMF(0.50,0.47)	

22	if X1 is GMF(0.50,0.72) and X2 is GMF(0.50,0.58) and X3 is GMF(0.50,0.32) and X4 is GMF(0.50,0.80) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.27) and X7 is GMF(0.50,0.20) and X8 is GMF(0.50,0.25) and X9 is GMF(0.50,0.36) and X10 is GMF(0.50,0.57) and X11 is GMF(0.50,0.31) and X12 is GMF(0.50,0.34) and X13 is GMF(0.50,0.58) and X14 is GMF(0.50,0.34)	
23	if X1 is GMF(0.50,0.42) and X2 is GMF(0.50,0.26) and X3 is GMF(0.50,0.73) and X4 is GMF(0.50,0.84) and X5 is GMF(0.50,0.74) and X6 is GMF(0.50,0.71) and X7 is GMF(0.50,0.68) and X8 is GMF(0.50,0.70) and X9 is GMF(0.50,0.78) and X10 is GMF(0.50,0.80) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.70) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.71)	
24	if X1 is GMF(0.50,0.36) and X2 is GMF(0.50,0.08) and X3 is GMF(0.50,0.46) and X4 is GMF(0.50,0.83) and X5 is GMF(0.50,0.10) and X6 is GMF(0.50,0.38) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.42) and X9 is GMF(0.50,0.58) and X10 is GMF(0.50,0.78) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.48) and X13 is GMF(0.50,0.48) and X14 is GMF(0.50,0.27)	
25	if X1 is GMF(0.50,0.54) and X2 is GMF(0.50,0.46) and X3 is GMF(0.50,0.38) and X4 is GMF(0.50,0.84) and X5 is GMF(0.50,0.20) and X6 is GMF(0.50,0.39) and X7 is GMF(0.50,0.38) and X8 is GMF(0.50,0.29) and X9 is GMF(0.50,0.37) and X10 is GMF(0.50,0.49) and X11 is GMF(0.50,0.24) and X12 is GMF(0.50,0.15) and X13 is GMF(0.50,0.41) and X14 is GMF(0.50,0.25)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
26	if X1 is GMF(0.50,0.95) and X2 is GMF(0.50,0.32) and X3 is GMF(0.50,0.05) and X4 is GMF(0.50,0.77) and X5 is GMF(0.50,0.05) and X6 is GMF(0.50,0.05) and X7 is GMF(0.50,0.05) and X8 is GMF(0.50,0.05) and X9 is GMF(0.50,0.05) and X10 is GMF(0.50,0.05) and X11 is GMF(0.50,0.05) and X12 is GMF(0.50,0.16) and X13 is GMF(0.50,0.28) and X14 is GMF(0.50,0.05)	
27	if X1 is GMF(0.50,0.31) and X2 is GMF(0.50,0.54) and X3 is GMF(0.50,0.57) and X4 is GMF(0.50,0.82) and X5 is GMF(0.50,0.65) and X6 is GMF(0.50,0.72) and X7 is GMF(0.50,0.62) and X8 is GMF(0.50,0.31) and X9 is GMF(0.50,0.44) and X10 is GMF(0.50,0.43) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.43) and X13 is GMF(0.50,0.56) and X14 is GMF(0.50,0.36)	
28	if X1 is GMF(0.50,0.76) and X2 is GMF(0.50,0.46) and X3 is GMF(0.50,0.26) and X4 is GMF(0.50,0.87) and X5 is GMF(0.50,0.16) and X6 is GMF(0.50,0.31) and X7 is GMF(0.50,0.21) and X8 is GMF(0.50,0.19) and X9 is GMF(0.50,0.23) and X10 is GMF(0.50,0.39) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.91) and X13 is GMF(0.50,0.46) and X14 is GMF(0.50,0.42)	

29	if X1 is GMF(0.50,0.29) and X2 is GMF(0.50,0.48) and X3 is GMF(0.50,0.58) and X4 is GMF(0.50,0.75) and X5 is GMF(0.50,0.62) and X6 is GMF(0.50,0.70) and X7 is GMF(0.50,0.66) and X8 is GMF(0.50,0.44) and X9 is GMF(0.50,0.52) and X10 is GMF(0.50,0.69) and X11 is GMF(0.50,0.34) and X12 is GMF(0.50,0.61) and X13 is GMF(0.50,0.70) and X14 is GMF(0.50,0.38)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
30	if X1 is GMF(0.50,0.52) and X2 is GMF(0.50,0.11) and X3 is GMF(0.50,0.82) and X4 is GMF(0.50,0.89) and X5 is GMF(0.50,0.77) and X6 is GMF(0.50,0.58) and X7 is GMF(0.50,0.58) and X8 is GMF(0.50,0.70) and X9 is GMF(0.50,0.90) and X10 is GMF(0.50,0.90) and X11 is GMF(0.50,0.37) and X12 is GMF(0.50,0.91) and X13 is GMF(0.50,0.88) and X14 is GMF(0.50,0.62)	
31	if X1 is GMF(0.50,0.65) and X2 is GMF(0.50,0.17) and X3 is GMF(0.50,0.16) and X4 is GMF(0.50,0.78) and X5 is GMF(0.50,0.23) and X6 is GMF(0.50,0.22) and X7 is GMF(0.50,0.16) and X8 is GMF(0.50,0.15) and X9 is GMF(0.50,0.14) and X10 is GMF(0.50,0.16) and X11 is GMF(0.50,0.15) and X12 is GMF(0.50,0.11) and X13 is GMF(0.50,0.95) and X14 is GMF(0.50,0.12)	
32	if X1 is GMF(0.50,0.28) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.59) and X4 is GMF(0.50,0.05) and X5 is GMF(0.50,0.30) and X6 is GMF(0.50,0.54) and X7 is GMF(0.50,0.60) and X8 is GMF(0.50,0.73) and X9 is GMF(0.50,0.83) and X10 is GMF(0.50,0.81) and X11 is GMF(0.50,0.28) and X12 is GMF(0.50,0.57) and X13 is GMF(0.50,0.55) and X14 is GMF(0.50,0.21)	
33	if X1 is GMF(0.50,0.31) and X2 is GMF(0.50,0.45) and X3 is GMF(0.50,0.67) and X4 is GMF(0.50,0.81) and X5 is GMF(0.50,0.36) and X6 is GMF(0.50,0.68) and X7 is GMF(0.50,0.71) and X8 is GMF(0.50,0.70) and X9 is GMF(0.50,0.73) and X10 is GMF(0.50,0.67) and X11 is GMF(0.50,0.36) and X12 is GMF(0.50,0.79) and X13 is GMF(0.50,0.62) and X14 is GMF(0.50,0.40)	
34	if X1 is GMF(0.50,0.25) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.70) and X4 is GMF(0.50,0.80) and X5 is GMF(0.50,0.68) and X6 is GMF(0.50,0.83) and X7 is GMF(0.50,0.76) and X8 is GMF(0.50,0.54) and X9 is GMF(0.50,0.59) and X10 is GMF(0.50,0.62) and X11 is GMF(0.50,0.15) and X12 is GMF(0.50,0.66) and X13 is GMF(0.50,0.68) and X14 is GMF(0.50,0.21)	
35	if X1 is GMF(0.50,0.30) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.86) and X4 is GMF(0.50,0.87) and X5 is GMF(0.50,0.79) and X6 is GMF(0.50,0.74) and X7 is GMF(0.50,0.78) and X8 is GMF(0.50,0.74) and X9 is GMF(0.50,0.85) and X10 is GMF(0.50,0.88) and X11 is GMF(0.50,0.45) and X12 is GMF(0.50,0.83) and X13 is GMF(0.50,0.75) and X14 is GMF(0.50,0.64)	

36	if X1 is GMF(0.50,0.70) and X2 is GMF(0.50,0.11) and X3 is GMF(0.50,0.26) and X4 is GMF(0.50,0.79) and X5 is GMF(0.50,0.19) and X6 is GMF(0.50,0.30) and X7 is GMF(0.50,0.18) and X8 is GMF(0.50,0.24) and X9 is GMF(0.50,0.37) and X10 is GMF(0.50,0.25) and X11 is GMF(0.50,0.06) and X12 is GMF(0.50,0.43) and X13 is GMF(0.50,0.57) and X14 is GMF(0.50,0.29)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
37	if X1 is GMF(0.50,0.26) and X2 is GMF(0.50,0.40) and X3 is GMF(0.50,0.95) and X4 is GMF(0.50,0.78) and X5 is GMF(0.50,0.89) and X6 is GMF(0.50,0.95) and X7 is GMF(0.50,0.95) and X8 is GMF(0.50,0.95) and X9 is GMF(0.50,0.89) and X10 is GMF(0.50,0.56) and X11 is GMF(0.50,0.47) and X12 is GMF(0.50,0.77) and X13 is GMF(0.50,0.81) and X14 is GMF(0.50,0.95)	
38	if X1 is GMF(0.50,0.80) and X2 is GMF(0.50,0.09) and X3 is GMF(0.50,0.50) and X4 is GMF(0.50,0.88) and X5 is GMF(0.50,0.43) and X6 is GMF(0.50,0.45) and X7 is GMF(0.50,0.26) and X8 is GMF(0.50,0.38) and X9 is GMF(0.50,0.57) and X10 is GMF(0.50,0.45) and X11 is GMF(0.50,0.28) and X12 is GMF(0.50,0.73) and X13 is GMF(0.50,0.71) and X14 is GMF(0.50,0.36)	
39	if X1 is GMF(0.50,0.39) and X2 is GMF(0.50,0.79) and X3 is GMF(0.50,0.37) and X4 is GMF(0.50,0.81) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.46) and X7 is GMF(0.50,0.45) and X8 is GMF(0.50,0.27) and X9 is GMF(0.50,0.34) and X10 is GMF(0.50,0.30) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.41) and X13 is GMF(0.50,0.46) and X14 is GMF(0.50,0.55)	
40	if X1 is GMF(0.50,0.48) and X2 is GMF(0.50,0.25) and X3 is GMF(0.50,0.35) and X4 is GMF(0.50,0.77) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.41) and X7 is GMF(0.50,0.38) and X8 is GMF(0.50,0.28) and X9 is GMF(0.50,0.32) and X10 is GMF(0.50,0.10) and X11 is GMF(0.50,0.16) and X12 is GMF(0.50,0.34) and X13 is GMF(0.50,0.55) and X14 is GMF(0.50,0.27)	
41	if X1 is GMF(0.50,0.36) and X2 is GMF(0.50,0.48) and X3 is GMF(0.50,0.54) and X4 is GMF(0.50,0.78) and X5 is GMF(0.50,0.79) and X6 is GMF(0.50,0.53) and X7 is GMF(0.50,0.52) and X8 is GMF(0.50,0.49) and X9 is GMF(0.50,0.67) and X10 is GMF(0.50,0.73) and X11 is GMF(0.50,0.34) and X12 is GMF(0.50,0.55) and X13 is GMF(0.50,0.67) and X14 is GMF(0.50,0.58)	
42	if X1 is GMF(0.50,0.58) and X2 is GMF(0.50,0.43) and X3 is GMF(0.50,0.25) and X4 is GMF(0.50,0.75) and X5 is GMF(0.50,0.32) and X6 is GMF(0.50,0.32) and X7 is GMF(0.50,0.31) and X8 is GMF(0.50,0.27) and X9 is GMF(0.50,0.41) and X10 is GMF(0.50,0.30) and X11 is GMF(0.50,0.17) and X12 is GMF(0.50,0.36) and X13 is GMF(0.50,0.47) and X14 is GMF(0.50,0.25)	

