Rule Extraction from Neural Networks and Optimization Techniques: Application to Banking

A Thesis submitted in fulfillment of the requirement of the Degree of

Doctor of Philosophy

in

Computer Science

by

Nekuri Naveen

07MCPC02

Department of Computer and Information Sciences

School of Mathematics and Computer & Information Sciences

University of Hyderabad

Hyderabad – 500 046, INDIA

November 2012

CERTIFICATE

This is to certify that the thesis work entitled "**Rule Extraction from Neural Networks and Optimization Techniques: Application to Banking**" submitted to University of Hyderabad by **Nekuri Naveen** bearing Reg. No. **07MCPC02** in partial fulfillment of the requirements for the award of **Doctor of Philosophy** in Computer Science, is a bonafide work carried out by him under 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.

Dr. Vadlamani Ravi
Supervisor
IDRBT
Hyderabad

Prof. Chillarige Raghavendra Rao
Supervisor
Department of CIS
University of Hyderabad

Prof. P. N. Girija
Head of the Department
Department of CIS
University of Hyderabad

Prof. T. Amaranath
Dean
School of MCIS
University of Hyderabad

DECLARATION

I am **Nekuri Naveen** hereby declare that this thesis entitled **"Rule Extraction from Neural Networks and Optimization Techniques: Application to Banking"** submitted by me under the guidance and supervision of **Dr. V. Ravi, Associate Professor, Institute for Development and Research in Banking Technology, Hyderabad** and **Prof. C. Raghavendra Rao, Department of Computer and Information Sciences, University of Hyderabad, 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: **November, 2012**

Signature of the Student

Name: **Nekuri Naveen**

Reg. No. 07MCPC02

DEDICATION

My Parents, my Brother and Sister, who offered me unconditional love and support throughout the course of PhD.

ACKNOWLEDGMENTS

First of all, I am very grateful to **GOD** 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. Also, I would like to thank my brother (Narendran) and sister (Nalini Devi) for their kind support.

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. I owe my greatest gratitude to my supervisors **Dr. V. Ravi**, Associate Professor, IDRBT, Hyderabad and **Dr. C. Raghavendra Rao**, 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.

In the beginning of my research, I am scared of Dr. Ravi's nature interms of strictness in work and monitoring the work 24 × 7. He helped me in lot many things some of them are, in research how to approach, think and analyze a problem. He gave advice in every aspect of my life from the day it starts and ends. I am also thankful for his strict nature which helped me to learn many things towards research and personal life. I am very much thankful to Mrs. V. Ravi for treating me as their family member and gave encouragement throughout my research work and advised me to take care of my health whenever I am sick by calling me over phone.

I am very much thankful to Prof. C. Raghavendra Rao for keeping confidence on me through out my research work. He also encouraged me whenever I started

a new work. I am greatly thankful to him in responding to my phone calls either despite of his busy schedule or in meetings. I learned many things from him in aspects in regards to professional and personal life. I am also thankful to him for helping me in many aspects and treated me as a family member. In all, both of my supervisors contributed a lot in building my confidence levels to tackle a research problem and as a good researcher.

It is an honor for me to thank **Dr. Siba Kumar Udgata, Reader** and **Dr. Rajeev Wankar**, for serving on my examining committee. I thank them for their precious time in reviewing my research progress and providing thoughtful comments and suggestions. I also thank **Mr. B. Sambamurthy, Director, IDRBT, Dr. Radha Binod Barman, RBI, Dr. Krishna Mohan, Former Director IDRBT, Prof. T. Amaranath, Dean, School of Mathematics Computers and Information Sciences, Prof. P. N. Girija, Head of the Department, Department of Computer and Information Sciences, University of Hyderabad and Prof. Arun Agarwal** 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 kind support. Among them are, Yenuga Padma, Yenuga Jyotsna, K. Praveen, Yugandhar Varma, Phani Chakradhar, Khuda Bakshi, Yerneni Prashant, Pilli Kishore, Gadepaka Latha, Sujatha Pilli, Velagapudi Haritha, Suresh Kumar Digumarthi, Dr. Vikram, Dr. Kishore, Dr. Chalapathi, Sareen, Prashanth, Dr. Seethapathi, Sudha Kishore, Nikunj, Anil Kumar, Ravi Shankar, Vasu, Jagan, Suresh, Anki Reddy, Mahesh, Paramjeet, Vishal Arora, Sridharan Reddy, Sarath, Nishanth, Narender, Siva, Venkatesh, Prudvi, Sridhar, Srilakshmi Durga, Swathi, Satyanarayana, Sirajuddin, Vasavi, Dr. Farquad, Shakeel Ahamad, Dr. Padmaja, Kavitha, Illiah, Srinu Naik, K. Hima Bindhu, Sai Prasad, Pradeep Dadabada, Srinivas Naik, Veerana, Ganashyam, Kalyan, Bharath, Anand, Dr. Gopal Krishna, Vikranth, etc. for their entertaining friendship, moral support and financial support to accomplish this thesis. I would like to express my special thanks to my colleagues Badugu Srinivas, Pavan and Sagar who has given me immense support

and help in writing this thesis in Latex. If I miss any name I apologize 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 would like to thank faculty of IDRBT, Prof. V. N. Sastry, Dr. M. V. N. K. Prasad, Dr. Mahil Carr, Dr. V. Radha, for the support they have given me during the stay at IDRBT. I also would like to thank the staff of IDRBT, Mr. Rashmi Dev, HR Manager, for sanctioning the amount towards registration fee for conferences whenever needed and Mr. T. Ashok Kumar, accounts for passing the papers with in time and T. Srilatha, who has given me printouts. Also, I would like to thank the staff of library, IDRBT, Mr. P. Ratna Kumar and Mr. D. Prakash, 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 required and requested. Last but not least, I would like to thank RKHS team as a whole for serving me the most hygienic and energetic food during my stay at IDRBT.

Although there are lots of other people whom I wish to thank and there are many of them whom I have missed. I sincerely apologize for that ...

... Thank you all!

Nekuri Naveen

ABSTRACT

Neural Networks have been used in data mining in solving classification and regression problems in a wide variety of real world problems (for classification and regression problems). Most popular Neural Network is MultiLayer Perceptron (MLP) using to its versatility and power. However, Radial Basis Function (RBF) network is equally versatile and powerful. Similarly, Group Method of Data Handling (GMDH) network and Wavelet Neural Network (WNN) also gained popularity due to the same reasons mentioned for the MLP and RBF. However, one common drawback that makes them less attractive and the lack of interceptability i.e., their black box nature. This is the root cause of these networks for being less attraction. Further, privacy preserving data mining (PPDM) is another important area of data mining. Neural networks can achieve PPDM. Hence, in this thesis all the four models, particle swarm optimization trained auto associative neural network (PSOAANN) for privacy preserving data mining, differential evolution trained radial basis function network (DERBF), group method of data handling (GMDH) and differential evolution trained wavelet neural network (DEWNN) are all converted into transparent by adding rule generation modules to them. Further, we also generated rules using firefly optimization algorithm.

The primary contribution of the thesis is the proposal of various algorithms to overcome the significant limitations of neural networks 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 neural networks, and proposing novel approaches. The contributions are grouped under three classes, decompositional, pedagogical and eclectic/hybrid approaches. Decompositional approach is closely intertwined with the internal workings of the neural networks. Pedagogical approach uses neural networks as an oracle to re-label training examples as well as artificially generated examples. In the eclectic/hybrid approach, combination of these decompositional and pedagogical methods is adopted.

This thesis investigates the efficiency of the proposed rule extraction approaches in solving a couple of analytical problems occurring in banking viz., (i) bankruptcy prediction in banks and (ii) churn prediction in credit card customers and also a host of standard benchmark classification and regression problems taken from literature. The benchmark problems include Iris, Wine, Wisconsin Breast Cancer, New Thyroid, US congressional, Auto MPG, Forest Fires, Pollution, Boston Housing, Body Fat data sets. For classification problems, various rule extraction methods such as Decision Tree (DT), Ripper and GATree are employed. Additionally for regression problems, rule extraction methods such as Dynamic Evolving Neuro Fuzzy Inference System (DENFIS) and Classification and Regression Tree (CART) have also been employed.

The prediction of bankruptcy for financial firms especially banks has been the extensively researched area with the application of statistical and machine learning techniques. Creditors, auditors, stockholders and senior management are all interested in bankruptcy prediction because it affects all of them. Bank management would be interested in the comprehensibility of the algorithms used for predictions and then banks would like to use them as early warning expert systems. We extracted crisp rules for bankruptcy prediction datasets and credit card churn prediction dataset and all the benchmark datasets using the above mentioned architectures.

Results are analyzed using Sensitivity, Specificity and Accuracy for classification problems and Mean Squared Error (MSE) for regression problems 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 ANN in order to extract rules by using the knowledge in form of weights and hidden output values as well as to other problem domains.

Research Publications out of Thesis

Journals:

(1) **Nekuri Naveen**, Vadlamani Ravi and C. Raghavendra Rao (2012) "Rule Extraction from Privacy Preserving Neural Network: An Application to Banking", ***Advanced Materials Research***, Vol: 403-408, pp: 920-928 *(indexed by SCOPUS).*

(2) Paramjeet, V. Ravi, **Naveen Nekuri** and C. Raghavendra Rao (2012) "Privacy Preserving Data Mining using Particle Swarm Optimization trained Auto-Associative Neural Network: an Application to Bankruptcy Prediction in Banks", ***International Journal of Data Mining Modeling and Management***, Vol:4, No: 1, pp: 39-56 *(indexed by DBLP).*

(3) **Nekuri Naveen**, Vadlamani Ravi, C. Raghavendra Rao and Nikunj Chauhan (2010) "Differential evolution trained radial basis function network: application to bankruptcy prediction in banks", ***International Journal of Bio-Inspired Computation***, Vol. 2, Nos. 3/4, pp. 222-232 *(indexed by DBLP).*

Conferences:

(1) **Nekuri Naveen**, Vadlamani Ravi, C. Raghavendra Rao and K. N. V. D. Sarath (2012) "Rule Extraction Using Firefly Optimization: Application to Banking", ***International Conference on Industrial Engineering and Engineering Management (IEEM)***, December, 2012, Hongkong (Accepted) *(indexed by DBLP).*

(2) **Nekuri Naveen**, Vadlamani Ravi and C. Raghavendra Rao (2012) "Rule Extraction from differential evolution trained wavelet neural network to Solve Classification and Regression Problems", ***Swarm, Evolutionary and Memetic Computing Conference (SEMCCO 2012)***, December, 2012, Bhubavenshwar, India (Accepted) *(indexed by DBLP).*

(3) **Nekuri Naveen**, Vadlamani Ravi and C. Raghavendra Rao (2011) "Forecasting Using Rules Extracted from Privacy Preservation Neural Network", ***Multi-disciplinary International Workshop on Artificial Intelligence (MIWAI 2011)***, pp: 248-260, December 7-9, 2011, Hyderabad, India *(indexed by DBLP).*

(4) **Nekuri Naveen**, Vadlamani Ravi and C. Raghavendra Rao (2010) "Data Mining via Rules Extracted from GMDH: An Application to Predict Churn in Bank Credit Cards", ***Knowledge-Based and Intelligent Information and Engineering Systems (KES 2010)***, Vol. 6276, pp.80-89, 8-10 September 2010, Cardiff, Wales, UK *(indexed by DBLP).*

(5) **Nekuri Naveen**, Vadlamani Ravi and C. Raghavendra Rao (2009) "Rule extraction from Differential Evolution trained Radial Basis Function Network using Genetic Algorithms", ***5th Annual IEEE Conference on Automation Science and Engineering (CASE 2009)***, August 22-25, Bangalore, India *(indexed by DBLP).*