43	if X1 is GMF(0.50,0.81) and X2 is GMF(0.50,0.35) and X3 is GMF(0.50,0.16) and X4 is GMF(0.50,0.76) and X5 is GMF(0.50,0.41) and X6 is GMF(0.50,0.20) and X7 is GMF(0.50,0.18) and X8 is GMF(0.50,0.13) and X9 is GMF(0.50,0.20) and X10 is GMF(0.50,0.16) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.16) and X13 is GMF(0.50,0.47) and X14 is GMF(0.50,0.42)	Y = 1.99 - 0.99 * X1 + 0.01 * X3 - 0.01 * X4 + 0.01 * X7 + 0.01 * X8 - 0.01 * X10
44	if X1 is GMF(0.50,0.50) and X2 is GMF(0.50,0.60) and X3 is GMF(0.50,0.44) and X4 is GMF(0.50,0.77) and X5 is GMF(0.50,0.52) and X6 is GMF(0.50,0.43) and X7 is GMF(0.50,0.53) and X8 is GMF(0.50,0.44) and X9 is GMF(0.50,0.54) and X10 is GMF(0.50,0.56) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.54) and X13 is GMF(0.50,0.52) and X14 is GMF(0.50,0.42)	
45	if X1 is GMF(0.50,0.66) and X2 is GMF(0.50,0.20) and X3 is GMF(0.50,0.45) and X4 is GMF(0.50,0.88) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.44) and X7 is GMF(0.50,0.35) and X8 is GMF(0.50,0.38) and X9 is GMF(0.50,0.39) and X10 is GMF(0.50,0.49) and X11 is GMF(0.50,0.95) and X12 is GMF(0.50,0.50) and X13 is GMF(0.50,0.48) and X14 is GMF(0.50,0.40)	
46	if X1 is GMF(0.50,0.05) and X2 is GMF(0.50,0.49) and X3 is GMF(0.50,0.68) and X4 is GMF(0.50,0.70) and X5 is GMF(0.50,0.71) and X6 is GMF(0.50,0.80) and X7 is GMF(0.50,0.89) and X8 is GMF(0.50,0.67) and X9 is GMF(0.50,0.56) and X10 is GMF(0.50,0.33) and X11 is GMF(0.50,0.27) and X12 is GMF(0.50,0.65) and X13 is GMF(0.50,0.58) and X14 is GMF(0.50,0.40)	

GMF(*x*, *y*) indicates Gaussian Membership function with mean *x* and variance *y*.

Table D.7: Rules Set using SVR+CART for *Boston Housing* dataset

#	**Antecedents**	**Prediction**
01	if CRIM <= 0.059321 and LSTAT <= 0.104305 and RM <= 0.620617	0.522726
02	if CRIM <= 0.059321 and LSTAT <= 0.104305 and RM > 0.620617 and RM <= 0.661142	0.580579
03	if CRIM <= 0.059321 and LSTAT > 0.104305 and LSTAT <= 0.164045 and RM <= 0.573864	0.45106
04	if CRIM <= 0.059321 and LSTAT > 0.104305 and LSTAT <= 0.164045 and RM > 0.573864 and RM <= 0.661142	0.505173
05	if LSTAT <= 0.164045 and RM <= 0.661142 and CRIM > 0.059321	0.826973
06	if LSTAT <= 0.164045 and CRIM <= 0.0449 and RM > 0.661142 and RM <= 0.704158	0.61574
07	if LSTAT <= 0.164045 and CRIM <= 0.0449 and RM > 0.704158 and RM <= 0.820272	0.71813
08	if LSTAT <= 0.164045 and RM > 0.661142 and RM <= 0.820272 and CRIM > 0.0449	0.900976

09	if LSTAT <= 0.164045 and RM > 0.820272 and RAD <= 0.217392	0.892593
10	if LSTAT <= 0.164045 and RM > 0.820272 and RAD > 0.217392	0.787818
11	if LSTAT > 0.164045 and LSTAT <= 0.357754 and PTRATIO <= 0.132978	0.639806
12	if PTRATIO > 0.132978 and LSTAT > 0.164045 and LSTAT <= 0.274835 and CHAS <= 0.5 and RM <= 0.48927	0.361998
13	if PTRATIO > 0.132978 and LSTAT > 0.164045 and LSTAT <= 0.274835 and CHAS <= 0.5 and RM > 0.48927 and TAX <= 0.271946	0.442761
14	if PTRATIO > 0.132978 and LSTAT > 0.164045 and LSTAT <= 0.274835 and CHAS <= 0.5 and RM > 0.48927 and TAX > 0.271946	0.398301
15	if PTRATIO > 0.132978 and LSTAT > 0.164045 and LSTAT <= 0.274835 and CHAS > 0.5	0.591983
16	if PTRATIO > 0.132978 and LSTAT > 0.274835 and LSTAT <= 0.357754 and CRIM <= 0.321786 and NOX <= 0.101749	0.216231
17	if LSTAT > 0.274835 and LSTAT <= 0.357754 and CRIM <= 0.321786 and NOX > 0.101749 and PTRATIO > 0.132978 and PTRATIO <= 0.888298	0.353425
18	if LSTAT > 0.274835 and LSTAT <= 0.357754 and CRIM <= 0.321786 and NOX > 0.101749 and PTRATIO > 0.888298	0.268955
19	if PTRATIO > 0.132978 and LSTAT > 0.274835 and LSTAT <= 0.357754 and CRIM > 0.321786	0.161555
20	if LSTAT > 0.357754 and LSTAT <= 0.450745 and CRIM <= 0.0012965	0.414246
21	if LSTAT > 0.357754 and LSTAT <= 0.450745 and CRIM > 0.0012965 and CRIM <= 0.117411	0.272778
22	if CRIM <= 0.117411 and LSTAT > 0.450745 and NOX <= 0.436214	0.240371
23	if LSTAT > 0.450745 and NOX > 0.436214 and CRIM <= 0.0221815	0.125006
24	if LSTAT > 0.450745 and NOX > 0.436214 and CRIM > 0.0221815 and CRIM <= 0.117411	0.196233
25	if LSTAT > 0.357754 and CRIM > 0.117411 and NOX <= 0.50926	0.224651
26	if CRIM > 0.117411 and NOX > 0.50926 and LSTAT > 0.357754 and LSTAT <= 0.415287	0.227456
27	if NOX > 0.50926 and LSTAT > 0.415287 and CRIM > 0.117411 and CRIM <= 0.157842	0.0778943
28	if NOX > 0.50926 and LSTAT > 0.415287 and CRIM > 0.157842 and CRIM <= 0.249894	0.15585
29	if NOX > 0.50926 and LSTAT > 0.415287 and CRIM > 0.249894	0.0781522

Table D.8: Rules Set using SVR+DENFIS for *Pollution* dataset

#	Antecedents	Predictions
01	if X1 is GMF(0.50,0.82) and X2 is GMF(0.50,0.59) and X3 is GMF(0.50,0.69) and X4 is GMF(0.50,0.35) and X5 is GMF(0.50,0.88) and X6 is GMF(0.50,0.39) and X7 is GMF(0.50,0.05) and X8 is GMF(0.50,0.25) and X9 is GMF(0.50,0.95) and X10 is GMF(0.50,0.34) and X11 is GMF(0.50,0.91) and X12 is GMF(0.50,0.09) and X13 is GMF(0.50,0.13) and X14 is GMF(0.50,0.55) and X15 is GMF(0.50,0.31)	Y = 1.30 + 0.38 * X1 - 0.20 * X2 - 0.11 * X3 - 0.10 * X4 - 0.12 * X6 - 0.09 * X7 + 0.29 * X8 + 0.51 * X9 - 0.05 * X10 - 0.03 * X11 - 0.03 * X12 + 0.02 * X13 + 0.15 * X14 + 0.01 * X15
02	if X1 is GMF(0.50,0.64) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.37) and X4 is GMF(0.50,0.82) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.93) and X7 is GMF(0.50,0.70) and X8 is GMF(0.50,0.40) and X9 is GMF(0.50,0.11) and X10 is GMF(0.50,0.77) and X11 is GMF(0.50,0.13) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.13) and X14 is GMF(0.50,0.48) and X15 is GMF(0.50,0.39)	
03	if X1 is GMF(0.50,0.68)and X2 is GMF(0.50,0.24) and X3 is GMF(0.50,0.15) and X4 is GMF(0.50,0.95) and X5 is GMF(0.50,0.53) and X6 is GMF(0.50,0.64) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.29) and X9 is GMF(0.50,0.05) and X10 is GMF(0.50,0.46) and X11 is GMF(0.50,0.28) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.09) and X15 is GMF(0.50,0.39)	
04	if X1 is GMF(0.50,0.77) and X2 is GMF(0.50,0.54) and X3 is GMF(0.50,0.79) and X4 is GMF(0.50,0.29) and X5 is GMF(0.50,0.95) and X6 is GMF(0.50,0.44) and X7 is GMF(0.50,0.26) and X8 is GMF(0.50,0.27) and X9 is GMF(0.50,0.91) and X10 is GMF(0.50,0.36) and X11 is GMF(0.50,0.95) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.09) and X14 is GMF(0.50,0.28) and X15 is GMF(0.50,0.50)	
05	if X1 is GMF(0.50,0.31) and X2 is GMF(0.50,0.05) and X3 is GMF(0.50,0.31) and X4 is GMF(0.50,0.57) and X5 is GMF(0.50,0.58) and X6 is GMF(0.50,0.93) and X7 is GMF(0.50,0.67) and X8 is GMF(0.50,0.12) and X9 is GMF(0.50,0.07) and X10 is GMF(0.50,0.95) and X11 is GMF(0.50,0.05) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.22) and X15 is GMF(0.50,0.46)	
06	if X1 is GMF(0.50,0.62) and X2 is GMF(0.50,0.33) and X3 is GMF(0.50,0.26) and X4 is GMF(0.50,0.79) and X5 is GMF(0.50,0.42) and X6 is GMF(0.50,0.36) and X7 is GMF(0.50,0.53) and X8 is GMF(0.50,0.27) and X9 is GMF(0.50,0.08) and X10 is GMF(0.50,0.05) and X11 is GMF(0.50,0.37) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.17) and X15 is GMF(0.50,0.39)	
07	if X1 is GMF(0.50,0.53) and X2 is GMF(0.50,0.36) and X3 is GMF(0.50,0.42) and X4 is GMF(0.50,0.40) and X5 is GMF(0.50,0.54) and X6 is GMF(0.50,0.87) and X7 is GMF(0.50,0.49) and X8 is GMF(0.50,0.36) and X9 is GMF(0.50,0.34) and X10 is GMF(0.50,0.79) and X11 is GMF(0.50,0.27) and X12 is GMF(0.50,0.08) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.14) and X15 is GMF(0.50,0.46)	

08	if X1 is GMF(0.50,0.51) and X2 is GMF(0.50,0.24) and X3 is GMF(0.50,0.15) and X4 is GMF(0.50,0.54) and X5 is GMF(0.50,0.63) and X6 is GMF(0.50,0.47) and X7 is GMF(0.50,0.67) and X8 is GMF(0.50,0.61) and X9 is GMF(0.50,0.22) and X10 is GMF(0.50,0.30) and X11 is GMF(0.50,0.20) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.08) and X14 is GMF(0.50,0.30) and X15 is GMF(0.50,0.58)	Y = 1.30 + 0.38 * X1 - 0.20 * X2 - 0.11 * X3 - 0.10 * X4 - 0.12 * X6 - 0.09 * X7 + 0.29 * X8 + 0.51 * X9 - 0.05 * X10 - 0.03 * X11 - 0.03 * X12 + 0.02 * X13 + 0.15 * X14 + 0.01 * X15
09	if X1 is GMF(0.50,0.69) and X2 is GMF(0.50,0.76) and X3 is GMF(0.50,0.90) and X4 is GMF(0.50,0.05) and X5 is GMF(0.50,0.70) and X6 is GMF(0.50,0.73) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.18) and X9 is GMF(0.50,0.53) and X10 is GMF(0.50,0.61) and X11 is GMF(0.50,0.49) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.06) and X14 is GMF(0.50,0.05) and X15 is GMF(0.50,0.50)	
10	if X1 is GMF(0.50,0.64) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.37) and X4 is GMF(0.50,0.71) and X5 is GMF(0.50,0.76) and X6 is GMF(0.50,0.19) and X7 is GMF(0.50,0.52) and X8 is GMF(0.50,0.24) and X9 is GMF(0.50,0.10) and X10 is GMF(0.50,0.39) and X11 is GMF(0.50,0.17) and X12 is GMF(0.50,0.06) and X13 is GMF(0.50,0.06) and X14 is GMF(0.50,0.26) and X15 is GMF(0.50,0.31)	
11	if X1 is GMF(0.50,0.68) and X2 is GMF(0.50,0.39) and X3 is GMF(0.50,0.47) and X4 is GMF(0.50,0.35) and X5 is GMF(0.50,0.78) and X6 is GMF(0.50,0.70) and X7 is GMF(0.50,0.63) and X8 is GMF(0.50,0.23) and X9 is GMF(0.50,0.32) and X10 is GMF(0.50,0.64) and X11 is GMF(0.50,0.11) and X12 is GMF(0.50,0.06) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.33) and X15 is GMF(0.50,0.39)	
12	if X1 is GMF(0.50,0.08) and X2 is GMF(0.50,0.66) and X3 is GMF(0.50,0.05) and X4 is GMF(0.50,0.25) and X5 is GMF(0.50,0.72) and X6 is GMF(0.50,0.95) and X7 is GMF(0.50,0.95) and X8 is GMF(0.50,0.18) and X9 is GMF(0.50,0.10) and X10 is GMF(0.50,0.95) and X11 is GMF(0.50,0.05) and X12 is GMF(0.50,0.19) and X13 is GMF(0.50,0.13) and X14 is GMF(0.50,0.06) and X15 is GMF(0.50,0.95)	
13	if X1 is GMF(0.50,0.84) and X2 is GMF(0.50,0.74) and X3 is GMF(0.50,0.74) and X4 is GMF(0.50,0.31) and X5 is GMF(0.50,0.72) and X6 is GMF(0.50,0.24) and X7 is GMF(0.50,0.27) and X8 is GMF(0.50,0.24) and X9 is GMF(0.50,0.78) and X10 is GMF(0.50,0.51) and X11 is GMF(0.50,0.83) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.09) and X14 is GMF(0.50,0.05) and X15 is GMF(0.50,0.61)	
14	if X1 is GMF(0.50,0.49) and X2 is GMF(0.50,0.31) and X3 is GMF(0.50,0.21) and X4 is GMF(0.50,0.51) and X5 is GMF(0.50,0.60) and X6 is GMF(0.50,0.64) and X7 is GMF(0.50,0.79) and X8 is GMF(0.50,0.22) and X9 is GMF(0.50,0.38) and X10 is GMF(0.50,0.45) and X11 is GMF(0.50,0.14) and X12 is GMF(0.50,0.09) and X13 is GMF(0.50,0.10) and X14 is GMF(0.50,0.49) and X15 is GMF(0.50,0.54)	

15	if X1 is GMF(0.50,0.66) and X2 is GMF(0.50,0.49) and X3 is GMF(0.50,0.58) and X4 is GMF(0.50,0.43) and X5 is GMF(0.50,0.65) and X6 is GMF(0.50,0.61) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.30) and X9 is GMF(0.50,0.72) and X10 is GMF(0.50,0.79) and X11 is GMF(0.50,0.47) and X12 is GMF(0.50,0.06) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.38) and X15 is GMF(0.50,0.27)	Y = 1.30 + 0.38 * X1 - 0.20 * X2 - 0.11 * X3 - 0.10 * X4 - 0.12 * X6 - 0.09 * X7 + 0.29 * X8 + 0.51 * X9 - 0.05 * X10 - 0.03 * X11 - 0.03 * X12 + 0.02 * X13 + 0.15 * X14 + 0.01 * X15
16	if X1 is GMF(0.50,0.62) and X2 is GMF(0.50,0.39) and X3 is GMF(0.50,0.47) and X4 is GMF(0.50,0.65) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.05) and X7 is GMF(0.50,0.40) and X8 is GMF(0.50,0.95) and X9 is GMF(0.50,0.14) and X10 is GMF(0.50,0.28) and X11 is GMF(0.50,0.31) and X12 is GMF(0.50,0.06) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.38) and X15 is GMF(0.50,0.31)	
17	if X1 is GMF(0.50,0.05) and X2 is GMF(0.50,0.72) and X3 is GMF(0.50,0.05) and X4 is GMF(0.50,0.57) and X5 is GMF(0.50,0.06) and X6 is GMF(0.50,0.93) and X7 is GMF(0.50,0.95) and X8 is GMF(0.50,0.40) and X9 is GMF(0.50,0.21) and X10 is GMF(0.50,0.75) and X11 is GMF(0.50,0.19) and X12 is GMF(0.50,0.95) and X13 is GMF(0.50,0.95) and X14 is GMF(0.50,0.95) and X15 is GMF(0.50,0.05)	
18	if X1 is GMF(0.50,0.49) and X2 is GMF(0.50,0.61) and X3 is GMF(0.50,0.95) and X4 is GMF(0.50,0.27) and X5 is GMF(0.50,0.47) and X6 is GMF(0.50,0.84) and X7 is GMF(0.50,0.54) and X8 is GMF(0.50,0.05) and X9 is GMF(0.50,0.38) and X10 is GMF(0.50,0.91) and X11 is GMF(0.50,0.39) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.05) and X15 is GMF(0.50,0.31)	
19	if X1 is GMF(0.50,0.49) and X2 is GMF(0.50,0.23) and X3 is GMF(0.50,0.26) and X4 is GMF(0.50,0.85) and X5 is GMF(0.50,0.33) and X6 is GMF(0.50,0.61) and X7 is GMF(0.50,0.50) and X8 is GMF(0.50,0.36) and X9 is GMF(0.50,0.11) and X10 is GMF(0.50,0.87) and X11 is GMF(0.50,0.30) and X12 is GMF(0.50,0.06) and X13 is GMF(0.50,0.07) and X14 is GMF(0.50,0.31) and X15 is GMF(0.50,0.42)	
20	if X1 is GMF(0.50,0.36) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.74) and X4 is GMF(0.50,0.25) and X5 is GMF(0.50,0.56) and X6 is GMF(0.50,0.93) and X7 is GMF(0.50,0.59) and X8 is GMF(0.50,0.29) and X9 is GMF(0.50,0.21) and X10 is GMF(0.50,0.93) and X11 is GMF(0.50,0.24) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.05) and X15 is GMF(0.50,0.31)	
21	if X1 is GMF(0.50,0.62) and X2 is GMF(0.50,0.39) and X3 is GMF(0.50,0.53) and X4 is GMF(0.50,0.65) and X5 is GMF(0.50,0.13) and X6 is GMF(0.50,0.53) and X7 is GMF(0.50,0.68) and X8 is GMF(0.50,0.71) and X9 is GMF(0.50,0.30) and X10 is GMF(0.50,0.73) and X11 is MF(0.50,0.19) and X12 is GMF(0.50,0.10) and X13 is GMF(0.50,0.12) and X14 is GMF(0.50,0.80) and X15 is GMF(0.50,0.46)	

22	if X1 is GMF(0.50,0.44) and X2 is GMF(0.50,0.26) and X3 is GMF(0.50,0.26) and X4 is GMF(0.50,0.82) and X5 is GMF(0.50,0.46) and X6 is GMF(0.50,0.64) and X7 is GMF(0.50,0.64) and X8 is GMF(0.50,0.37) and X9 is GMF(0.50,0.15) and X10 is GMF(0.50,0.57) and X11 is GMF(0.50,0.11) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.17) and X15 is GMF(0.50,0.54)	Y = 1.30 + 0.38 * X1 - 0.20 * X2 - 0.11 * X3 - 0.10 * X4 - 0.12 * X6 - 0.09 * X7 + 0.29 * X8 + 0.51 * X9 - 0.05 * X10 - 0.03 * X11 - 0.03 * X12 + 0.02 * X13 + 0.15 * X14 + 0.01 * X15
23	if X1 is GMF(0.50,0.68) and X2 is GMF(0.50,0.51) and X3 is GMF(0.50,0.69) and X4 is GMF(0.50,0.44) and X5 is GMF(0.50,0.65) and X6 is GMF(0.50,0.36) and X7 is GMF(0.50,0.18) and X8 is GMF(0.50,0.18) and X9 is GMF(0.50,0.53) and X10 is GMF(0.50,0.55) and X11 is GMF(0.50,0.83) and X12 is GMF(0.50,0.07) and X13 is GMF(0.50,0.08) and X14 is GMF(0.50,0.59) and X15 is GMF(0.50,0.39)	
24	if X1 is GMF(0.50,0.95) and X2 is GMF(0.50,0.95) and X3 is GMF(0.50,0.79) and X4 is GMF(0.50,0.69) and X5 is GMF(0.50,0.05) and X6 is GMF(0.50,0.76) and X7 is GMF(0.50,0.87) and X8 is GMF(0.50,0.40) and X9 is GMF(0.50,0.35) and X10 is GMF(0.50,0.64) and X11 is GMF(0.50,0.74) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.05) and X15 is GMF(0.50,0.54)	
25	if X1 is GMF(0.50,0.68) and X2 is GMF(0.50,0.34) and X3 is GMF(0.50,0.31) and X4 is GMF(0.50,0.59) and X5 is GMF(0.50,0.60) and X6 is GMF(0.50,0.50) and X7 is GMF(0.50,0.78) and X8 is GMF(0.50,0.12) and X9 is GMF(0.50,0.15) and X10 is GMF(0.50,0.16) and X11 is GMF(0.50,0.09) and X12 is GMF(0.50,0.05) and X13 is GMF(0.50,0.05) and X14 is GMF(0.50,0.07) and X15 is GMF(0.50,0.39)	
26	if X1 is GMF(0.50,0.47) and X2 is GMF(0.50,0.38) and X3 is GMF(0.50,0.63) and X4 is GMF(0.50,0.59) and X5 is GMF(0.50,0.49) and X6 is GMF(0.50,0.24) and X7 is GMF(0.50,0.43) and X8 is GMF(0.50,0.45) and X9 is GMF(0.50,0.44) and X10 is GMF(0.50,0.48) and X11 is GMF(0.50,0.35) and X12 is GMF(0.50,0.09) and X13 is GMF(0.50,0.09) and X14 is GMF(0.50,0.52) and X15 is GMF(0.50,0.42)	

GMF(*x*, *y*) indicates Gaussian Membership function with mean *x* and variance *y*.