Contents

CERTIFICATE **ii**

DECLARATION **ii**

DEDICATION **iv**

ACKNOWLEDGEMENTS **vii**

ABSTRACT **ix**

LIST OF FIGURES **xvi**

LIST OF TABLES **xviii**

1 Introduction to Rule Extraction **1**

1.1 Motivation to extract rules from ANN 2

1.2 Significance of Rule Extraction 2

1.2.1 Provision of user explanation capability 2

1.2.2 Transparency 3

1.3 Taxonomy of Rule Extraction 3

1.3.1 Decompositional Approaches 4

1.3.2 Pedagogical Approaches 5

1.3.3 Eclectic/Hybrid Approaches 6

1.4 Rule Quality Criteria 7

1.5 Datasets Analyzed 8

1.5.1 Classification Datasets 8

1.5.2 Regression Datasets 10

1.6 Experimental Setup 12

1.7 Outline of Thesis 12

2 Literature Review 15

2.1 Rule Extraction from Neural Networks 15

2.1.1 Decompositional Techniques 15

2.1.2 Pedagogical Techniques 18

2.1.3 Eclectic Techniques 21

2.2 Classification Rule Generation using evolutionary algorithms 22

2.2.1 Rule extraction using Genetic Algorithm 22

2.2.2 Rule generation using Ant Colony Optimization Algorithm (ACO) 23

2.2.3 Rule Extraction using Particle Swarm Optimization Algorithm 24

2.2.4 Rule Extraction using Differential Evolution Algorithm . . . 24

2.3 Outline of Proposed Approaches 25

Part I - Decompositional Approach 27

3 Rule Extraction from Auto Associtive Neural Network: a Hybrid Approach for Privacy Preserving Data Mining 28

3.1 Introduction 28

3.2 Literature survey 31

3.2.1 Perturbative methods 32

3.2.2 Non-perturbative methods 33

3.2.2.1 Data Swapping 33

3.2.2.2 K-Anonymity 33

3.2.2.3 Cryptographic Techniques 33

3.3 Proposed approach 34

3.3.1 Particle Swarm Optimization Trained Auto Associative Neural Network 34

3.3.2 Proposed Feature Selection Method 37

3.4 Datasets Analyzed 39

3.5 Results and Discussion 41

3.6 Conclusions 65

Part II - Pedagogical Approach 66

4 Rule Extraction from Differential Evolution trained Radial Basis Function Neural Networks in solving Classification problems 67

4.1 Introduction 67

4.1.1 Literature review in improving RBF 69

4.2 Proposed Hybrid Approach 70

4.2.1 DERBF - Training Algorithm 71

4.3 Datasets Analyzed 73

4.4 Results and Discussions 74

4.5 Conclusions 79

5 Rule Extraction from GMDH Neural Network: Application to Banking 80

5.1 Introduction 80

5.2 Customer Relationship Management 81

5.3 Churn Prediction Problem 82

5.4 Overview of Group Method of Data Handling 83

5.4.1 Description of algorithm 84

5.4.2 Application of GMDH Algorithm 86

5.5 Data Imbalance Problem 87

5.5.1 Literature Review of techniques Dealing with Unbalanced Data 87

5.5.2 Synthetic Minority Oversampling Technique 89

5.5.3 Random Undersampling 89

5.5.4 Random Oversampling 90

5.6 Proposed Rule Extraction Technique 90

5.7 Datasets Analyzed 91

5.8 Results and Discussion 92

5.9 Conclusion 96

Part III - Eclectic Approach 98

6 Rule Extraction from DEWNN for solving Classification and Regressions problems 99

6.1 Introduction 99

6.2 Proposed Rule Extraction approach . . . 100
6.2.1 Wavelet Neural Network . . . 101
6.2.2 Feature Selection . . . 104
6.3 Datasets Analyzed . . . 106
6.4 Results and Discussions . . . 107
6.5 Conclusions . . . 173

Part IV - Optimization Technique **174**

7 Rule Extraction from Firefly Optimization Algorithm for solving Classification problems **175**
7.1 Introduction . . . 175
7.2 Firefly Miner . . . 176
7.2.1 Fitness function . . . 177
7.3 Datasets Analyzed . . . 178
7.4 Results and Discussions . . . 178
7.5 Conclusions . . . 183

8 Overall Conclusions **185**

Appendix A Overview of Optimzation Algorithms **188**
A.1 Particle Swarm Optimization (PSO) . . . 188
A.2 Differential Evolution . . . 190
A.3 Firefly Optimization . . . 194
A.3.1 Algorithm . . . 195
A.3.2 Attractiveness . . . 195
A.3.3 Distance and Movement . . . 198

Appendix B Overview of Rule Extraction Methods **199**
B.1 GATree . . . 199
B.2 Decision Tree . . . 202
B.2.1 Classification . . . 203
B.2.2 Algorithm . . . 204
B.2.3 Splitting Rule . . . 207
B.3 Repeated Incremental Pruning to Produce Error Reduction (Ripper) 208

B.4 CART 210
B.4.1 Construction of maximum tree 211
B.4.2 Choice of the right tree size 213
B.4.3 Classification of new data using constructed tree 214
B.4.4 Advantages of CART 214
B.4.5 Disadvantages of CART 215
B.5 Dynamic Evolving Fuzzy Inference Systems (DENFIS) 215
B.5.1 Evolving clustering method (ECM) 215
B.5.2 Online ECM 216
B.5.3 ECM Algorithm 216
B.5.4 DENFIS 217

Appendix C Datasets Description **219**

REFERENCES **227**

List of Figures

1.1 Various Categories of Rule Extraction Approaches 5
1.2 Area Under ROC. 8
1.3 Experimental Setup 12

2.1 Classification of the proposed approaches for Rule Extraction from Neural Networks 26

3.1 PSOAANN 36
3.2 Proposed Feature Selection 39
3.3 Proposed approach data flow diagram 40

4.1 DERBF Architecture 71
4.2 DERBF + GATree data flow diagram 73

5.1 GMDH Architecture 86
5.2 GMDH + DT data flow diagram 91

6.1 DEWNNRE data flow diagram 101
6.2 Wavelet Neural Network 103

A.1 Flow chart of Particle Swarm Optimization 190
A.2 Flow chart of Differential Evolution 193
A.3 Flow chart of Firefly Algorithm 196

B.1 GATree Mutation examples 200
B.2 GATree Crossover examples 201
B.3 Example of Decision Tree for Bankruptcy prediction in Banks . . . 203
B.4 Three possibilities for partitioning objects based on the splitting criterion 206

B.5 CART AutoMPG example 211
B.6 Splitting Algorithm of CART 212

List of Tables

1.1 Datasets Dimensionality 11

3.1 Important features selected in classification datasts 42
3.2 Important features selected in regression datasts 43
3.3 Average accuracies of PSOAANN + Decision tree 44
3.4 Average accuracies of PSOAANN + Decision tree over reduced features 44
3.5 Average accuracies of PSOAANN + Ripper 45
3.6 Average accuracies of PSOAANN + Decision tree over reduced features 45
3.7 t-statistic values computed 46
3.8 Average RMSE values over 10-folds of PSOAANN + DENFIS . . . 51
3.9 Average RMSE values over 10-folds of PSOAANN + CART 52
3.10 Average rule base size of full and reduced features 52

4.1 Average results of 10FCV of DERBF + GATree 75

5.1 Average accuracies for benchmark datasets 93
5.2 Average accuracies for churn prediction dataset 95
5.3 Average rule base size of DT and proposed hybrid approach for the best fold 96

6.1 Important features selected in classification datasets 108
6.2 Important features selected in regression datasets 109
6.3 Average results of 10FCV of full features datasets for classification problems 110
6.4 Average results of 10FCV of reduced features datasets for classification problems 114

6.5 Average MSE values of 10FCV of full features datasets for regression problems . . . 118
6.6 Average MSE values of 10FCV of reduced features datasets for regression problems . . . 118
6.7 Average rules of 10FCV for classification problems . . . 168
6.8 Average rules of 10FCV for regression problems . . . 169
6.9 t-test values between DEWNN and Hybrids of classification problems for full features . . . 170
6.10 t-test values between DEWNN and Hybrids of classification problems for reduced features . . . 171
6.11 t-test values between DEWNN and Hybrids of regression problems for full features . . . 171
6.12 t-test values between DEWNN and Hybrids of regression problems for reduced features . . . 172
6.13 t-test values between DERBF + GATree and DEWNN + DT / Ripper of classification problems for full features . . . 173

7.1 Average results of firefly miner . . . 179
7.2 Average results of Decision Tree . . . 180

C.1 Financial Ratios of Spanish Banks Dataset . . . 219
C.2 Financial Ratios of Turkish Banks Dataset . . . 220
C.3 Financial Ratios of US Banks Dataset . . . 220
C.4 Financial Ratios of UK Banks Dataset . . . 221
C.5 Feature description of Churn Prediction Dataset . . . 222
C.6 Feature description of Auto MPG Dataset . . . 223
C.7 Feature description of Body Fat Dataset . . . 223
C.8 Feature description of Boston Housing Dataset . . . 224
C.9 Feature description of Forest Fires Dataset . . . 225
C.10 Feature description of Pollution Dataset . . . 226

Chapter 1

Introduction to Rule Extraction

There has been a great interest in the area of data mining in which the general goal is to discover knowledge that is not only correct, but also comprehensible and interesting for the user. Data mining has been defined as the nontrivial extraction of implicit, previously unknown and potentially useful information from data [Frawley et al., 1992]. Data Mining encompasses a number of different technical approaches such as clustering, data summarization, learning classification rules, finding dependency networks and detecting anomalies.

Artificial neural networks (ANN) have been used during the last few decades in a wide variety of applications. It is often useful to have a symbolic representation of the rule base or the function calculated by the network. When the application is decision support, like in classification or clustering problems, it is often desirable to understand how the ANN determines its decision. Such understanding is often desired in areas such as data mining and financial engineering.

Over the last three decades, data mining and machine learning techniques 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) are the most successful machine learning techniques applied in the area of data mining which 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 Motivation to extract rules from ANN

Artificial neural networks (ANN) have proved to be good machine learning techniques specifically for solving classification and regression problems. However they are treated as blackbox 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, bankruptcy prediction, churn 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. Much attention has been paid during last decades to find effective ways of extracting rules from ANN.

1.2 Significance of Rule Extraction

Andrews et al., [Andrews et al., 1995] describe the motivation behind rule extraction from neural networks. A brief review of the arguments of Andrews et al., [Andrews et al., 1995] will help to establish aims and significance for rule extraction from ANN 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 [Gallant, 1988] observes that an explanation capability enables a novice user to gain insights into the problem at hand. Davis et al., [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, 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 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].

1.3 Taxonomy of Rule Extraction

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 ANNs. Rule extraction from neural networks has previously almost exclusively been used to generate propositional rule sets [Nayak et al., 2007]. 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 [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 [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 behavior.

Over the last few years, a number of studies on rule extraction from ANN 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 ANN 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.

1.3.1 Decompositional Approaches

The decompositional algorithms extract rules by decomposing the ANN and extracting rules from each unit in ANN and aggregate them. Towell and Shavlik [Towell and Shavlik, 1993] focused on extracting symbolic rules from the trained feed-forward 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 [Arbatli and Akin, 1997] extracted rules form neural network using Genetic Algorithm.

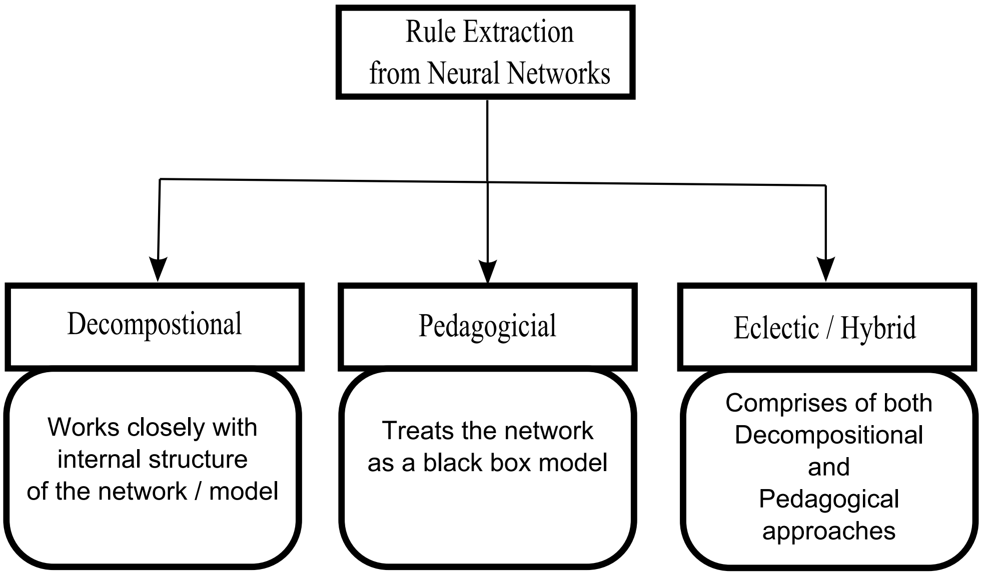

Figure 1.1: Various Categories of Rule Extraction Approaches

Craven and Shavlik [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 techniques treat the network as a black box [Clark and Niblett, 1989]; [Craven and Shavlik, 1996] and make no attempt to disassemble its architecture to examine how it works, instead this approach extracts rules by examining the relationship between the inputs and outputs. These techniques typically use trained ANN 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 [Saito and Nakano, 1988] employs a breadth-first search process to extract conjunctive rules in binary problem domains. Gallant [Gallant, 1988] developed a similar rule extraction technique, which like the method of Saito and Nakano [Saito and Nakano, 1988], 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 [Saito and Nakano, 1988] method, Gallant's [Gallant, 1988] rule-testing procedure is guaranteed to accept only rules that are valid.