References

Aghaie, A. and Saeedi, A., (2009) Using Bayesian Networks for Bankruptcy Prediction: Empirical Evidence from Iranian Companies, *International Conference on Information Management and Engineering*, April 3-5, 2009, Kuala Lumpur, Malaysia.

Aizerman, M.A., Braverman, E.M. and Rozono´er, L.I., (1964) Theoretical foundations of the potential function method in pattern recognition learning, *Automation and Remote Control*, 25, 821–837.

Alam, P., Booth, D., Lee, K. and Thordarson, T., (2000) The use of fuzzy clustering algorithm and self-organization neural networks for identifying potentially failing banks: an experimental study, *Expert Systems with Applications*, 18(3), 185-199.

Aliev, R.A., Aliev, R.R., Guirimov, B. and Uyar, K., (2008) Dynamic data mining technique for rules extraction in a process of battery charging, *Applied Soft Computing*, 8, 1252-1258.

Altman, E., (1968) Financial Ratios. Discriminant Analysis and the Prediction of Corporate Bankruptcy. *Journal of Finance*, 23, 589-609.

Amari, S. and Kasabov, N., (1997) Eds., Brain-Like Computing and Intelligent Information Systems. New York: Springer Verlag.

Andres, J.D., Landajo, M. and Lorca, P., (2005) Forecasting business profitability by using classification techniques: A comparative analysis based on a Spanish case, *European Journal of Operational Research*, 167(2), 518-542.

Andrews, R, Diederich, J, Tickle, A.B., (1995) A Survey and Critique of Techniques For Extracting Rules From Trained Artificial Neural Networks, *Knowledge Based Systems*, 8, 373–389.

Anton, J., (1996) Customer Relationship Management: Making Hard Decisions with Soft Numbers, Prentice Hall, Englewood Cliffs, New Jersey.

Arbatli, A.D. and Akin, H.L., (1997) Rule extraction from trained neural networks using genetic algorithms, *Nonlinear Analysis, theory, Methods and Applications*, 30(3), 1639-1648.

Aronszajn, N., (1950) Theory of reproducing kernels, *Transactions of American Mathematical Society*, 686, 337–404.

Asuncion, A., and Newman, D.J., (2007) *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science.

Atiya, A.F., (2001) Bankruptcy prediction for credit risk using neural networks: A survey and new results, *IEEE Transactions on Neural Networks*, 12, 929-935.

Baek, J. and Cho, S., (2003) Bankruptcy prediction for credit risk using an auto-associative neural network in korean firms, *In the proceedings of the* ***CIFEr,*** 25-29, Hong Kong.

Barakat, N.H. and Bradley, A.P., (2006) Rule Extraction from Support Vector Machines: Measuring the Explanation Capability Using the Area under the ROC Curve, *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)*, Hong Kong.

Barakat, N.H. and Bradley, A.P., (2007) Rule Extraction from Support Vector Machines: A Sequential Covering Approach, *IEEE Transactions on Knowledge and Data Engineering*, 19(6), 729-741.

Barakat, N.H. and Diederich, J., (2004a) Learning-based rule-extraction from support vector machines: Performance on benchmark datasets, *Proceedings of the conference on Neuro-Computing and Evolving Intelligence, Knowledge Engineering and Discovery Research Institute (KEDRI)*, Auckland, New Zealand.

Barakat, N.H. and Diederich, J., (2004b) Learning-based rule extraction from support vector machines, *Proceedings of the 14th International Conference on Computer Theory and Applications (ICCTA'2004)*.

Barakat, N.H. and Diederich, J., (2005) Eclectic Rule-Extraction from Support Vector Machines, *International Journal of Computational Intelligence*, 2(1), 59-62.

Batista, G.E.A.P.A., Monad, M.C. and Bazzan, A.L.C., (2004a) Improving rule induction precision for automated annotation by balancing skewed datasets, *Knowledge exploration in life science informatics (KELSI)*, 3303, 20–32.

Batista, G.E.A.P.A., Prati, R.C., and Monard, M.C., (2004b) A Study of the behaviour of several methods for balancing machine learning training data, *ACM SIGKDD Explorations: Special Issue on Imbalanced Datasets*, 6(1), 20–29.

Beaver, R., (1966) Financial ratios as predictors of failure. *Empirical Research in Accounting: Selected Studies 1966, J. Accounting Research*, 4, 71–111.

Becerra, V.M., Galvao, H. and Abou-Seads, M., (2005) Neural and Wavelet Network Models for Financial Distress Classification, *Data Mining and Knowledge Discovery*, 11, 35-55.

Bell, T., (1997) Neural Nets or the Logit Model? A Comparison of Each Model's Ability to Predict Commercial Bank Failures, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 6(3), 249-264.

Bellman, R.E., (1961) Adaptive Control Processes, Princeton, NJ: Princeton University Press.

Ben-Hui, A., Horn, D., Sidgelmann, H., and Vapnik, V.N., (2001) Support vector clustering. *Machine Learning Research*, 2, 125-137.

Beynon, M.J., and Peel, M.J., (2001) Variable Precision Refought Set Theory and Data Discretisation: An Application to Corporate Failure Prediction, *Omega*, 29, 561-576.

Blanz, V., Scholkopf, B., Bulthoff, H., Burges, C., Vapnik, V.N. and Vetter, T., (1996) Comparison of view–based object recognition algorithms using realistic 3D models, *In C. von der Malsburg, W. von Seelen, J. C. Vorbr¨uggen, and B. Sendhoff (Eds.), Artificial Neural Networks — ICANN'96*, Berlin, 1112, 251 – 256. Springer Lecture Notes in Computer Science.

Bloemer, J., Ruyter, K.D. and Peeters, P., (1998) Investigating drivers of bank loyalty: the complex relationship between image, service quality and satisfaction, *International Journal of Bank Marketing*, 16(7), 276–286.

Bolton, R.N., (1998) A Dynamic model of the Duration of the customer's relationship with a continuous service provider: The Role of Satisfaction, *Marketing Science*, 17(1), 45-65.

Bolton, R.N., Kannan, P.K., Bramlett, M.D., (2000) Implications of Loyalty Program Membership and Service Experiences for Customer Retention and Value, *Journal of the Academy of Marketing Science*, 28(1), 95-108.

Boser, B.E., Guyon, I.M. and Vapnik, V.N., (1992) A training algorithm for optimal margin classifiers, *In D. Haussler, editor, Proceedings of the Annual Conference on Computational Learning Theory*, 144-152, Pittsburgh, PA, july 1992, ACM Press.

Box, G.E.P. and Muller, M.E., (1958) A Note on the Generation of Random Normal Deviates, *The Annals of Mathematical Statistics*, 29(2), 610–611.

Brause, R., Langsdorf, T. and Hepp, M., (1999) Neural Data Mining for Credit Card Fraud Detection, *In Proceedings of 11th IEEE International Conference on Tools with Artificial Intelligence*, 103-106, Illinois, USA.

Breiman, L., Friedman, J., Olsen, R. and Stone, C., (1984) Classification and Regression Trees, Wadsworth and Brooks.

Brockett, P., Derrig, R., Golden, L., Levine, A. and Alpert, M., (2002) Fraud Classification using Principal Component Analysis of RIDITs, *Journal of Risk and Insurance*, 69(3), 341-371.

Browne, A., Hudson, B., Whitley, D. and Picton, P., (2004) Biological data mining with neural networks: implementation and application of a flexible decision tree extraction algorithm to genomic problem domains, *Neurocomputing*, 57, 275-293.

Buckinx, W. and Van den Poel, D., (2005) Customer base analysis: partial defection of behaviorally loyal clients in a non-contractual FMCG retail setting, *European Journal of Operational Research*, 164(1), 252–268.

Burbidge, R., Trotter, M., Buxton, B. and Holden, S., (2001) Drug design by machine learning: support vector machines for pharmaceutical data analysis, *Computers and Chemistry*, 26(1), 5-14.

Burges, C.J.C., (1998) A Tutorial on Support Vector Machines for Pattern Recognition, *Data Mining and Knowledge Discovery*, 2, 1-43.

Business Intelligence Cup-2004: Organized by the University of Chile. Available at: http://www.tis.cl/bicup_04/text-bicup/BICUP/202004/20public/20data.zip.

Byun, H. and Lee, S.W., (2002) Applications of support vector machines for pattern recognition: A Survey, *In Lee, S.W. and Verri, A., eds.*, LNCS, 213-236.

Cai, J., Dayanik, A., Yu, H., Hasan, N., Terauchi, T. and Grundy, W., (2000) Classification of cancer tissue types by support vector machines using micro array gene expression data, *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, August 19-23, 2000, La Jolla / San Diego, CA, USA. AAAI 2000, ISBN 1-57735-115-0.

Cai, L. and Hofmann, T., (2004) Hierarchical document categorization with support vector machines, *Proceedings of the International Conference on Information and Knowledge Management (CIKM'04)*, 78-87.

Campos, P.G. and Ludermir, T.B., (2005) Literal and ProRulext: Algorithms for Rule Extraction of ANNs, *Proceedings of the fifth international conference on Hybrid Intelligent Systems (HIS'05)*, November 6-9, 2005, Rio de Janeiro, Brazil. IEEE Computer Society 2005, ISBN 0-7695-2457-5.

Canbas, S.C., and Kilic, S.B., (2005) Prediction of commercial bank failure via multivariate statistical analysis of financial structures: The Turkish case, *European Journal of Operational Research,* 166, 528-546.

Cao, K. and Shao, P., (2008) Customer Churn Prediction Based on SVM-RFE, *International Seminar on Business and Information Management, ISBIM'08*, 1, 306-309.

Cao, L.J. and Tay, F.E.H., (2003) Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting, *IEEE Transactions on Neural Network*, 14(6), 1506-1518.

Carse, B., Fogarty, T.C. and Munro, A., (1996) Evolving fuzzy rule based controllers using genetic algorithms, *Fuzzy Sets Systems*, 80, 273–293, June 1996.

Chan, P.K., Fan, W., Prodromidis, A.L. and Stolfo, S.J., (1999) Distributed Data Mining in Credit Card Fraud Detection, *IEEE Transactions on Intelligent Systems*, 14, 67-74.

Chan, C.C., Lin, C.J. (2001) LIBSVM: a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Charalambous, C., Charitou, A. and Kaourou, F., (1999) Comparative analysis of artificial neural network models: application in bankruptcy prediction, *International Joint Conference on Neural Networks, IJCNN'99*, 6, 3888 – 3893, July 1999, Washington, DC.

Charalambous, C., Charitou, A. and Kaourou, F., (2000) Application of Feature Extractive Algorithm to Bankruptcy Prediction, *International Joint Conference on Neural Networks*, 5, 5303, July 24- 27, 2000, Como, Italy.

Chauhan, N., Ravi, V. and Chandra**,** D. K., (2008) Differential evolution trained wavelet neural network: application to bankruptcy prediction in banks, *Expert Systems with Applications*, 36(4), 7659-7665.

Chaves, Ad.C.F., Vellasco, M.M.B.R., and Tanscheit, R., (2005) Fuzzy Rule Extraction from Support Vector Machines, *Fifth International Conference on Hybrid Intelligent Systems*, November 06-09, Rio de Janeiro, Brazil.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P., (2002) SMOTE: Synthetic Minority Oversampling Technique, *Journal of Artificial Intelligence Research*, 16, 321-357.

Chen, H. (1995) Machine Learning for information retrieval: neural networks, symbolic learning and genetic algorithms, *Journal of the American Society for Information Science*, 46(3), 194-216. John Wiley and Sons, Inc. NY, USA.

Chen, K-Y. and Ho, C-H., (2005) An Improved Support Vector Regression Modelling for Taiwan Stock Exchange Market Weighted Index Forecasting, *International Conference on Neural Networks and Brain, ICNN&B'05*, 3, 1633-1638.

Chen, Z., Li, J. and Wei, L., (2007) A multiple kernel support vector machine scheme for feature selection and rule extraction from gene expression data of cancer tissue, *Artificial Intelligence in Medicine*, 41, 161-175.

Christianini, N. and Shawe-Taylor, J., (2000) An introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, New York, NY, USA.

Chu, B.H., Tsai, M-S. and Ho, C-S., (2007) Toward a hybrid data mining model for customer retention, *Knowledge-Based Systems*, 20(8), 703–718.

Clark, P. and Niblett, T., (1989) The CN2 Induction Algorithm, *Machine Learning*, 3(4), 261-283.

Cohen, G., Hilario, M., Sax, H., Hogonnet, S. and Geissbuhler, A., (2006) Learning from imbalanced data in surveillance of nosocomial infection, *Artificial Intelligence in Medicine*, 37, 7–18.

Cole, R. and Gunther, J., (1995) A CAMEL Rating's Shelf Life, *Federal Reserve Bank of Dallas Review*, December 13-20, 2005.

Cortes, C. and Vapnik, V.N., (1995) Support vector networks, *Machine Learning*, 20, 273-297.

Craven, M.W., (1996) Extracting Comprehensible Models from Trained Neural Networks, *PhD thesis*, Department of Computer Science, University of Wisconsin-Madison.

Craven, M.W. and Shavlik, J.W., (1994) Using sampling and queries to extract rules from trained neural networks, *Proceedings of the Eleventh International Conference on Machine Learning*, San Francisco, CA, USA.

Craven, M.W. and Shavlik, J.W., (1996) Extracting Tree-Structured Representations of Trained Networks, Touretzky, D., Mozer, M. and Hasselmo, M. eds. *Advances in Neural Information Processing Systems*, 8, 24-30, The MIT Press, citeseer.ist. psu.edu/craven96extracting.html, 1996

Davel, M. and Barnard, E., (2004) The effficient generation of pronunciation dictionaries: Machine learning factors during bootstrapping, *Proceedings of the 8th International Conference on Spoken Language Processing*, Korea, 2781–2784.

Davis, R., Buchanan, B.G. and Shortliffe, E., (1977) Production rules as a representation for a knowledge-based consultation program, *Artificial Intelligence*, 8(1), 15-45.

Domingos, P., (1999) MetaCost: A general method for making classifiers cost-sensitive, *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 155-164, San Diego, CA.

Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A. and Vapnik, V.N., (1997) Support vector regression machines, *In M.C. Mozer, M.I. Jordan and T. Pestche, eds, Advances in Neural Information Processing Systems*, 9, 155-161, MIT Press, MA.

Eads, D., Hill, D., Davis, S., Perkins, S., Ma, J., Porter, R. and Theiler, J., (2002) Genetic algorithms and support vector machines for time series classification, *In: 5th Conference on the Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation*. Symposium on Optical Science and Technology of the 2002 SPIE Annual Meeting. http://www.cs.rit.edu/~dre9227/papers/eadsSPIE4787.pdf.

Edwards, J., (2007) Get It Together with Collaborative CRM, *Inside CRM*. Tippit. http://www.insidecrm.com/features/collaborative-crm-112907/.

El-Naqa, I., Yang, Y., Wernick, M., Galatsanos, N. and Nishikawa, M., (2002) A support vector machine approach for detection of micro-calcifications, *IEEE Transactions on Medical Imaging*, 21, 1552-1563.

Estabrooks, A., Jo, T. and Japkowicz, N., (2004) A Multiple Re-sampling Method for Learning from Imbalances Datasets, *Computational Intelligence*, 20(1), 18-36.

Etemadi, H., Rostamy, A.A.A. and Dhlkordi, H.F., (2009) A genetic programming model for bankruptcy prediction: Empirical evidence from Iran, *Expert Systems with Applications*, 36(2), 3199-3207.

Euler, T., (2005) Churn Prediction in Telecommunications Using MiningMart, Available at: http://www-ai.cs.uni-dortmund.de, Viewed on 18 August 2007.

Faifer, M., Janikow, C. and Krawiec, K., (1999) Extracting fuzzy symbolic representation from artificial neural networks, *Proceedings of the 18th International Conference of the North American Fuzzy Information Processing Society (NAFIPS)*.

Fan, Y. and James Li, C., (2002) Diagnostic rule extraction from trained feed forward neural networks, *Mechanical Systems and Signal Processing*, 16(6), 1073-1081.

Fawcett, T., (2006) An introduction to ROC analysis, *Pattern Recognition Letters*, 27, 861–874.

Fawcett, T. and Provost, F., (1997) Adaptive fraud detection, *Data mining and Knowledge Discovery*, 1(3), 291-316.

Fayyad, U., Piatetsky-Shapiro, G. and Smyth, P., (1996) From data mining to knowledge discovery in databases, *AI Magazine*, 17(3), 37-54.

Ferreira, J.B., Vellasco, M., Pacheco, M.A. and Barbosa, C.H., (2004) Data mining techniques on the evaluation of wireless churn, *Proceedings of the European Symposium on Artificial Neural Networks Bruges, (ESANN'2004),* Belgium, d-side Publication, ISBN 2-930307-04-8, 483–488.

Fraser, D., (1976) The Determinants of Bank Profits: An Analysis of Extremes, *Financial Review*, 11(1), 69-87.

Fu, L.M., (1994) Rule generation from neural networks, *IEEE Transactions on Systems, Man and Cybernetics*, 24(8), 1114-1124.

Fu, X., Ong, C.J., Keerthi, S., Hung, G.G. and Goh, L., (2004) Extracting the Knowledge Embedded in Support Vector Machines, *In International Joint Conference on Neural Networks (IJCNN'04)*, Budapest, Hungary.

Fung, G., Sandilya, S. and Rao, R., (2005) Rule Extraction from Linear Support Vector Machines, *Proceedings of 11th International Conference on Knowledge Discovery in Data Mining (ACM SIGKDD'05)*, 32-40, August 21 - 24, 2005, Chicago, Illinois, USA.

Fürnkranz, J. and Flach, P.A., (2005) ROC 'n' Rule Learning-towards a better understanding of covering algorithms, *Machine Learning*, 58, 39-77.

Galindo, J. and Tamayo, P., (2000) Credit Risk Assessment using Statistical and Machine Learning: Basic Methodology and Risk Modelling Applications, *Computational Economics*, 15(1-2), 107-143.

Gallant, S., (1988) Connectionist expert systems, *Communications of the ACM*, 31(2), 152-169.

Gestel, T.V., Baesens, B., Suykens, J. and Espinoza, M., (2003) Bankruptcy Prediction with Least Squares Support Vector Machine Classifiers, *International Conference on Computational Intelligence for Financial Engineering (CIFEr2003)*, March 21-23, 2003, Hong Kong.

Gilbert, N., (1989) Explanation and dialogue, *The Knowledge Engineering Review*, 4(3), 235–247.

Giles, C. and Omlin, W., (1993) Extraction, insertion, and refinement of symbolic rules in dynamically driven recurrent network, *Connection Science*, 5, 307-328.

Girosi, F., (1997) An Equivalence Between Sparse Approximation and Support Vector Machines, A.I. Memo 1606, MIT Artificial Intelligence Laboratory.

Glady, N., Baesens, B. and Croux, C., (2009) Interfaces with Other Disciplines Modelling churn using customer lifetime value, *European Journal of Operational Research*, 197, 402–411.

Goodman, R., Higgins, C.M. Miller, J.W. and Smyth, P., (1992) Rule-based neural networks for classification and probability estimation, *Neural Computation*, 14, 781–804.

Gunn, S.R., Brown, M. and Bossley, K.M., (1997) Network performance assessment for neurofuzzy data modelling, *In Proceedings of Intelligent Data Analysis*, 1208, 313–323, Lecture Notes in Computer Science.

Guo, G. and Li, S., (2003) Content-based audio classification and retrieval by support vector machines, *IEEE Transactions on Neural Networks*, 14, 209-215.

Guo, H and Viktor, H.L., (2004) Learning from imbalanced datasets with boosting and data generation: The data boosting approach, *SIGKDD Explorations*, 6(1), 30-39.

Guo, X., Yin,Y., Dong, C., Yang, G. and Zhou,G., (2009) On the Class Imbalance Problem, *Fourth ICNC2008*, 4, 192-201, Jinan.

Guyon, I., Boser, B. and Vapnik, V.N., (1993) Automatic capacity tuning of very large VC-dimension classifiers, In S.J. Hanson, J.D. Cown and C.L. Giles, eds., *Advances in Neural Information Processing Systems*, 5, 147-155, Morgan Kaufmann Publishers.

Guyon, I., Weston, J., Barnhill, S. and Vapnik, V.N., (2002) Gene selection for cancer classification using support vector machines, *Machine Learning*, 46(1-3), 389-422.

Hadden, J., Tiwari, A., Roy, R. and Ruta, D., (2000) Computer assisted customer churn management: state-of-the-art and future trends, *Computers and Operations Research*, 34(10), 2902–2917.

Han, H., Wang, W.Y. and Mao, B.H., (2005) Borderline- SMOTE: A New Oversampling Method in Imbalanced Datasets Learning, *In Proceedings of the International Conference on Intelligent Computing*, Part I, 3644, 878–887, LNCS, Hefei, China.