Thrun [Thrun, 1995] developed a method called validity interval analysis (VIA) that is a generalized and more powerful version of Gallant's [Gallant, 1988] 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., [Krishnan et al., 1999] proposed a rule extraction approach which extracts crisp rules from the neural network.

1.3.3 Eclectic/Hybrid Approaches

Eclectic rule extraction techniques, which incorporate elements of both the decompositional and pedagogical approaches. Sato and Tsukimoto [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 [Campos and Ludermir, 2005] presented Literal and ProRulext algorithms to extract rules from the trained artificial neural network. Aliev et al., [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.

$$Accuracy = \frac{\text{\# Samples Correctly Classified by Rules}}{\text{Total \# samples in Test Data}} \times 100$$

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

$$Sensitivity = \frac{\text{\# Positive Samples Correctly Classified as Positive by Rules}}{\text{Total \# Positive Samples in Test Data}} \times 100$$

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

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 has long been used in signal detection theory to depict the trade-off between hit rates and false alarm rates of methods. The ROC graph is a two dimensional graph which represents various methods based on their output results in the point form in a region, which has FP rate (1-Specificity) on the X-axis and TP rate (Sensitivity) on the Y-axis. ROC graphs are a very useful tool for visualizing and evaluating methods. They are able to provide a richer measure of classification performance than scalar measures such as accuracy, error rate or error cost [Fawcett, 2006]. The AUC of a method A can be calculated as the sum of the areas of Triangle-CEA, Rectangle-EAFI and Triangle-FAH as depicted in Figure 1.2. Finally the comprehensibility of a rule set is determined by measuring the size of the rule set (in terms of number

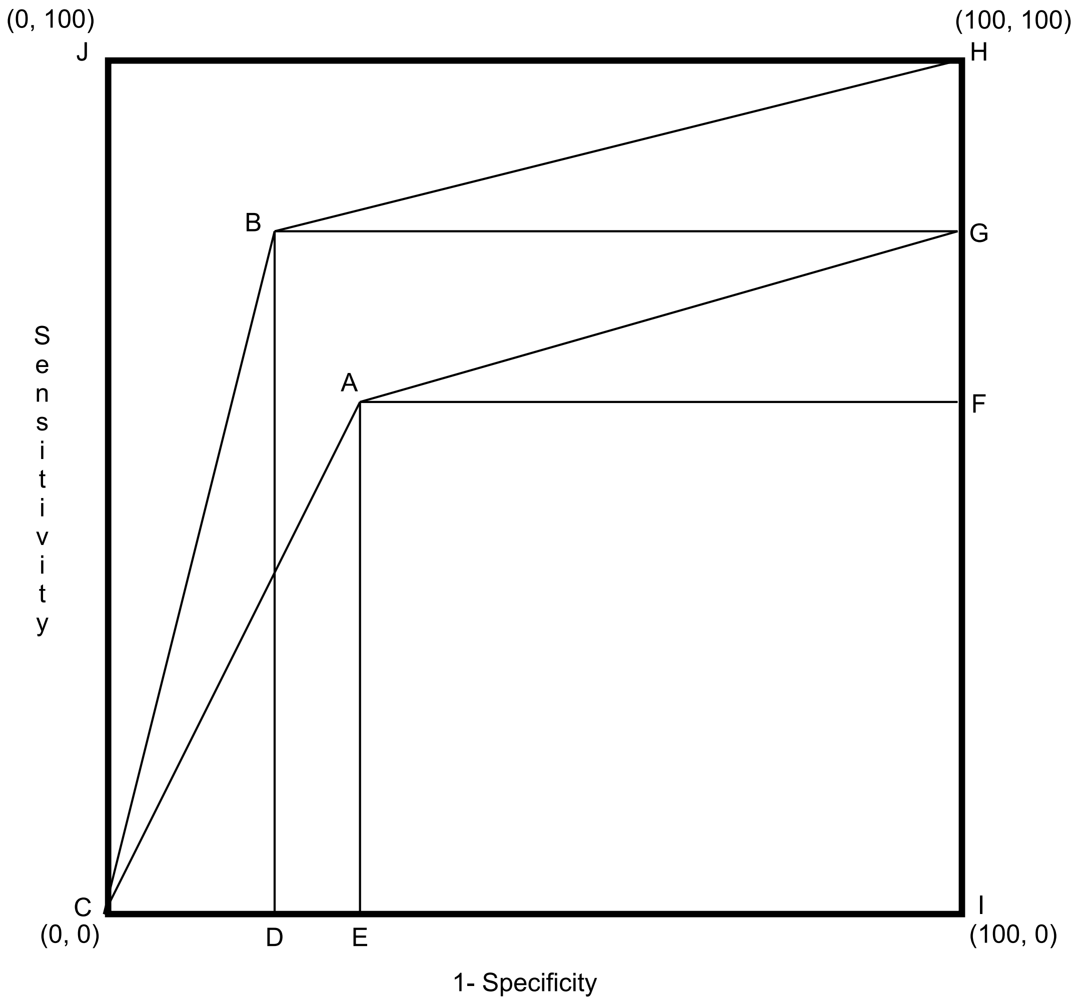

Figure 1.2: Area Under ROC.

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 Datasets Analyzed

1.5.1 Classification Datasets

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

Benchmark datasets

Iris dataset, Wine dataset and US Congressional dataset and Medical Diagnosis dataset i.e., Wisconsin Breast Cancer (WBC), New Thyroid are downloaded from UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`).

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 Fernndez, 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 et al., 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 et al., [Canbas et al., 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. Later they [Canbas et al., 2005] found that only 12 ratios in Table C.2 in Appendix C, act as early warning indicators that have the discriminating ability (i.e. significance level is l5% in ANOVA) for healthy and failed banks, one year in advance. Among these variables, 12th 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 [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 behavioral data, which are described in Table C.5 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 non-churners and 1002 are churners, which means there are 93.24% non-churners and 6.76% churners. Hence, the dataset is highly unbalanced in terms of the proportion of churners versus non-churners (Business Intelligence Cup 2004) [Chile, 2004].

1.5.2 Regression Datasets

AutoMPG 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.6 in Appendix C.

Body Fat 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 et al., 1985]. Its features are described in Table C.7 in Appendix C.

Boston Housing 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.8 in Appendix C presents the feature description.

Forest Fires 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.9 in Appendix C.

Pollution 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.10 in Appendix C presents the description of features.

The dimensionality of all the datasets are presented in Table 1.1.

Table 1.1: Datasets Dimensionality

Dataset	Total instances	Features	# of Classes
Classification			
IRIS	150	4	3
Wine	178	13	3
WBC	683	9	2
New Thyroid	215	5	3
US Congressional	435	16	2
Spanish Banks	66	9	2
Turkish Banks	40	12	2
US Banks	129	5	2
UK Banks	60	10	2
Churn Prediction	14814	21	2
Regression			
AutoMPG	392	7	Not Applicable
Bodyfat	252	14	- do -
Boston Housing	506	13	- do -
Forest Fires	517	12	- do -
Pollution	60	15	- do -

1.6 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.

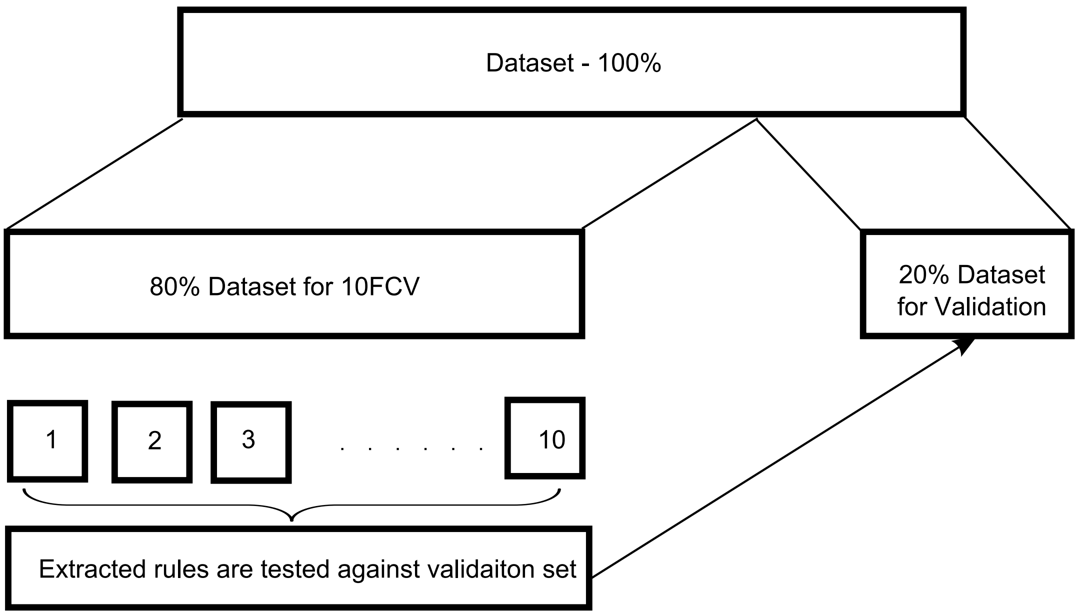

Figure 1.3: Experimental Setup

1.7 Outline of Thesis

The remaining chapters of the thesis are organized as follows:

Chapter 2 provides the literature review of the approaches in extracting *If-then* rules from ANN. 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 rule extraction approaches using various optimization algorithms. Final section provides outline of the proposed approaches.

Chapter 3 presents the proposed *decompositional* rule extraction approach from privacy preserving auto-associative neural network which is one of the main contributions of this thesis. First section of the chapter provides introduction and privacy preservation methods are presented. In third and fourth section the proposed approach and datasets analyzed are presented which is followed by Results and discussion in the fifth section. Section six concludes the chapter.

Chapter 4 presents a *pedagogical* rule extraction approach proposed which uses ANN as a *black box*. With the introduction and literature review in improving radial basis function network in the first section, the proposed hybrid approach is presented in detail in next section. In third section datasets analyzed is presented. Following section provides Results and discussions. Final section concludes the chapter.

Chapter 5 presents a *pedagogical* rule extraction approach proposed which analyzes medium scale dataset pertaining to finance. With the introduction of customer relationship management of first section followed by churn prediction problem. In fourth section, proposed rule extraction method and then in fifth section description about the group method of data handling presented. Data description, data imbalance problems and datasets analyzed are presented in section six, seven and eight. finally results and discussion and conclusions of the chapter are presented in ninth and tenth sections.

Chapter 6 presents an *eclectic* rule extraction approach proposed for solving classification and regression problems. First section presents the introduction about the wavelet followed by proposed rule extraction using wavelet neural network and feature selection. In third section, datasets analyzed by the proposed method are presented. Results and discussions are presented in fourth section. Finally section concludes the chapter.

Chapter 7 presents rule extraction from firefly optimization algorithm in solving classification problems. First section presents introduction followed by the firefly miner algorithm in second section. Dataset analyzed by firefly miner are presented in section three. Section four presents the results and discussions. Final section concludes the chapter. Finally, Chapter 8 presents the overall conclusion and contributions of this thesis.

Chapter 2

Literature Review

This Chapter provides the literature survey of the approaches proposed for extracting rules from Neural Networks. We implemented a two phase approach for constructing an expert systems (Rule Generation) from modified training examples. The first model is a network knowledge base generator and the second model is a stand alone expert system inference engine that interprets such knowledge bases in to the form *if-then* rules.

The first section describes in detail about the rule extraction approaches proposed in decompositional, pedagogical and eclectic/hybrid category. Next section about the classification rule generation using evolutionary algorithms.

2.1 Rule Extraction from Neural Networks

In this section, an overview of rule extraction methods. The rule extraction approaches are categorized into three types based on the neural network components utilized for rule extraction.

2.1.1 Decompositional Techniques

Decompositional rule extraction algorithms directly interpret the response of each node in the network, sometimes assigning linguistic meaning to the nodes. Rules are extracted by analysing the activations of hidden and output nodes and the weights attached to them.

Hayashi and Imura [Hayashi and Imura, 1990] extracted rules from the fuzzy feed forward neural network. Input layer consists of fuzzy cell gourps and crisp cell groups. Extraction of fuzzy *if-then* with the inputs from the fuzzy feed forward neural network and the actual outputs of the training dataset. Mcmillan et al., [McMillan et al., 1991] proposed RuleNet algorithm for extracting rules from the neural network. Fu [Fu, 1994] extracted rules from the neural networks. They presented an algorithm called knowledgetron (KT). Arbatli and Akin [Arbatli and Akin, 1997] use a genetic algorithm to optimize an ANN topology, and then extracted rules base from the relevant inputs from the input domain of the topology used. Finally conjunctive rules are extracted using the relevant inputs. McGarry et al., [McGarry et al., 1999] proposed a rule extraction approach to extract rules from Radial Basis Function Networks. Based on the hidden neurons of the network. They used the maximum and minimum of the attributes values in each of the cluster to form a rule. RULEX rule extraction given by Andrew and Geva [Andrews and Geva, 2002], RULEX is capable of extracting symbolic rules from weights of trained local cluster neural networks which are accurate and comprehensible propositional rules. Saito and Nakano [Saito and Nakano, 2002] extracted regression rules from trained neural network trained with multivariate data containing both nominal and numeric variables. After generating a regression rule for each training sample, they performed k-means algorithm to generate a much smaller set of rules having more general conditions, in which the number of distinct polynomial equations is determined through cross-validation. Finally, this method invokes decision tree induction to form logical formula of nominal conditions as consequent parts of final regression rules.

Setiono and Thong [Setiono and Thong, 2004] proposed rule extraction from ANNs that have been trained to solve regression problems. The extracted rules divide the data samples into groups.A linear function of the relevant input attributes of the data approximates the network output of all the samples in a group. Elalfi et al., [Elalfi et al., 2004] presented an algorithm for extracting accurate and comprehensible rules from databases via trained ANNs by using genetic algorithms. Their algorithm does not depend on the ANN training algorithms and does not modify the training results. They used genetic algorithms to find the optimal values of input attributes, which maximize the output function of the output nodes. They also decoded the optimal chromosome and used it to get a rule which belongs to the target class. Tokinaga et al., [Tokinaga et al., 2005] utilized the GP to automatize the rule extraction process in the trained neural networks where the statements changed into a binary classification. They utilized genetic programming to automate the rule extraction process. Wang and Fu [Wang and Fu, 2005] extracted rules from the radial basis function (RBF) neural network. First, they classified the data using RBF. During training, they allowed the overlap between the clusters of the same class in order to reduce the hidden neurons without compromising the accuracy. Secondly the centers of the kernel function are used as initial conditions when searching for rule premises by gradient descent algorithm. Lastly, they removed the redundant and unimportant rules based on the rule tuning by maintaining the accuracy. They generated rules for prediction of bankruptcy and creditworthiness for binary classifications and applied their method to multi-level classification of corporate bonds by using the financial indicators.

Tan et al., [Tan et al., 2007] proposed a hybrid neural network model, based on the integration of fuzzy ARTMAP and the rectangular basis function network (RecBFN), which is capable of learning and revealing fuzzy rules. Rules are directly extracted from the network weights for justifying its predictions and applied for fault detection and diagnosis task in a power generation station. Anbananthen et al., [Muthu Anbananthen et al., 2007] proposed Artificial Neural Network Tree (ANNT), i.e. ANN training preceded by Decision Tree rules extraction method, which is presented to overcome the comprehensibility problem of ANN. ANNT was motivated by the lack of an intelligent procedure for interpreting the knowledge learned by ANN and utilizing it as a data mining tool. ANNT use DT to approximate the function of the trained ANN to improve the comprehensibility of ANN. Ozbakir et al., [Ozbakir et al., 2008] presented a study on knowledge acquirement from trained ANNs for classification problems. Their method uses ant colony optimization algorithm for extracting accurate and comprehensible rules from trained ANNs. Amora et al., [Amora et al., 2009] extracted fuzzy rules from the trained neural network. the knowledge of the neural network associated with the connections and synaptic weights can be expressed as fuzzy rules which allows a better understanding of the knowledge gained by the trained neural network.

2.1.2 Pedagogical Techniques

Pedagogical rule extraction algorithms treat the network as a black box in that they are not concerned with the internal structure of the network. The extracted rules reflect the relationship between the input variables and the outputs, without the concern of the nodes in the network.

Carpenter and Tan [Gail A. and Ah-H Tee, 1995] extracted knowledge from Fuzzy ARTMAP in the form of fuzzy rules. Rule extraction proceeds in two stages: pruning and quantization of continuous leaned weights. Craven [Craven, 1996] proposed rule extraction process as an inductive learning problem called as TREPAN. It uses oracle queries to induce an M-of-N decision tree that approximates the concept represented by a given network by examining the relationship between the inputs and outputs. Mitra et al., [Mitra et al., 1997] extracted knowledge in the form of rules from fuzzy multilayer perceptron in two methods. In method (i), treating the network as a black-box and using the predicted outputs of the network along the with the inputs to *if-then* generate rules. In method (ii), backtracking along maximal weights paths using the trained network and utilizing the inputs and outputs of the network in order to generate the *if-then* rules. Setiono [Setiono, 1997] presented an algorithm for extracting rules from feedforward neural network. The trained network is first pruned inorder to remove redundant connections in the network and detected the relevant inputs. Discretized the activation values of the hidden units by clustering. By using the discretized values generate rules that describe the network outputs. Generate rules between each hidden unit in the hidden layer and input in the input layer. Finally merge these two set of rules that relate the inputs and outputs of the network. Krishnan et al., [Krishnan et al., 1999] extracted rules form the feed forward neural network. They presented an algorithm called COMBO for rule extraction with boolean inputs. The search methods of rule extraction from neural networks is by finding the correct combination of inputs. They COMBO algorithm works in three stages. In the first stage, extraction of rules for the output layer neurons. In the second stage, extraction of rules for the hidden layer neurons. In the final stage, combining the rules in the first stage and second stages in order to obtain a relationship between the input and outputs.

Setiono [Setiono, 2000a] extracted M-of-N rules from trianed feedforward neural network. Initially Setiono trained and pruned the network. Then clustered the hidden unit activation values of the pruned network and generated classification rules in terms of the clustered activation values. Setiono [Setiono, 2000b] proposed a method called NeuroRule which produces concise and accurate classification rules. The overview of NeuroRule is, the fully trained network is pruned without compromising its accuracy. Then discretized the hidden unit activation values of the pruned network by clustering. then extract rules that describes network outputs in terms of the discretized hidden unit activation values. Also, generate rules that describe the discretized hidden unit activation values in terms of the network inputs. Finally merge two sets of rules generated to obtain a set of rules that related the inputs and outputs of the network. Mitra and Hayashi [Mitra. and Hayashi, 2000] extracted rules using neuro-fuzzy system. They extracted rules from the neural network using the network parameters in the process. Garcez et al., [d'Avila Garcez et al., 2001] presented a method to extract non-monotonic rules from artificial neural networks formed by discrete input units. Fan and Li [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. Fujimoto and Nakabayashi [Fujimoto and Nakabayashi, 2003] extracted rules from group method of data handling. The advantage of their method are it accepts both categorical and continuous data at the same time and rules are extracted easily from the generated model. Markowska-Kaczmar and Trelak [Markowska-Kaczmar and Trelak, 2005] extracted fuzzy rules from trained neural network. They fed the training set to the neural network and taken the outputs and actual inputs of the neural network. Evolution algorithm called genetic algorithm is being used to encoded the input features and extracted fuzzy rules. They used two types of approaches, (i) Pittsburgh approach and (ii) Michigan approach and called the proposed method as REX Pitt and REX Michigan..

Etchells and Lisboa [Etchells and Lisboa, 2006] proposed a rule extraction method which is suitable for complex neural networks. The proposed method does not explicitly utilize the network structure, but depend on its inputs and outputs of the network. Guo et al ., [Guo et al., 2007] extracted understandable and concise rules from trained neural networks. They trained the network and pruned the redundant connections. Then discretized the hidden unit activation values of the hidden units and then extracted rules for the inputs and discretized hidden activation values. Then extracted rules for the outputs of the network interms of the discretized hidden activation values. Now, they merged two sets of rules to obtain the final set of rules which contains the discretized values and continuous values. Saad and Wunsch [Saad and II, 2007] proposed a new explanation algorithm which relies on network inversion. They present a new explanation algorithm which relies on network inversion; i.e. calculating the ANN input which produces a desired output. Their algorithm is a pedagogical algorithm that extracts rules from ANNs, which extracts hyperplane rules from continuous or binary attribute neural networks. Jian-guo et al., [Jian-guo et al., 2008] extracted rules from artificial neural network with optimized activation functions. Weight-decay approach is used in training the network and the unnecessary connections in the neural network are pruned at the cost of an increase in the error function within a predetermined limit. The algorithm is desgined as follows: Firstly, the rules are extracted from hidden layer to output layer. Secondly, rules are extracted from input layer to hidden layer. Finally, the rules from input layer to output layer can be obtained by merging the two parts rules. Mohamed [Mohamed, 2011] extracted rules from constructively trained neural networks based on genetic algorithms. He used genetic algorithms for encoding the inputs attributes in the input layer before the network is trained. Each encoded input record represent a chromosome in genetic algorithm. Genetic algorithm has to find a set of nontrivial rules hidden in the subsets associated to the data. Each individual in a population represents single rule.The individuals that make part of each generation are selected accordingly.The crossover operator is performed on the selected individuals to create the new individuals of the next generation. Mutation is performed by randomly changing the value of a gene. After applying the genetic algorithm operators and number of generations we get the best chromosomes which represent the rules. Bhalla et al., [Bhalla et al., 2012] proposed a method that derives linear equations by approximating the hidden unit activation function and splitting the input feature space into subregions. For each subregion there is a linear equation. By using these linear equations they extracted *if-then* rules from the trained neural network.

2.1.3 Eclectic Techniques

Setiono and Leow [Setiono and Leow, 2000] extracted rule by using FERNN a fast method of extracting rules from trained neural networks. consists of two main components. The first is a network training algorithm that minimizes a cross entropy error function augmented by a penalty function. The minimization of the augmented error function ensures that connections from irrelevant inputs have very small weights. Such connections can be removed without affecting the network's classification accuracy. The second component is a decision tree generating algorithm which generates a tree classifier using the activation values of the network's hidden units. Hruschka and Ebecken [Hruschka and Ebecken, 2006] proposed a method to extract rules from multilayer perceptrons trained in classification problems. The rule extraction algorithm basically consists of two steps. First, a clustering genetic algorithm is applied to find clusters of hidden unit activation values. Then, classification rules describing these clusters, in relation to the inputs are generated. Kahramanli and Allahverdi [Kahramanli and Allahverdi, 2009] developed new adaptive activation function and a new method for rule extraction from trained neural networks using artificial immune systems. This algorithm takes all input attributes into consideration and extracts rules from the trained neural network with adaptive activation function efficiently. Ozbakr et al., [zbakr et al., 2010] proposed a soft computing based approach for integrated training and rule extraction from artificial neural networks and named it as DIFA-

CONN - miner. The main idea behind the proposed miner is to use DE algorithm for training ANNs and TACO algorithm for extracting classification rules simultaneously. Classification rules can be directly obtained from ANNs without needing an additional step for rule extraction. This means that at every ANN training step the corresponding classification rules are simultaneously generated/evaluated and training is tried to be achieved for generating more accurate classification rules. The miner is composed of three interdependent parts which are categorized as training of feed forward ANNs with DE algorithm, rule extraction by TACO algorithm and then using fitness evaluation. Kulluk et al., [Kulluk et al., 2012] developed a soft computing based algorithm to generate fuzzy rules based on a data mining tool (DIFACONN-miner), which contains both categorical and continuous attributes. In the proposed approach the continuous valued attributes are fuzzified by using triangular membership function and categorical attributes are directly coded. Training and rule extraction phases of proposed algorithm are integrated within a multiple objective evaluation framework for generating fuzzy classification rules directly.