Hart, P.E., (1968) The condensed nearest neighbour rule, *IEEE Transactions on Information Theory*, 18, 515–516.

Hadamard, J., (1923) Lectures on the Cauchy Problem in Linear Partial Differential Equations, Yale University Press.

Hayward, R., Nayak, R. and Diederich, J., (2000) Using Predicates to Explain Networks, *In: ECAI-2000 Workshop: "Foundations of Connectionist-Symbolic Integration: Representation, Paradigms and Algorithms*, Berlin, Germany.

He, J., Hu, H.-J., Harrison, R., Tai, P.C. and Pan, Y., (2006) Rule Generation for Protein Secondary Structure Prediction With Support Vector Machines and Decision Tree, *IEEE Transactions On Nanobioscience*, 5, 46-53.

Heckman, N., (1997) The Theory and Application of Penalized Least Squares Methods or Reproducing Kernel Hilbert Spaces Made Easy.

Hu, X., (2005) A data mining approach for retailing bank customer attrition analysis, *Applied Intelligence*, 22(1), 47–60.

Hu, Y.C. and Tseng, F.M., (2007) Functional-link net with fuzzy integral for bankruptcy prediction, *Neuro-Computing*, 70, 2959-2968.

Huang, K., Yang, H., King, I. and Lyu, M.R., (2004) Learning Classifiers from Imbalanced Data Based on Biased Minimax Probability Machine, *In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR`04)*, 2, 558-563.

Huang-Su, K. and Young-Gul, K., (2009) A CRM performance measurement framework: Its development process and application, *Industrial Marketing Management*, 38, 477–489.

Hung, S-Y., Yen, D.C. and Wang, H., (2006) Applying data mining to telecom churn management, *Expert Systems with Applications*, 31(3), 515–524.

Ishibuchi, H., Nakashima, T. and Murata, T., (1999) Performance Evaluation of Fuzzy Classifier Systems for Multidimensional Pattern Classification Problems, *IEEE Transactions on Systems, Man, and Cybernetics*, PART B: CYBERNETICS, 29(5), 601-618.

Jackson, J.C. and Craven. M.W., (1996) Learning sparse perceptrons, *Advances in Neural Information Processing Systems (NIPS)*, 8.

Jang, J.S.R., (1993) ANFIS: Adaptive Network-based Fuzzy Inference System, *IEEE Transactions on Systems, Man and Cybernetics*, 23(3), 665-685.

Japkowicz, N. and Stephen, S., (2002) The class imbalance problem: A systematic study, *Intelligent Data Analysis*, 6(5), 429-450.

Jo, H., Han, I. and Lee, H., (1997) Bankruptcy prediction using case-based reasoning, neural networks and discriminant analysis, *Expert Systems with Applications*, 13(2), 97-108.

Jo, T. and Japkowicz, N., (2004) Class Imbalances versus Small Disjuncts, *SIGKDD Explorations*, 6(1), 40-49.

Joachims, T., (1999) Transductive Inference for Text Classification using Support Vector Machines. *Proceedings of the International Conference on Machine Learning (ICML)*, June 27 - 30, 1999, Bled, Slovenia.

Kalatzis, D., Pappas, N., Piliouras. and Cavouras, D., (2003) Support vector machines based analysis of brain SPECT images for determining cerebral abnormalities in asymptomatic diabetic Patients, *Medical Informatics and the Internet in Medicine*, 28, 221- 230.

Kandel, A., (1988) Fuzzy Expert Systems Reading, MA Addison-Wesley.

Kandel. A., (1992) Fuuy Expert Systems, Boca Raton, FL CRC Press.

Karels, G.V. and Prakash, A.J., (1987) Multivariate normality and forecasting for business bankruptcy, *Journal of Business Finance and Accounting*, 14, 573-593.

Kasabov, N., (1996) Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering, Cambridge, MA: MIT Press, 1996.

Kasabov, N., (1998a) Evolving fuzzy neural networks-Algorithms, applications and biological motivation, *In Methodologies for the Conception, Design and Application of Soft Computing*, Yamakawa, T. and Matsumoto, G., Eds, Singapore: World Scientific, 271–274.

Kasabov, N. (1998b) ECOS: A framework for evolving connectionist systems and the eco learning paradigm, *In Proceedings ICONIP'98*. Kitakyushu, Japan, Oct. 1998, 1222–1235.

Kasabov, N. and Woodford, B., (1999) Rule insertion and rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems, *Proceedings of FUZZ-IEEE*, Aug. 1999.

Kasabov, N., (2001) Evolving fuzzy neural networks for online, adaptive, knowledge-based learning, *IEEE Transactions Systems, Man, Cybernetics*, PART B, 31, 902–918, Dec. 2001.

Kecman, V., (2001) Learning and soft computing: support vector machines, neural networks, and fuzzy logic models, MIT Press, Cambridge.

Kohavi, R., (1996) Scaling Up the Accuracy of Naïve-Bayes Classifiers: a Decisoin Tree Hybrid, *Proceedings of KDD-96*, Portland, USA.

Kohavi, R., (1997) Wrappers for feature subset selection, *In Artificial Intelligence journal*, special issue on Relevance, 97(1-2), 273-324.

Kononenko, I., (2001) Machine learning for medical diagnosis: history, state of the art and perspective, *Artificial Intelligence in Medicine*, *23*(1), 89–109.

Korobow, L., Stuhr, D. and Martin, D., (1976) A Probabilistic Approach to Early Warning Changes in Bank Financial Condition, *Federal Reserve Bank of New York, Monthly Review*, 187-194.

Kotler, P., (2000) Marketing Management: The Millennium Edition, Englewood Cliffs, NJ: Prentice Hall International.

Kotsiantis, S., Kanellopoulos, D. and Pintelas, P., (2006) Handling imbalanced datasets: A review, *GESTS International Transactions on Computer Science and Engineering*, 30(1), 25-36.

Krishnan, R., Sivakumar, G. and Bhattacharya, P., (1999) A search technique for rule extraction from trained neural networks, *Pattern Recognition Letters*, 20, 273-280.

Kubat, M., Holte, R. and Matwin, S., (2004) Machine learning for the detection of oil spills in satellite radar images, *Machine Learning*, 30(2-3), 195-215.

Kubat, M. and Matwin, S., (1997) Addressing the Curse of Imbalanced Training Sets: One Sided Selection, *In Proceedings of the Fourteenth International Conference on Machine Learning*, 179-186, Nashville, Tennessee, USA.

Kumar, D.A. and Ravi, V., (2008) Predicting credit card customer churn in banks using data mining, *International Journal for Data Analysis, Techniques and Strategies*, 1(1), 4-28.

Langley, P. and Simon, H.A., (1995) Applications of machine learning and rule induction, *Communications of the ACM*, 38(11), 54-64.

Larivie`re, B. and Van den Poel, D., (2004) Investigating the role of product features in preventing customer churn, by using survival analysis and choice modelling: The case of financial services, *Expert Systems with Applications*, 27(2), 277–285.

Laurikkala, J., (2001) Improving identification of difficult small classes by balancing class Distribution, *Artificial Intelligence in Medicine*, 2101, 63-66.

Lee, K.C., Han, I. and Kwon, Y., (1996) Hybrid neural networks for bankruptcy predictions, *Decision Support Systems*, 18(1), 63-72.

Lee, W., Stolfo, S. and Mok, K., (1999) Mining in a Data-flow Environment: Experience in Network Intrusion Detection, *In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*, 114-124, San Diego, CA.

Lemon, K.N., White, T.B. and Winer, R., (2002) Dynamic customer relationship management: Incorporating future considerations into the service retention decision, *Journal of Marketing*, 66, 1-14.

Lin, S., Wang, G., Zhang, S. and Li, J., (2006) Time Series Prediction Based on Support Vector Regression, *Information Technology Journal*, 5(2), 353-357.

Ling, C.X. and Li, C., (1998) Data Mining for Direct Marketing Problems and Solutions, *In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, 73-79, New York City, New York, USA.

Liu, Y., Chawla, N.V., Harper, M.P., Shrilberg, E. and Stolcke, A., (2006) A study in machine learning from imbalanced data for sentence boundary detection in speech, *Computer Speech and Language*, 20(4), 468–494.

Maloof, M.A., Langley, P., Binford, T.O., Nevatia, R. and Sage, S., (2003) Improved rooftop detection in aerial images with machine learning, *Machine Learning*, 53, 157–191.

Markowska-Kaczmar, U. and Trelak, W., (2003) Extraction of Fuzzy Rules from Trained Neural Network Using Evolutionary Algorithm, *Proceedings of European Symposium on Artificial Neural Networks (ESANN '03)*, 149-154.

Martens, D., Baesens, B., Gestel, T.V. and Vanthienen, J., (2006) Comprehensible credit scoring models using rule extraction from support vector machines, *European Journal of Operational Research*, 183, 1466–1476.

Martens, D., Baesens, B. and Gestel, T.V., (2009) Decompositional Rule Extraction from Support Vector Machines by Active Learning, *IEEE Transactions on Knowledge and Data Engineering*, 21(2), 178-191.

Massey, A., Montoya-Weiss, M. and Holcom, K., (2001) Re-engineering the customer relationship: leveraging knowledge assets at IBM, *Decision Support Systems*, 32, 155–170.

Mattera, D. and Haykin, S., (1999) Support vector machines for dynamic reconstruction of a chaotic system, In Scholkopf, B., burges, C.J.C, Smola, A.J., Eds., *Advances in Kernal Methods-Support Vector Learning*, 211-242, Cambridge, MA, MIT Press.

McDonald, G.C. and Schwing, R.C., (1973) Instabilities of regression estimates relating air pollution to mortality, *Technometrics*, 15, 463–482.

McGarry, K.J., Tait, J., Wermter, S. and MacIntyre, J., (1999) Rule-Extraction from Radial Basis Function Networks, *International Conference on Artificial Neural networks*, Edinburgh, September 7-10, 1999, University of Edinburgh, UK.

McKee, T.E., (2000) Developing a bankruptcy prediction model via rough set theory, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 9(3), 159-173.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T., (2006) YALE: Rapid Prototyping for Complex Data Mining Tasks, *In Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (KDD-06)*, August 20-23, 2006, Philadelphia, USA.

Min, J.H. and Lee, Y-C, (2005) Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters, *Expert Systems with Applications*, 28(4), 603-614.

Min, S-H., Lee, J. and Han, I., (2006) Hybrid genetic algorithms and support vector machines for bankruptcy prediction, *Expert Systems with Applications*, 31(3), 652-660.

Minoux, M., (1986) Mathematical Programming: Theory and Algorithms, John Wiley and Sons.

Mols, N.P., (1998) The Behavioral consequences of PC banking, *International Journal of Bank Marketing*, 16(5), 195-201.

Monard, M.C., Batista, G.E.A.P.A., (2002) Learning with Skewed Class Distribution, *In Advances in Logic, Artificial Intelligence and Robotics*, 173-180.

Mozer, M.C., Wolniewicz, R., Grimes, D.B., Johnson, E. and Kaushansky, H., (2000) Predicting subscriber dissatisfaction and improving retention in the wireless telecommunications industry, *IEEE Transactions of Neural Networks*, 11(3), 690–696.