2.2 Classification Rule Generation using evolutionary algorithms

Many approaches, methods and goals have been tried out for Classification rule generation using Evolutionary approaches such as Genetic Algorithms and swarm-based approaches like Ant Colonies.

2.2.1 Rule extraction using Genetic Algorithm

Firstly, Mahfoud and Mani [Mahfoud and Mani, 1996] used GA and extracted rules to predict the performance of individual stocks. Shin and Lee [Shin and Lee, 2002] used Genetic algorithm for bankruptcy prediction modeling. An advantage of this approach using GAs is that it is capable of extracting rules that are easy to understand for users like expert systems. Kim and Han [Kim and Han, 2003] proposed genetic algorithm-based data mining method for discovering bankruptcy decision rules from experts qualitative decisions. This method is a suitable tool for eliciting and representing experts decision rules and thus it provides effective decision supports for solving bankruptcy prediction problems. The fitness function of the GA is the composite measure to discover decision rules that satisfy two different conditions: accuracy and coverage.

2.2.2 Rule generation using Ant Colony Optimization Algorithm (ACO)

Parpinelli et al., [Parpinelli et al., 2002] firstly proposed ACO based Ant-Miner algorithm for extracting classification rules from data. Ant-Miner discovered rules referring only to nominal attributes. Continuous attributes has to be discretized. In the initial population, Ant-Miner using the entropy measures has more the quality of the rules than a GA algorithm generating the initial population at random. Ant-Miner adopts the normalized information-theoretic heuristic function which computed the entropy for an attribute-value pair only. Later, Ji et al., [Ji et al., 2006] presents an enhanced Ant Miner, which includes two main contributions. Firstly, a rule punishing operator is employed to reduce the number of rules and the number of conditions. Secondly, an adaptive state transition rule and a mutation operator are applied to the algorithm to speed up the convergence rate. Then, Holden and Freitas [Holden and Freitas, 2007] proposed a hybrid PSO/ACO algorithm for discovering classification rules. In PSO/ACO, the rule discovery process is divided into two separate phases. In the first phase, ACO discovers a rule containing nominal attributes only. In the second phase, PSO discovers the rule potentially extended with continuous attributes. Ozbakir et al., [Ozbakir et al., 2009] [zbakr et al., 2011] proposed Touring Ant Colony Optimization based rule miner (TACO miner) for rule extraction from artificial neural networks (ANN). The proposed rule extraction algorithm actually works on the trained ANNs in order to discover the hidden knowledge which is available in the form of connection weights within ANN structure.

2.2.3 Rule Extraction using Particle Swarm Optimization Algorithm

DeSousa et al., [DeSousa et al., 2003] opted for the Constricted Particle Swarm Optimization (CPSO) variant, for it proved to be more qualified when dealing with continuous attributes, which was one of the goals. Temporal complexity was another concern, for without optimization in this area, expansion to more demanding problems is seriously affected or even made impossible. They first normalized the data as a pre-processing step and then generated rules from the normalized data using PSO. Later, Liu et al., [Liu et al., 2004] used PSO for extract the classification rules from data base. The potential if-then rules are encoded into real-valued particles that contain all types of attributes in data sets. Rule discovery task is formulated into an optimization problem with the objective to get the high accuracy, generalization performance, and comprehensibility, and then PSO algorithm is employed to resolve it. The advantage of the this approach is that it can be applied on both categorical data and continuous data. Zhao et al., [Zhao et al., 2006] uses binary version PSO based algorithm for fuzzy classification rule generation, also called fuzzy PSO. Classification rule generation problem is abstracted as a multiple objectives optimization problem. Namely the candidate rule should be accurate, general and interesting according to the sample data set. Then a fitness function, as a criterion to evaluate the strength of a certain rule, is designed. On this point, their algorithm has common background with the traditional evolutionary algorithm for knowledge discovery. The fitness function consists of three parts which are called accuracy, coverage and interestingness separately. It should be mentioned that in the interestingness part, we generalize the objective measure for the interestingness of a crisp classification rule to that for a fuzzy rule.

2.2.4 Rule Extraction using Differential Evolution Algorithm

The quantum-inspired differential evolution algorithm (QDE) is a new optimization algorithm in the binary valued space. Su et al., [Su et al., 2010] proposed DE/QDE algorithm for the discovery of classification rules. DE/ QDE combine the characteristics of the conventional DE algorithm and the QDE algorithm. Based on some strategies of DE and QDE, DE/QDE can directly cope with the continuous, nominal attributes without discretizing the continuous attributes in the preprocessing step. DE/QDE also has specific weight mutation for managing the weight value of the individual encoding.

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 viz., bankruptcy prediction in banks, churn prediction in credit card customers. Further, regression problems are also solved using the proposed approaches. In this thesis using decompostional approach, bankruptcy prediction and regression problems are solved. Using pedagogical approach bankruptcy prediction and churn prediction problems are solved. And using eclectic approach, bankruptcy prediction and regression problems are solved. Figure 2.1 present the classification of the proposed approaches into *decompositional*, *pedagogical* and *eclectic*. Finally, bankruptcy prediction problems are solved using firefly miner.

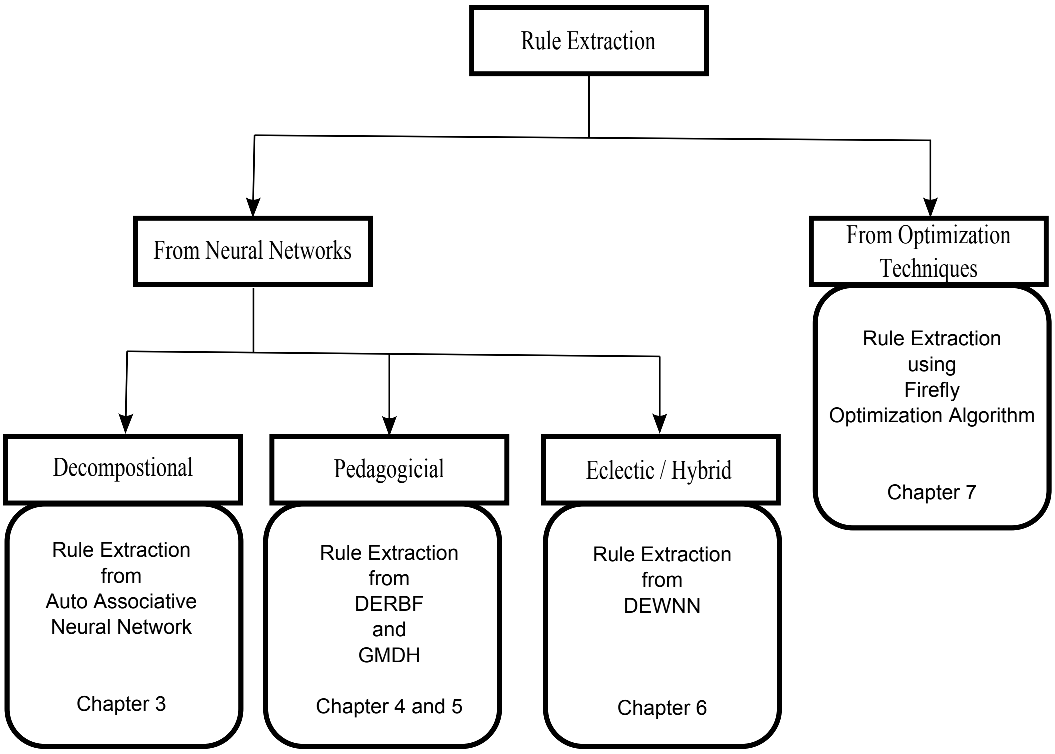

Figure 2.1: Classification of the proposed approaches for Rule Extraction from Neural Networks

Part I - Decompositional Approach

Chapter 3

Rule Extraction from Auto Associtive Neural Network: a Hybrid Approach for Privacy Preserving Data Mining

This Chapter presents a decompositional rule extraction approach from a novel 3-layered auto associative neural network (AANN) which is one of the main contributions of this thesis. The advantage of using a 3-layered AANN for privacy preservation over the traditional five layered AANN is (i) decrease in computations (ii) the complexity (iii) simpler to understand and implement. First section of the chapter provides introduction and privacy preservation methods. In third section the proposed auto associative neural network and the novel feature selection method are presented. In fourth section the datasets analyzed are discussed, which is followed by Results and discussion in the fifth section. Section six concludes the chapter.

3.1 Introduction

In recent years, advancement in the hardware and software technology has led to increase in the capability to store and record data pertaining to individual criminal records, health records, shopping habits, credit records. Today's globally networked society demand for sharing of large amount of sensitive data. Such data is an important for business organizations and governments for decision making and social benefits. This has led to concerns that the data may be misused for a variety of purposes like violating individual privacy. This has a reaction to a trend on the technology of Data Mining (DM). Various articles appeared in literature and in some countries even some acts have been passed.

Directive on Privacy Concerns (Oct 1998) and Canadian Personal Information Protection act (Jan 2001) to safeguard the interest and privacy of people. PPDM term was introduced by Agrawal and Srikant [Agrawal and Srikant, 2000] and Lindell and Pinkas [Lindell and Pinkas, 2000]. Issues related to PPDM were first discussed by national statistical agencies collecting private social and economical data such as census, tax record and making it available for analysis by public servants, companies and research institutes. There had been many developments in the field of cryptography, information hiding and data mining to address the issue regarding the privacy preservation. As a result, new class of data mining methods, known as Privacy Preserving Data Mining methods were developed by researchers working on security and DM [Xiao-dan et al., 2006].

Privacy means freedom from unauthorized intrusion and as per many laws it applies only to individually identifiable data. PPDM means ensuring privacy before actually mining database. Data mining ideally refers to the extracting knowledge from the datasets without leaking individual information i.e. building a model extract knowledge without any access to the precise information in the individual information records. The goal of PPDM is to effectively deal with the trade-off viz., successfully transforming/perturbing/distorting original dataset on one hand and then mining the resultant modified dataset which may result in a marginal reduction in accuracies due to the perturbation introduced in the datasets. In other words, we should perturb datasets such that on the one hand the privacy in the datasets is preserved and on the other hand there should not be a significant decline in the accuracy of data-mining results. There is a growing body of literature on data mining techniques that try to extract patterns from the modified data without directly accessing the original data. The idea is to guarantee that the mining process gets insufficient information to reconstruct the original data that might be misused resulting in violation of individual privacy. This process thus accomplishes privacy preservation.

PPDM can be used in various real world applications such as health-care [Behlen and Johnson, 1999], counter-terrorism [Thuraisingham, 2003], homeland security [Fienberg, 2005], medicine [Berman, 2002], business collaboration [Oliveira and Zaane, 2007], bio-terrorism surveillance [Xiao-dan et al., 2005], etc. For example data mining techniques can be applied to identify the disease outbreak which may require data on disease incidence, patient background, etc. Such sensitive data cannot be used freely as it is protected by the legal authority. In some situations the target data can be divided into many organizational divisions and commercial concerns may restrict the data use. The data required to support initiatives like improving the quality of health-care system comes from several governmental and non-governmental organizations such as insurers, physicians, hospitals, pharmacies and labs. Boyens et al., [Boyens et al., 2004] proposed a methodology that chooses the optimal level of aggregation of the data taking into account simultaneously both factors the risk of disclosure as well as the utility of the released health-care data to legitimate users. Thuraisingham [Thuraisingham, 2004] provided an overview of the various types of terrorist threats and described how data mining techniques can provide solutions to counter terrorism and related problems.

The motivation behind employing PPDM methods to bankruptcy prediction is the sheer ignorance of the issue of privacy of the bank's crucial operational data, which if falls in wrong hands, can jeopardize bank's business interest and may trigger bank's collapse. Hence, there is a strong need to embrace the privacy preserving techniques before actually a data miner/analyst takes a look at the original data. However, the privacy preserving techniques entail inevitable loss of information that may lead to decreased prediction accuracies compared to when the original data is used for data mining purpose. Therefore, the data miner/analyst has to be wary of the trade-off between the level of privacy he/she would like to achieve and the decrement in prediction accuracy that would ensue after a data mining technique is employed on the modified data. PPDM methods contribute to practical applications in the real world problems involving Banks financial data and clinical and/or genetics profiles. If the privacy of such data sets is violated then it will have serious legal implications meaning that the customer or patient can sue the bank or hospital. Thus, privacy preservation of data becomes utmost important.

In a typical transformation-based-PPDM scenario, we start with an input data matrix and transform it by some means. Let M be the high dimensional database, which we have to transform in such a way that the privacy is preserved as well as the data is not perturbed to a large extent. Our aim is to transform M into $M^{|}$ so that the following constraints are satisfied in the transformed data.

- A transformation T when applied to M must preserve the privacy of individual records, so that the released database $M^{|}$ conceals the values of confidential features, such as salary, health records, credit rating and others.

- Although, the transformed database $M^{|}$ is dissimilar from M, the resulting classification accuracy should be as close as possible to that of the original database.

In this chapter, the PSO trained Auto-Associative Neural Network (PSOAANN) performs the job of the transformation T.

3.2 Literature survey

Several researchers have done a significant amount of work in PPDM from different perspectives e.g., statistics, database, data mining and information hiding. Clifton and Marks [Clifton and Marks, 1996] reported that adopting a common framework for discussing privacy preservation would enable next generation data mining technology to make substantial advances in reducing privacy concerns to a great extent. Verykios et al., [Verykios et al., 2004] presented a state-of-the-art review in the area of PPDM and classified the existing privacy preservation algorithms into five different categories: data distribution, data modification, data mining algorithm, data or rule hiding, and privacy preservation. Zhang et al., [Zhang et al., 2005] suggested a set of metrics for assessing PPDM performance. Bertino et al., [Bertino et al., 2005] proposed taxonomy for classifying existing PPDM algorithms. Granmo and Oleshchuk [Granmo and Oleshchuk, 2005] presented an overview of selected approaches in area of PPDM. Vaidya et al., [Vaidya et al., 2006] classified PPDM algorithms into 3 major classes based on the type of modifications as presented in Table 1. In section 2.1 a brief overview of these methods is presented. Ramu and Ravi [Ramu and Ravi, 2009] proposed a novel PPDM method by hybridizing random projection and random rotation and applied to solve benchmark classification problems and bankruptcy prediction in banks. Bansal et al., [Bansal et al., 2011] presented privacy preserving algorithm for the neural network learning when the datasets is arbitrarily partitioned into two parties. They claimed further that apart from the weights no other information is learnt by the other party.

3.2.1 Perturbative methods

Perturbative methods rely on transforming the original data distribution using some mathematical transformation. Data perturbation approaches can be classified into two major categories: the probability distortion approach and the value distortion approach. The probability distortion approach replaces the data with another sample from the same (or estimated) distribution or the distribution by itself and the value distortion approach perturbs data elements or features directly by either additive noise, multiplicative noise, or some other randomization procedures. Evfimievski [Evfimievski, 2002] suggested using additive noise to distort the original data probability distribution of the original data. According to him any confidential feature X can be modified by adding noise e, chosen from some known probability distribution. The perturbed feature is Y=X+e. When this method is used for multi-feature databases, each feature in the database is perturbed independently. He also suggested to use Bayes function for the reconstruction of the original data distribution. The privacy of this technique is based on how closely the original values of a modified feature can be estimated. The other perturbative methods which can be used for modifying the original data include multiplicative noise, Random projection [Liu et al., 2006a] based transformation and Rotation based transformation [Ketel and Homaifar, 2005]. These methods can be used for distributed data mining and for the centrally stored data mining. Liu et al., [Liu et al., 2006a] used Random projection approach in the case of distributed data mining where two or more number of clients involve in data mining task.

3.2.2 Non-perturbative methods

Non-perturbative methods do not rely on distortion of original data but on partial suppressions or reductions of detail. Crises [Crises, 2004] summarized some of the non-perturbative techniques. Some of the techniques in this category are as follows:

3.2.2.1 Data Swapping

Dalenius and Reiss [Dalenius and Reiss, 1982] first proposed data swapping with the idea of transforming the database by switching a subset of features between selected pairs of records so that the lower order frequency counts or marginals are preserved and data confidentiality is not compromised. This technique is equally used for classification under the data perturbation category. A variety of refinements and applications of data swapping have been proposed.

3.2.2.2 K-Anonymity

The k-Anonymity model [Sweeney, 2002] considers the problem that the data owner wants to share a collection of person-specific data without revealing the identity of an individual. To achieve this goal, data generalization and suppression techniques are used to protect the sensitive information. All features (termed as quasi-identifier) in the private database that could be used for linking with external information that would be determined, and the data is released only if the information of each person contained in the release cannot be distinguished from at least k-1 other people. Some of the other techniques like Data shuffling [Muralidhar and Sarathy, 2006], local suppression [Crises, 2004] include in this category.

3.2.2.3 Cryptographic Techniques

These techniques are generally used in distributed data mining. Pinkas [Pinkas, 2002] reviewed various cryptographic techniques for PPDM. He offered a broad view of Secure Multiparty Computation (SMC) framework and its applications to data mining. The SMC technique introduced by Yao [Yao, 1982] considers the problem of evaluating a function of the secret inputs from two or more parties, such that no party learns anything except the designated output of the function. Yao [Yao, 1986] used SMC to solve a problem of two millionaires who want to know who is richer without disclosing their net worth to each other. A large body of cryptographic protocols including circuit evaluation protocol, oblivious transfer, homomorphic encryption, and commutative encryption, serve as the building blocks of SMC. The disadvantages of the cryptographic techniques are the computational time and the overhead.

3.3 Proposed approach

3.3.1 Particle Swarm Optimization Trained Auto Associative Neural Network

Baldi and Hornik introduced the auto associative neural networks in 1989. They used the auto associative neural network in finding the principle components [Baldi and Hornik, 1989].

The proposed PSOAANN is a three layered neural network architecture consisting of input layer, hidden layer and output layer. The input and output layers consist not only equal number of nodes but also represent the same input features. The number of hidden nodes is a user defined parameter. Each input node in the input layer is connected to each node in the hidden layer and each hidden node is connected to the each of the node in the output layer. Sigmoid activation function is used in the hidden and output layers. The idea behind using the input features in the output layer is, at the end of training, the output nodes contain the perturbed values of the original input features. In other words, the PSOAANN is transforming the original set of input features nonlinearly into a set of perturbed values.

In the past, Hruschka and Natter [Hruschka and Natter, 1999] used a three layered architecture having input, hidden and output layers for market segmentation, where they treated the hidden nodes as clusters. They used sigmoid activation function in hidden and output layers and the backpropagation algorithm for training the network (weight updation). However, there are well-known drawbacks in back propagation algorithm such as entrapment in local minima and slow convergence. To overcome these drawbacks we propose to use PSO for weight updation, instead of the backpropagation.

The difference between the proposed approach and Hrushcka and Natter [Hruschka and Natter, 1999] approach are (i) Hruschka and Natter treated the hidden nodes as clusters, where as we didn't treat like that, (ii) They used backpropagation algorithm for training the network, where employed PSO for updation of weights. (iii) they used the network for market segmentation, however we used for privacy preservation purpose.

The architecture of PSOAANN is depicted in Figure 3.1. In this paper, the error function is taken as a normalized root mean squared error (NRMSE). The advantage of using a three layered AANN for privacy preservation over the traditional five layered AANN [Kramer, 1991]; [Pramodh and Ravi, 2007]; [Ravi and Pramodh, 2010] is the obvious decrease in computations and the associated complexity and thereby making it much simpler to understand and implement.

The training algorithm for PSOAANN is as follows:

1. Specify the required number of hidden nodes. Initialize randomly the weight values between the input and the hidden layers and also between the hidden and the output layers using uniform distribution in the range [-5 5]. The output nodes contain the input features as the target features thereby bringing in the auto association concept.

2. Compute $\hat{x}_k$ as follows: x_k is the actual input and $\hat{x}_k$ is the predicted input. Let *nin* and *nhn* be the number of input nodes and hidden nodes. The

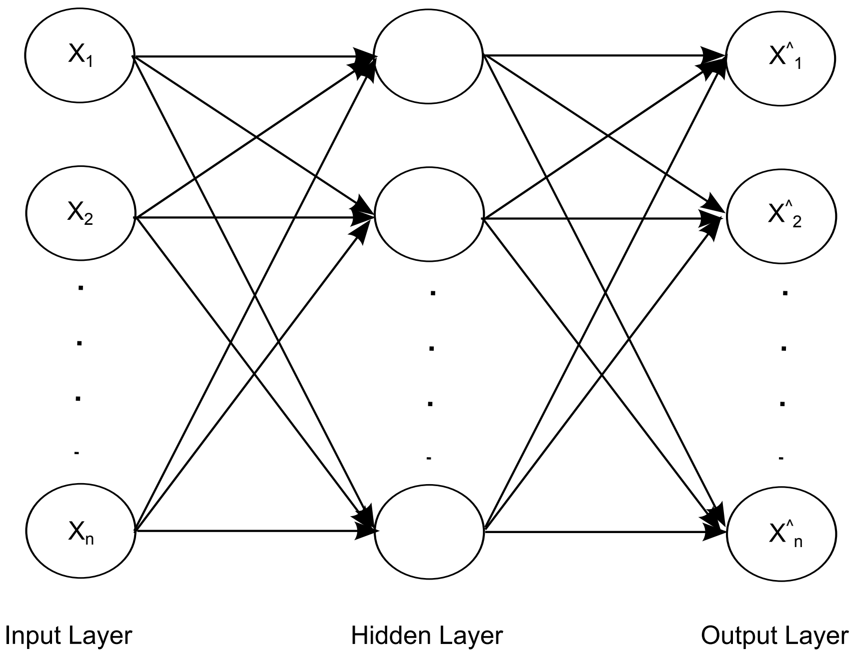

Figure 3.1: PSOAANN

predicted output is calculated as follows:

$$\hat{x}_k = \frac{1}{1 + \exp\left[-\sum_{j=1}^{nhn} w_j \times \frac{1}{1+\exp\left[-\sum_{i=1}^{nin} w_{ij}x_{ij}\right]}\right]} \tag{3.1}$$

Where k= 1 to nin; i, j, o represent the input nodes, hidden nodes and output nodes respectively; w_ij represents the weights between the input and hidden layer; w_jo represents the weights between the hidden and output layer; x_ij represents the input data.

3. Compute error measure (NRMSE) E as follows:

$$E = \sqrt{\frac{\sum_{k=1}^{nin}(x_k - \hat{x}_k)^2}{\sum_{k=1}^{nin}(x_k)^2}} \tag{3.2}$$

4. Test whether convergence is achieved by checking if $\mid E_{old} - E_{new} \mid < \epsilon$, ϵ is a predefined small value which is 10^{-6}.

5. Update all weight values by PSO algorithm.

6. Repeat step 2 to 5 until convergence is achieved or the max number of iterations is completed.

The PSOAANN is interesting mix of both unsupervised and supervised learning. The unsupervised learning flavor is present because we do not supply the output feature information to the network during training. This feature accomplishes the auto-association task. However, the supervised learning flavor is also conspicuously present because in this special feed-forward network, we need a training algorithm to update the weights between the 3 layers involved. Thus, the PSO based training algorithm accomplishes the auto-association that is required to preserve privacy. The end result is that the original input features undergo a nonlinear transformation yielding their perturbed version.

3.3.2 Proposed Feature Selection Method

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. In this chapter, we proposed a feature selection method for auto associative neural network.

It is 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, Ripper are used to generate rules for classification problems and CART 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.

Feature selection is a procedure to determine the important features of a given task. Guyon and Elisseeff [Guyon and Elisseeff, 2003] indicated that there are many potential benefits of feature selection: facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times, and defying the curse of dimensionality to improve prediction performance.