Muller, K.R., Smola, A., Ratsch, G., Scholkopf, B., Kohlmorgen, J. and Vapnik, V.N., (1997) Predicting time series with support vector machines, In Gerstner, W., Germond, A., Hasler, M. and Nicoud, J.D., Eds, *Artificial Neural Networks ICANN'97*, 1327, 999-1004, Berlin. Springer Lecture Notes in Computer Science.

Mutanen, T., (2008) Customers churn analysis - a case study, *Research Report No. VTT-R-01184-06*, March 15.
Available at http://www.vtt.fi/inf/julkaisut/muut/2006/customer_churn_case_study.pdf, Retrieved on 19 August 2008.

Nakaoka, I., Tani, K., Hoshino, Y. and Kamei, K., (2006) A Bankruptcy Prediction Method Based on Cash flow Using SOM, *IEEE International Conference on Systems, Man, and Cybernetics*, October 8-11, 2006, Taipei, Taiwan.

Neslin, S.A., Gupta, S., Kamakura, W., Lu, J. and Mason, C., (2006) Defection detection: improving predictive accuracy of customer churn models, *Journal of Marketing Research*, 43(2), 204–211.

Nogueira, R., Vieira, S.M. and Sousa, J.M.C., (2004) The prediction of bankruptcy using fuzzy classifiers, *Transactions on Fuzzy Systems*, 12(5), 688-696.

Núñez, H., Angulo, C. and Catala, A., (2002a) Rule-extraction from support vector machines, *Proceedings of the European Symposium on Artificial Neural Networks*, 107-112.

Núñez, H., Angulo, C. and Catala, A., (2002b) Support vector machines with symbolic Interpretation, *Proceedings of the VII Brazilian Symposium on Neural networks (SBRn'02)*.

Núñez, H., Angulo, C. and Catala, A., (2002c) Rule based learning systems from SVM and RBFNN, *Neural Processing Letters*, 24(1), 1-18.

Núñez, H., Angulo, C. and Catala, A., (2006) Rule-Based Learning Systems for Support Vector Machines, *Neural Processing Letters*, 24, 1–18.

Odom, M.D. and Sharda, R., (1990) A Neural Network Model for Bankruptcy Prediction, *IJCNN International Joint Conference on Neural Networks*, 2, 163-168, San Diego, CA.

Ohlson, J.A., (1980) Financial Rations and the Probabilistic Prediction of Bankruptcy, *Journal of Accounting Research*, 18, 109-131.

O'Keefe, R.A., (1983) Concept formation from very large training sets, *In Proceedings of the Eighth International Joint Conference on Artificial Intelligence,* Karlsruhe, West Germany: Morgan Kaufmann.

Olmeda, I. and Fernandez, E., (1997) Hybrid classifiers for financial multicriteria decision making: the case of bankruptcy prediction, *Computational Economics*, 10, 317-335.

Pai, G.A.R. and Pai, G.A.V., (2004) Performance analysis of a Statistical and an Evolutionary Neural Networks based classifier for the prediction of industrial bankruptcy, *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*, December 1-3, Singapore.

Pantalone, C.C. and Platt, M.B., (1987) Predicting bank failure since deregulation, *New England Economic Review,* Federal Reserve Bank of Boston, 37-47.

Park, C-S. and Han, I., (2002) A case-based reasoning with the feature weights derived by analytic hierarchy process for bankruptcy prediction, *Expert Systems with Applications*, 23(3), 255-264.

Pearl, J., (1978a) Entropy, information and rational decisions (Technical report), Cognitive Systems Laboratory, University of California, Los Angeles.

Pedrycz, W., (1989) Fuzzy Control and Fuzzy Systems, New York: Wiley, 1989.

Pendharkar, P.C., (2009) Genetic algorithm based neural network approaches for predicting churn in cellular wireless network services, *Expert Systems with Applications*, 36, 6714-6720.

Penrose, K.W., Nelson, A.G. and Fisher, A.G., (1985) FACSM, Human Performance Research Center, Brigham Young University, Provo, Utah 84602 as listed in Medicine and Science in Sports and Exercise, 17 (2), 189.

Peppers, D. and Rogers, M., (1995) A new marketing paradigm, *Planning Review*, 23(2), 14-18.

Phua, C., Damminda, A. and Lee, V., (2004) Minority Report in Fraud Detection: Classification of Skewed Data, *SIGKDD Explorations: Special Issue on Imbalanced Datasets*, 6(1), 50-59.

Phua, C., Lee, V., Smith, K. and Gayler, R., (2005) A Comprehensive Survey of Data Mining-based Fraud Detection Research, *Artificial Intelligence review*, 2005.

Poggio, T., Torre, V. and Koch, C., (1985) Computational vision and regularization theory, *Nature*, 317, 314–319.

Pramodh, C. and Ravi, V., (2007) Modified Great Deluge Algorithm based Auto Associative Neural Network for Bankruptcy Prediction in Banks, *International Journal of Computational Intelligence Research*, 3(4), 363-370.

Provost, F., (2000) Machine learning from imbalanced datasets, *Invited paper for the AAAI'2000 Workshop on Imbalanced Datasets*.

Pyle D., (1999) Data Preparation for Data Mining, Morgan Kaufmann Publishers, San Francisco, USA.

Quinlan, J.R., (1983a) Learning efficient classification procedures and their application to chess endgames, In Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., Eds., *Machine learning: An artificial intelligence approach.* Palo Alto: Tioga Publishing Company.

Quinlan, J.R., (1993) C4.5 Programs for Machine Learning. Morgan Kaufmann.

Rahimian, E., Singh, S., Thammachote, T. and Virmani, R., (1996) Bankruptcy prediction by Neural network, In Trippi, R.R. and Turban, E., Eds., *Neural Networks in Finance and Investing*, Irwin Professional Publishing, Burr Ridge, USA.

Ravi, V., Kurniawan, H., Thai, P.N. and Ravikumar, P., (2008) Soft Computing System for Bank Performance prediction, *Applied Soft Computing Journal*, 8(1), 305-315.

Ravikumar, P. and Ravi, V., (2006a) Bankruptcy prediction in banks by Fuzzy Rule based classifier, *In the proceedings of 1st IEEE International Conference on Digital and Information Management*, Bangalore, 222-227.

Ravikumar, P. and Ravi, V., (2006b) Bankruptcy prediction in Banks by an Ensemble classifier, *In the proceedings of IEEE International Conference on Industrial Technology*, Mumbai, 2032-2036.

Ravikumar, P. and Ravi, V., (2007) Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A Review, *European Journal of Operational Research*, 180(1), 1-28.

Ravisankar, P. and Ravi, V., (2009) Failure Prediction of Banks Using Threshold Accepting Trained Kernel Principal Component Neural Network, *World Congress on Nature and Biologically Inspired Computing (NaBIC)*, 7-12, Coimbatore, India.

Richeldi, M. and Perrucci, A., (2002) Churn analysis case study enabling end-user data warehouse mining.
Available at: http://www-ai.informatik.uni-dortmund.de/DOKUMENTE/richeldi-_perrucci_2002b.pdf, Retrieved on 20 August 2002.

Ryu, Y.U. and Yue, W.T., (2005) Firm Bankruptcy Prediction: Experimental Comparison of Isotonic Separation and Other Classification Approaches, *IEEE Transactions on Systems, Management and Cybernetics*, PART A: SYSTEMS AND HUMANS, 35(5), 727-737.

Sanchez, D., Vila, M.A., Cerda, L. and Serrano, J.M., (2009) Association rules applied to credit card fraud detection, *Expert Systems with Applications*, 36(2), 3630-3640.

Sebastiani, F., (2002) Machine Learning in automated text categorization, *ACM Computing Surveys (CSUR)*, 34(1), 1-47, NY, USA.

Saito, K. and Nakano. R., (1988) Medical diagnostic expert system based on pdp model, *In Proceedings of IEEE International Conference on Neural Networks*, 255-262.

Salchenberger, L.C. and Lash, N., (1992) Neural Networks, Mine: A Tool for Predicting Thrift Failures, *Decision Sciences*, 23, 899-916.

Sato, M. and Tsukimoto, H., (2001) Rule Extraction from Neural Networks via Decision Tree Induction, *IEEE Conference on Machine Learning*, 393-400, June 28-July 01, 2001, Williamstown, MA, USA.

Scholkopf, B., Burges, C. and Vapnik, V.N., (1995) Extracting support data for a given task, In Fayyad, U.M. and Uthurusamy, R. Eds., *Proceedings of First International Conference on Knowledge Discovery and Data Mining*, Menlo Park, AAAI Press.

Scholkopf, B., Burges, C. and Vapnik, V.N., (1996) Incorporating invariances in support vector learning machines, In von der Malsburg, C., von Seelen, W., Vorbruggen, J.C. and Sendhoff, B., eds., *International Conference on Artificial Neural Networks ICANN'96*, 1112, 47-52, Berlin. Springer Lecture Notes in Computer Science.

Shaw, M., Subramaniam, C., Tan, G.W. and Welge, M., (2001) Knowledge management and data mining for marketing, *Decision Support Systems*, 32, 127–137.

Shin, K.S. and Lee, Y-J., (2002) A genetic algorithm application in bankruptcy prediction modeling, *Expert Systems with Applications*, 23(3), 321-328.

Shin, K.S., Lee, T.S., and Kim, H.J., (2005) An application of support vector machines in bankruptcy prediction model, *Expert Systems with Applications*, 28(1), 127-135.

Smith, K.A. and Gupta, J.N.D., (2000) Neural networks in business: techniques and applications for the operations researcher, *Computers and Operations Research*, 27(11–12), 1023–1044.

Smola, A.J. and Scholkopf, B., (1998) On a Kernel–based Method for Pattern Recognition, Regression, Approximation and Operator Inversion, *Algorithmica*, 22, 211-231. Technical Report 1064, GMD FIRST, April 1997.

Stefano, B. and Gisella, F., (2001) Insurance Fraud Evaluation: A Fuzzy Expert System, *In Proceedings of 10th IEEE International Conference Fuzzy Systems*, 3, 1491-1494, Melbourne, Australia.

Stitson, M., Gammerman, A., Vapnik, V.N., Vovk, V., Watkins, C. and Weston, J., (1999) Support vector regression with ANOVA decomposition kernels, In Scholkopf, B., Burges, C.J.C. and Smola, A.J. eds., *Adcances in Kernal Methods – Support Vector Learning*, 285-292, Cambridge, MA. MIT Press.

Stolfo, S.J., Fan, D.W., Lee, W. and Prodromidis, A.L., (1997a) Credit card fraud detection using meta-learning: Issues and initial results, *AAAI Workshop on AI Approaches to Fraud Detection and Risk Management*, 83-90, Providence, Rhode Island.

Stolfo, S.J., Prodromidis, A.L., Tselepis, S., Lee, W. and Fan, D.W., (1997b) JAM: Java agents for meta-learning over distributed databases, *AAAI Workshop on AI Approaches to Fraud Detection, In Proc. 3rd Intl. Conf. Knowledge Discovery and Data Mining*, 74-81.