We proposed a new method of feature selection between input and hidden layers only. Since the input and output layer represent the same input features, we consider the input and hidden layers in this architecture for feature selection.

The proposed method of feature selection:

(1) For each hidden node j in the hidden layer, sum the absolute weights as $S_j = \sum_{i=1 j=1}^{nin nhn} \mid w_{ij} \mid$. w_{ij} represents the weight between the ith input and jth hidden node. nin and nhn represent number of input nodes and number of hidden nodes respectively.

(2) Each of the weight connection from all the input nodes to each of the hidden layer is converted into the contribution of each input nodes with respect to each of the hidden node as follows.

$$r_{ij} = \frac{\mid w_{ij} \mid}{S_j} * 100\% \tag{3.3}$$

This percentage serves the measure of importance of each of the input features at each of the hidden node in the hidden layer.

(3) At each of the hidden node, ranking is given to each of the input features. We took a maximum of top 3 or 4 rank features at each of the hidden node. Because each hidden nodes yields different ranks to features, we calculate the frequency of occurrence of the selected input features and considered 50% of total features in number.

The data flow diagram of the proposed feature selection approach is shown in Figure 3.2.

The full features and selected features are fed to the DT / Ripper and DENFIS / CART for the rule generation purpose of classification and regression problems respectively. DT / Ripper and DENFIS / CART rule generation methods which are discussed in Appendix B.5.4. Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data. Generated rules are

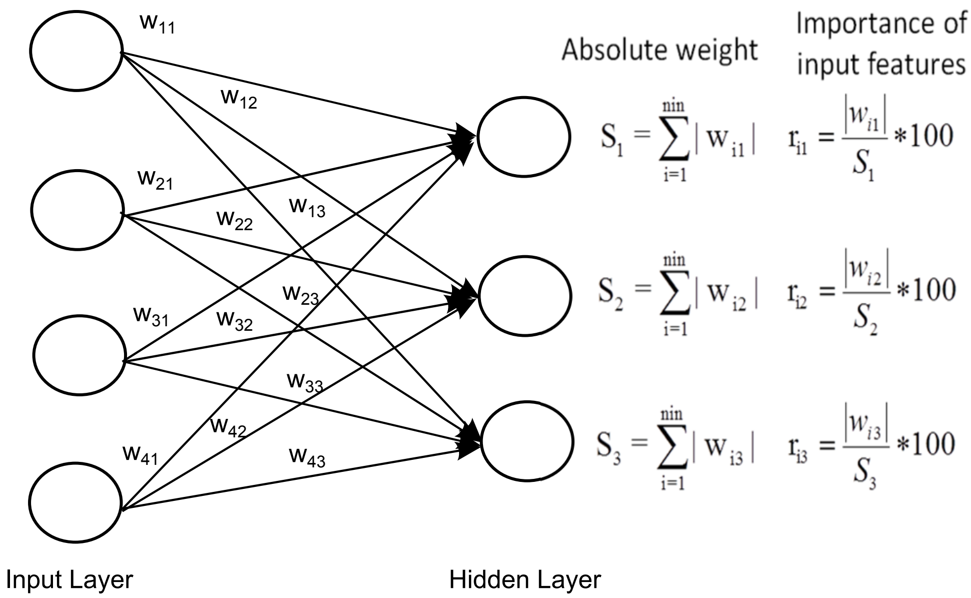

Figure 3.2: Proposed Feature Selection

then tested against the validation set. Prediction accuracy of the rules is determined in terms of accuracy for classification problems and RMSE for regression problems on validation set.

The data flow diagram of the proposed approach is show in Figure 3.3.

3.4 Datasets Analyzed

We chose publicly available benchmark datasets from UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`) such as IRIS, WINE and WBC datasets.The effectiveness of the proposed model is tested over classification datasets of bank bankruptcy datasets such as Spanish banks [Olmeda and Fernndez, 1997], Turkish banks [Canbas et al., 2005], UK banks [Beynon and Peel, 2001] and US banks [Rahimian et al., 1996].

In addition to the classification problems, we also analyzed regression problems viz., Auto MPG dataset [Asuncion and Newman, 2007], Boston Housing dataset [Asuncion and Newman, 2007], Forest Fires dataset [Asuncion and Newman, 2007] and Body Fat dataset [Penrose et al., 1985], Pollution dataset [McDonald and Schwing, 1973]. Information about the classification anad regression

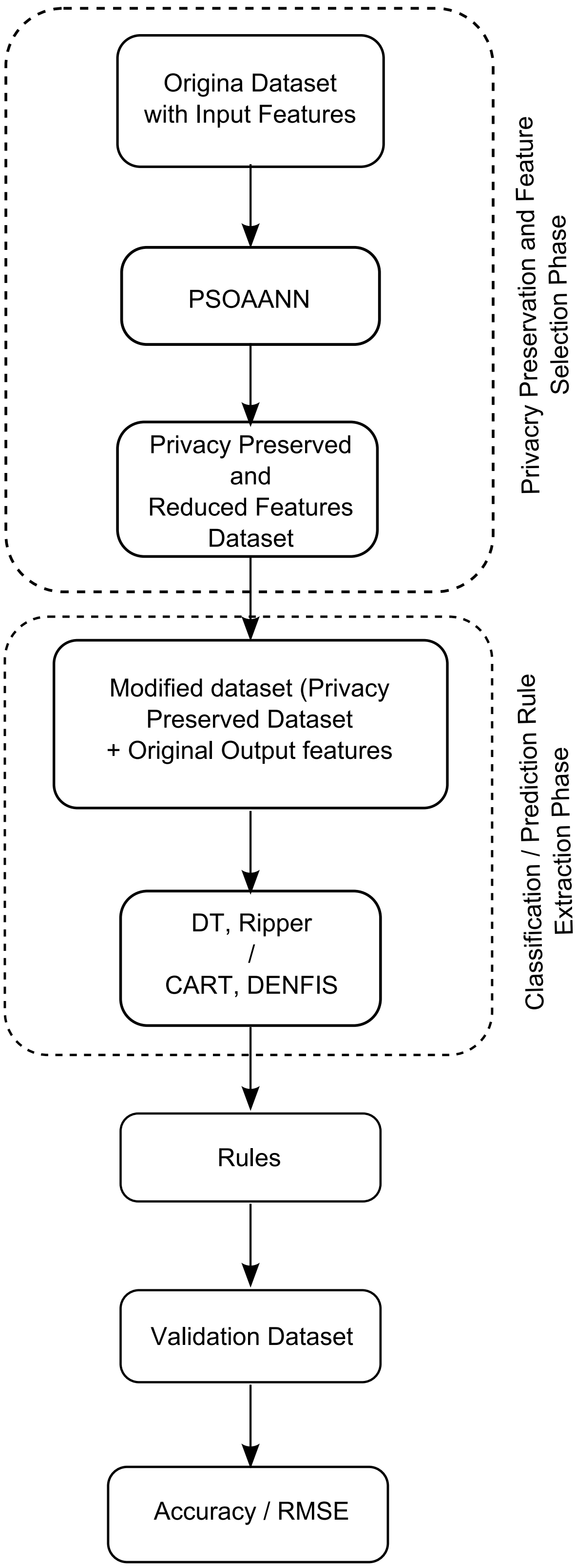

Figure 3.3: Proposed approach data flow diagram

datasets analyzed is presented in Appendix C, out of the listed datasets in Table 1.1 for demonstration purpose.

3.5 Results and Discussion

We developed the PSOAANN in C language in Windows 7 environment on a desktop with 2 GB RAM The parameters of the PSO algorithm are the number of particles and stochastic acceleration terms (c1 and c2). The number of particle is taken as either 30 or 40 in the experiments and c1 and c2 is chosen to be 2 in all the experiments. The number of hidden nodes is varied between the 3 and 7 in PSOAANN. We used KNIME 2.0.0 (`www.knime.org`) to run the DT (J48) and Ripper in the experiments. The reduced features of the classification and regression datasets are tabulated in Table 3.1 and Table 3.2 respectively.

The average accuracies obtained by DT in the case of transformed datasets and the original dataset are presented in Table 3.3 . Thus, we compared the results of datasets with and without privacy preservation. We performed t-test and the t-statistic values are presented in Table 3.7. In the case of WBC, New thyroid and UK datasets, at 1% level of significance, the results of with and without privacy preservation are found to be statistically insignificant. Further, we found that in other datasets, at 1% level of significance, they are statistically significant and also the case of without privacy preservation yielded better result. Further, in all other datasets except WINE, privacy preservation yielded worse result compared to the case of without privacy preservation.

The average accuracies obtained by DT in the case of transformed datasets of the reduced features are presented in Table 3.4. Thus, we compared the results of datasets with full features of the privacy preserved dataset. In the case of WBC and Spanish banks dataset, the full features and reduced features result is almost equal. Where in the case of Wine, Turkish banks and UK banks datasets reduced feature results did not outperformed the full features result.

The average accuracies obtained by Ripper are presented in Table 3.5. Here also performed t-test and t-statistic values are presented in Table 3.7. In the case of wine, WBC and UK dataset at 1% level of significance the results of with and without privacy preservation turned out to be statistically insignificant. Further,

Table 3.1: Important features selected in classification datasts

Dataset	**Features selected**
Wine	Malic acid
	Ash
	Magnesium
	Total phenols
	Proanthocyanins
	Proline
WBC	Clump Thickness
	Marginal Adhesion
	Bare Nuclei
	Bland Chromatin
	Normal Nucleoli
Spanish	CA/TA
	CAC/TA
	CA/L
	NI/TA
	NI/L
Turkish	IE/APA
	II+IE
	(SE+TI)/TA
	(SEB+RR)/P
	IE/TE
	LA/TA
UK	SALES
	FF/TL
	CL/TA
	LAG
	AGE

we also found that in other datasets, at 1% level of significance, they are statistically significant and also the case of without privacy preservation yielded better result. Further, we also computed the t-test between accuracies obtained by the DT and Ripper on the transformed data and t-statistic values are presented in Table 3.7. Here, we found that in all datasets except New Thyroid, DT and Ripper are statistically insignificant at 1% level of significance. From this we can infer that one can use either DT or Ripper for rule extraction purpose in the framework of the proposed privacy preserved architecture. The rules along with their coverage corresponding to the fold which yielded highest accuracy by DT and Ripper are presented in Appendix. Apart from accuracies, the rule base size obtained by

Table 3.2: Important features selected in regression datasts

Dataset	**Features selected**
AutoMPG	weight
	acceleration
	model year
	origin
Bodyfat	weight
	height
	abdomen
	hip
	thigh
	forearm
	wrist
BostonHousing	crim
	indus
	nox
	ptratio
	lstat
Forestfires	x-axis
	day
	ffmc
	dc
	rh
	rain
Pollution	jant
	ovr
	hous
	dens
	hc
	so
	humid

the DT and Ripper is almost the same in all datasets except WINE and UK.

The average accuracies obtained by Ripper in the case of transformed datasets of the reduced features are tabulated in Table 3.6. Thus, we compared the results of datasets with full features of the privacy preserved dataset. In the case of all the datasets except Wine dataset, the reduced features of the hybrid PSOAANN + Ripper of the classification problems achieved better results compared to the full features of the same hybrid.