Stolfo, S.J., Fan, D.W., Lee, W., Prodromidis, A.L. and Chan, P., (2000) Cost-based modeling for fraud and intrusion detection: Results from the JAM Project, *In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX`2000)*, 2, 130-144.

Sugeno, M., Ed., (1985) Industrial Applications of Fuzzy Control, New York. Elsevier.

Sun, J. and Li, H., (2008) Listed companies' financial distress prediction based on weighted majority voting combination of multiple classifiers, *Expert Systems with Applications*, 35(3), 818-827.

Swicegood, P. and Clark, J.A., (2001) Off-site monitoring for predicting bank under performance: A comparison of neural networks, discriminant analysis and professional human judgment, *International Journal of Intelligent Systems in Accounting, Finance and Management*, 10(3), 169-186.

Takagi, T. and Sugeno, M., (1983) Derivation of fuzzy control rules from human operator's control actions. *In Proceedings of IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis*, 55-60.

Takagi, T. and Sugeno, M., (1985) Fuzzy identification of systems and its applications to modelling and control, *IEEE Transactions on System, Man and Cybernetics*, 15, 116-132.

Tam, K.Y., (1991) Neural Network Models and the Prediction of Bank Bankruptcy, *OMEGA*, 19, 429-445.

Tam, K.Y. and Kiang, M., (1992) Predicting Bank Failures: A Neural Network Approach, *Decision Sciences*, 23, 926-947.

Tay, F.E.H. and Cao, L.J., (2002) e-Descending Support vector Machines for Financial Time Series Forecasting, *Neural Processing Letters*, 15, 179-195. Kluwer Academic Publishers, Netherlands.

Thrun, S., (1995) Extracting rules from artificial neural networks with distributed representations, *Advances in Neural Information Processing Systems (NIPS)*.

Tickle, A., Andrews, R., Golea, M. and Diederich, J., (1998) The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural network, *IEEE Transactions on Neural Networks*, 9(6), 1057–1068.

Torres, D. and Rocco, C., (2005) Extracting trees from trained SVM models using a TREPAN based approach, *Proceedings of Fifth International Conference on Hybrid Intelligent Systems*.

Towlell, G.G. and Shavlik, J.W., (1993) The extraction of refined rules from knowledge-based neural networks, *Machine Learning*, 13(1), 71-101.

Tsai., and Yu-Hsin, Lu., (2009) Customer churn prediction by hybrid neural networks, *Expert Systems with Applications*, 36, 12547–12553.

Tung, W.L., Quek, C. and Cheng, P., (2004) GenSo-EWS: a novel neural-fuzzy based early warning system for predicting bank failures, *Neural Networks*, 17(4), 567-587.

UCI Machine Learning Repository, http://archive.ics.uci.edu/ml/

Ultsch, A., (2002) Emergent self-organising feature maps used for prediction and prevention of churn in mobile phone markets, *Journal of Targeting, Measurement and Analysis for Marketing*, 10(4), 314–324.

Utgoff, P.E. (1988) Perceptron trees: a case study in hybrid concept representation, *In Proceedings of the Seventh National Conference on Artificial Intelligence*, 601-606. Morgan Kaufmann.

Vapnik, V.N., (1995) The Nature of Statistical Learning Theory, Springer-Verlag, New York Inc., New York, NY, USA.

Vapnik, V.N., Golowich, S. and Smola, A., (1997) Support vector method for function approximation, regression estimation and signal processing, In Mozer, M.C., Jordan, M.I. and Petsche, T., eds., *Advances in Neural Information Processing systems*, 9, 281-287, Cambridge, MA, MIT Press.

Verhoef, P. and Donkers, B., (2001) Predicting Customer Potential Value an Application in the Insurance Industry, *Decision Support Systems*, 32, 189–199.

Viaene, S., Derrig, R. and Dedene, G., (2004) A Case Study of Applying Boosting Naive Bayes to Claim Fraud Diagnosis, *IEEE Transactions on Knowledge and Data Engineering*, 16(5), 612-620.

Viera, A.S., Ribeiro, B., Mukkamala, S., Neves, J.C. and Sung, A.H., (2004) On the performance of learning machines for bankruptcy detection, IEEE *International Conference on Computational Cybernetics ICCC 2004*. Vienna University of Technology, August 30 - September 1, *2004*. Austria.

Visa, S. and Ralescu, A., (2005) Issues in Mining Imbalanced Datasets-A Review Paper, *In Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference*, MAICS-2005, 67-73.

Wahba, G., (1990) Spline Models for Observational Data, *Philadelphia: Series in Applied Mathematics*, 59, SIAM.

Wang, Y-F., Chiang, D-A., Hsu, M-H., Lin, C-J. and Lin, I-L., (2009) A recommender system to avoid customer churn: A case study, *Expert Systems with Applications*, 36, 8071–8075.

Weiss, G.M., (1995) Learning with Rare Cases and Small Disjuncts, *In Proceedings of the Twelfth International Conference on Machine Learning*, 558-565.

Weiss, G.M., (2004) Mining with rarity: A unifying framework, *SIGKDD Explorations*, 6(1), 7-19.

Werbos, P., (1974) Beyond regression: New tools for prediction and analysis in the behavioural sciences, *Ph.D. dissertation*, Harvard University, Cambridge, MA.

Wezel, M.V. and Potharst, R., (2007) Improved customer choice predictions using ensemble methods, *European Journal of Operational Research*, 181(1), 436–452.

Weiss, G.M. and Provost, F., (2001) The Effect of Class Distribution on Classifier Learning, *Technical Report Machine Learning-43*, Department of Computer Science, Rutgers University, January 2001.

Weiss, G.M. and Provost, F., (2003) Learning when training data are costly: the effect of class distribution on tree induction, *Journal of Artificial Intelligence Research*, 19, 315-354.

Wheeler, R. and Aitken, S., (2000) Multiple algorithms for fraud detection, *Knowledge-Based Systems*, 13(2-3), 93-99.

Widrow, B., Rumelhart, D.E. and Lehr, M.A., (1994) Neural networks: Applications in Industry, Business and Science, *Communications of the ACM*, 37(3), 93-105.

Wilson, D.L., (1972) Asymptotic Properties of Nearest Neighbour Rules using Edited Data, *IEEE Transactions on Systems, Man and Cybernetics*, 2, 408-420.

Wilson, R.L. and Sharda, R., (1994) Bankruptcy prediction using neural networks, *Decision Support Systems*, 11, 545-557.

Xie, Y., Li, X., Ngai, E.W.T. and Ying, W., (2009) Customer churn prediction using improved balanced random forests, *Expert Systems with Applications*, 36, 5445–5449.

Yuan, S-T. and Chang, W-L., (2001) Mixed-initiative synthesized learning approach for web-based CRM, *Expert Systems with Applications*, 20(2), 187–200.

Zadeh. L.A., (1965) Fuzzy sets, *Information and Control*, 8, 338-353.

Zadeh, L.A., (1973) Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Transactions on Systems, Man and Cybernetics*, 3, 28-44.

Zhang, G., Michael Y., Hu, B., Patuwo, E. and Indro, D.C., (1999) Artificial neural networks in bankruptcy prediction: general framework and cross validation analysis, *European Journal of Operational Research*, 116(1), 16-32.

Zhang, Y., Su, H., Jia, T. and Chu, J., (2005) Rule Extraction from Trained Support Vector Machines, 3518, 61-70, Springer Lecture Notes in Computer Science.

Zhao, Y., Li, B., Li, X., Liu, W. and Ren, S., (2005) Customer Churn Prediction Using Improved One-Class Support Vector Machine, *Advanced Data Mining and Applications*, 3584, 300-306, Springer Lecture Notes in Computer Science.

Papers Published

M.A.H. Farquad, V. Ravi and S.B. Raju, "Support vector regression based hybrid rule extraction methods for forecasting", *Expert Systems with Applications*, Volume 37, Issue 8, pp. 5577-5589, 2010.

M.A.H. Farquad, V. Ravi and S.B. Raju, "Rule Extraction from Support Vector Machines: A Hybrid Approach for classification and regression problems", International Journal of Information and Decision Sciences *(IJIDS)*, 2010. (In press)

M.A.H. Farquad, V. Ravi and S. Bapi Raju, "Rule Extraction from SVM for Analytical CRM: an Application to Predict Churn in Bank Credit Cards", *Applied Soft Computing*, 2010. (Under Review)

M.A.H. Farquad, V. Ravi and S. Bapi Raju, "Analytical CRM using SVM: a Modified Active Learning Based Rule Extraction approach", *Information Sciences*, 2010, (Under Review).

M.A.H. Farquad, V. Ravi and S.B. Raju, "Rule Extraction from Support Vector Machine using modified Active Learning Based Approach: An application to CRM", R. Setchi et al. (Eds.): *14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, KES 2010, Part I, LNAI 6276, pp. 461–470, September 8-10, 2010, Cardiff, Wales, UK.

M.A.H. Farquad, V. Ravi and S.B. Raju, "Data Mining using Rules Extracted from SVM: an Application to Churn Prediction in Bank Credit Cards", *Presented in 12th International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC'09)*, December 16-18, 2009, LNAI 5908, pp. 390-397, New Delhi, India.

M.A.H. Farquad, V. Ravi and S.B. Raju, "Rule Extraction using Support Vector Machine Based Hybrid Classifier", *Presented in TENCON-2008, IEEE Region 10 Conference*, November 19-21, 2008, Hyderabad, India.

M.A.H. Farquad, V. Ravi and S.B. Raju, "Support Vector Machine based Hybrid Classifiers and Rule Extraction Thereof: Application to Bankruptcy Prediction in Banks", *In Soria, E., Martín, J.D., Magdalena, R., Martínez, M., Serrano, A.J., editors, Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods and Techniques*, Vol. II, pp. 404-426, 2010, IGI Global, USA.

Brief Bio Data of the Author

NAME: **Mohammed Abdul Haque Farquad**

DATE OF BIRTH: **14th June, 1981**

SPECIALIZATION: Data Mining, Soft Computing (Support Vector Machine, Neural Network, Fuzzy Logic, Decision Tree, etc.), Banking and Finance (Bankruptcy prediction, Churn prediction, Fraud detection, Credit Scoring, forecasting), Customer Relationship Management (CRM).

AD HOC REFEREE:

- Knowledge Based Systems (Elsevier)
- IEEE Symposium on Computers and Informatics (ISCI'2011)
- Journal of Engineering and Computer Innovations (Academic Journals)

TOOLS ANALYZED:

- SAS EnterprizeMiner 6.1
- PASW Modeller
- RapidMiner
- Knime
- Weka
- MATLAB

HONOURS AND AWARDS:

- International Travel Grant by Department of Science and Technology, Government of India
- IDRBT Research Fellowship
- Earn While You Learn Scholarship

CONTACT DETAILS:

H.No 7-2-301,
Mankamma Thota,
Karimnager – 505002,
Andhra Pradesh, INDIA.
Phone: +91-9908202687, +91-9848316791
e-mail: farquadonline@gmail.com, Website: http://dcis.ernet.in/~farquad