Table 3.3: Average accuracies of PSOAANN + Decision tree

Dataset	Original Dataset			Privacy Preserved Dataset		
	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*
Iris	NA	NA	96.00	NA	NA	93.33
Wine	NA	NA	94.44	NA	NA	97.77
WBC	95.53	93.86	90.29	100	96.00	98.23
New Thyroid	NA	NA	94.18	NA	NA	94.88
Spanish	100	60.00	83.33	90.00	46.00	71.66
Turkish	100	92.50	95.71	43.00	83.00	67.17
US	90.77	97.50	94.00	87.00	74.00	80.00
UK	78.33	38.33	58.33	35.00	93.00	64.16

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

Table 3.4: Average accuracies of PSOAANN + Decision tree over reduced features

Dataset	Sens*	Spec*	Acc*
Wine	NA	NA	80.58
WBC	100	96.59	99.63
Spanish	88.57	46	70.93
Turkish	66.66	100	85.71
UK	36.66	90	63.33

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

Rules extracted using the proposed approach for classification problems of full features are presented below:

Rules generated by Decision Tree (C4.5):

IRIS DATASET

(1) If Petal width $\leq$ 0.505359 and Sepal length $\leq$ 0.443342 then IRIS VERSICOLOR

(2) If Petal width $\leq$ 0.505359 and Sepal length $>$ 0.443342 then IRIS VIRGINICA

(3) If Petal width$>$ 0.505359 then IRIS SETOSA

WBC DATASET

Table 3.5: Average accuracies of PSOAANN + Ripper

Dataset	Original Dataset			Privacy Preserved Dataset		
	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*
Iris	NA	NA	97.33	NA	NA	92.99
Wine	NA	NA	97.18	NA	NA	97.62
WBC	97.02	92.27	90.88	99.00	97.00	85.29
New Thyroid	NA	NA	96.51	NA	NA	90.46
Spanish	98.57	76.00	89.16	76.00	57.00	66.66
Turkish	100	97.50	98.57	57.00	100.00	81.42
US	87.69	97.50	92.40	85.00	75.00	80.00
UK	85.00	40.00	62.50	60.00	77.00	68.33

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

Table 3.6: Average accuracies of PSOAANN + Decision tree over reduced features

Dataset	Sens*	Spec*	Acc*
Wine	NA	NA	76.76
WBC	98.68	96.93	98.026
Spanish	84.28	54	71.66
Turkish	59.99	100	82.85
UK	65	71.66	68.33

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

(1) If clumpthickness $\leq$ 0.350595 then BENIGN

(2) If clumpthickness >0.350595 then MALIGNANT

NEW THYROID DATASET

(1) If SThyroxin $\leq$ 0.307997 and TSH $\leq$ 0.160963 then NORMAL

(2) If SThyroxin $\leq$ 0.307997 and TSH > 0.160963 then HypoThyroid

(3) If SThyroxin >0.307997 then HyperThyroid

WINE DATASET

(1) If Ash $\leq$ 0.538132 and Alcalinity of ash $\leq$ 0.455098 and Nonflavanoidphenols $\leq$ 0.402591 then CLASS B

Table 3.7: t-statistic values computed

Dataset	DT (Original vs Transformed)	Ripper (Original vs Transformed)	Dt vs Ripper (Transformed Dataset)
Iris	4.00	7.80	0.55
Wine	6.26	1.23	1.50
WBC	1.25	2.57	1.67
New Thyroid	0.66	6.28	5.09
Spanish	4.58	8.99	1.49
Turkish	5.03	6.57	2.71
US	12.67	11.19	1.00
UK	2.08	1.02	0.75

(2) If Ash $\leq$ 0.538132 and Alcalinity of ash $>$ 0.455098 and Ash $\leq$ 0.528514 and Alcalinity of ash $\leq$ 0.47219 and Hue $\leq$ 0.369992 then CLASS C

(3) If Ash $\leq$ 0.538132 and Alcalinity of ash>0.455098 and Ash $\leq$ 0.528514 Alcalinity of ash $\leq$ 0.47219 and Hue >0.369992 then CLASS C

(4) If Ash $\leq$ 0.538132 and Alcalinity of ash>0.455098 and Ash $\leq$ 0.528514 and Alcalinity of ash>0.47219 then CLASS C

(5) If Ash $\leq$ 0.538132 and Alcalinity of ash>0.455098 and Ash >0.528514 then CLASS B

(6) If Ash >0.538132 then CLASS A

SPANISH DATASET

(1) If (CAC/TA) $\leq$ 0.431644 then NonBankrupt

(2) If (CAC/TA) $>$ 0.431644 then Bankrupt

TURKISH DATASET

(1) If (SHE/TI)/(TA+CC) $\leq$ 0.973129 then Bankrupt

(2) If (SHE/TI)/(TA+CC) $>$ 0.973129 then NonBankrupt

US DATASET

(1) If (EIT/TA) $\leq$ 0.794781 then Bankrupt

(2) If (EIT/TA) $>$ 0.794781 then NonBankrupt

UK DATASET

(1) If (CA/CL) $\leq$ 0.204983 then NonBankrupt

(2) If (CA/CL) >0.204983 and (CA/CL) $\leq$ 0.207137 then Bankrupt

(3) If (CA/CL) >0.204983 and (CA/CL) >0.207137 and (FF/TL) $\leq$ 0.326856 then NonBankrupt

(4) If (CA/CL) >0.204983 and (CA/CL) >0.207137 and (FF/TL) $>$ 0.326856 then Bankrupt

Rules generated by Ripper

IRIS DATASET

(1) If Petal length $\leq$ 0.364484 then IRIS SETOSA

(2) If Petal length $\leq$ 0.422152 then IRIS VERSICOLOR

(3) If Petal length $>$ 0.422152 then IRIS VERGINICA

WBC DATASET

(1) If Clumpthickness $\geq$ 0.376957 then Malignant

(2) If Clumpthickness $\geq$ 0.351184 and Clumpthickness $\leq$ 0.368407 then Malignant

(3) If Clumpthickness $>$ 0.351184 and Clumpthickness $>$ 0.368407 then BENIGN

WINE DATASET

(1) If Alcalinity of ash $\geq$ 0.46734 and Proanthocyanins $\leq$ 0.328308 then Class C

(2) If Proanthocyanins $\leq$ 0.318861 then Class C

(3) If Ash $\geq$ 0.539347 and Hue $\geq$ 0.365639 then Class A

(4) If Proanthocyanins $\geq$ 359059 and Alcalinity of ash $\geq$ 0.446665 then Class A

(5) If Proanthocyanins $\geq$ 359059 and Alcalinity of ash $<$ 0.446665 then Class B

NEW THYROID DATASET

(1) If TD $\geq$ 0.175793 and Sthyroxin $\leq$ 0.296417 then HypoThyroid

(2) If SThyroxin $\geq$ 0.310244 then HyperThyroid

(3) If SThyroxin $<$ 0.310244 then NORMAL

SPANISH DATASET

(1) If CAC/TA $\leq$ 0.431934 then NonBankrupt

(2) If CAC/TA $>$ 0.431934 then Bankrupt

TURKISH DATASET

(1) If II/IE $\geq$ 0.415229 then Bankrupt

(2) If II/IE $<$ 0.415229 then NonBankrupt

US DATASET

(1) If EIT/TA $\geq$ 0.794884 then NonBankrupt

(2) If EIT/TA $<$ 0.794884 then Bankrupt

UK DATASET

(1) If CL/TA $\leq$ 0.515514 then NonBankrupt

(2) If CL/TA $>$ 0.515514 then Bankrupt

Rules extracted using the proposed approach for classification problems of reduced features are presented below:

Rules generated by Decision Tree (C4.5):

WBC DATASET

(1) If CLUMPtHICKNESS $\leq$ 0.350595 then Benign

(2) If CLUMPtHICKNESS $>$ 0.350595 then Malignant

WINE DATASET

(1) If Magnesium $\leq$ 0.46666 and Proanthocyanins $\leq$ 0.396937 then Class B

(2) If Magnesium ≤ 0.46666 and Proanthocyanins > 0.396937 and Ash ≤ 0.347187 then Class A

(3) If Magnesium ≤ 0.46666 and Proanthocyanins > 0.396937 and Ash > 0.347187 and Total phenols ≤ 0.299409 then Class B

(4) If Magnesium ≤ 0.46666 and Proanthocyanins > 0.396937 and Ash > 0.347187 and Total phenols > 0.299409 then Class A

(5) If Magnesium > 0.46666 then Class C

SPANISH DATASET

(1) If CAC/TA ≤ 0.431644 then NonBankrupt

(2) If CAC/TA > 0.431644 then Bankrupt

TURKISH DATASET

(1) If II+IE ≤ 0.415126 then NonBankrupt

(2) If II+IE > 0.415126 then Bankrupt

UK DATASET

(1) If FF/TL ≤ 0.326856 then NonBankrupt

(2) If FF/TL > 0.326856 then Bankrupt

Rules generated by Ripper

WBC DATASET

(1) If ClumpThickness ≥ 0.351184 then Malignant

(2) If ClumpThickness < 0.351184 then Benign

WINE DATASET

(1) If Magnesium ≥ 0.46734 then Class C

(2) If Magnesium ≥ 0.458254 and Total phenols ≤ 0.289867 then Class C

(3) If Proanthocyanins ≥ 0.397648 and Ash ≤ 0.347018 then Class A

(4) If Total phenols $\geq$ 0.313 and Malic acid $\geq$ 0.708156 then Class A

(5) If Total phenols $\geq$ 0.313 and Malic acid $<$ 0.708156 then Class B

SPANISH DATASET

(1) If CAC/TA $\leq$ 0.431934 then NonBankrupt

(2) If CAC/TA $>$ 0.431934 then Bankrupt

TURKISH DATASET

(1) If II+IE $\geq$ 0.415229 then Bankrupt

(2) If II+IE $<$ 0.415229 then NonBankrupt

UK DATASET

(1) If LAG $\geq$ 0.719857 then Bankrupt

(2) If LAG $<$ 0.719857 then NonBankrupt

Hence, we infer that our PSOAANN based hybrid model can be used for rule extraction in solving the financial classification problems while preserving privacy. Also, we noticed that the privacy preservation module did not worsen the accuracies drastically, which in itself is a significant outcome of the present study.

Further, we solved the regression problems using PSOAANN and rule extraction methods. We employed DENFIS and CART for extracting rules from PSOAANN. DENFIS is available in NeuCom tool, whose student version is freely available at `http://www.aut.ac.nz/`. CART is freely available at www.saflord-systems.com.

Average RMSE values, obtained over 10-folds for the hybrid PSOAANN + DENFIS, with and without feature selection, are tabulated in Table 3.8. From Table 3.8, it is observed that the reduced feature set yielded better RMSE values compared to that of the full feature set in three datasets namely Auto MPG, Boston Housing, and Forest Fires. Percentage relative difference between RMSE values yielded by full feature set and reduced feature set is also computed and presented in Table 3.8. Since the obtained average RMSE values for with and without feature selection are numerically close to each other, we performed t-test whether they are statistically significant or not. The t-test is performed on the hybrid of PSOAANN + DENFIS on full features and reduced features at 1% level of significance, presented in Table 3.8. The t-test indicates that on Auto MPG, Boston Housing, Forest Fires and Pollution dataset with full features and reduced features are statistically significant. However in the case of Body Fat dataset the RMSE values are insignificant.

Table 3.8: Average RMSE values over 10-folds of PSOAANN + DENFIS

Dataset	Full Features	Reduced Features	% Relative difference	t-test values
Auto MPG	0.11505	.09384	18.43546	4.79
Body Fat	0.04759	0.04789	0.630385	0.28
Boston Housing	0.41484	0.17154	58.64912	5.20
Forest Fires	0.02393	0.01735	27.49687	5.62
Pollution	0.02306	0.11185	385.039	30.16

Average RMSE values, obtained over 10-folds for the hybrid PSOAANN + CART, with and without feature selection are tabulated in Table 3.9. From Table 3.9, it is observed that the full feature set yielded better RMSE values compared to reduced feature set in all the datasets except in Forest fires dataset. Percentage relative difference between RMSE values yielded by full feature set and reduced feature set is also computed and presented in Table 3.9. Since the obtained average RMSE values for with and without feature selection are numerically close to each other, we performed t-test whether they are statistically significant or not. The t-test is performed on the hybrid of PSOAANN + CART on full features and reduced features at 1% level of significance, tabulated in Table 3.9. In the case of Auto MPG, Boston Housing, Forest Fires and Pollution both the full feature set and reduced feature set hybrid are statistically insignificant.

Average rule base size over 10-folds is computed and presented in Table 3.10, for full feature set and reduced feature set for all the datasets. In the case of PSOAANN + DENFIS hybrid, the rule base size of reduced features is higher

Table 3.9: Average RMSE values over 10-folds of PSOAANN + CART

Dataset	Full Features	Reduced Features	% Relative difference	t-test values
Auto MPG	0.1075	0.10944	1.804534	0.537005
Body Fat	0.10187	0.11815	15.98233	3.717352
Boston Housing	0.18098	0.18379	1.553729	0.384312
Forest Fires	0.01956	0.01409	27.95869	0.980145
Pollution	0.11549	0.12541	8.590848	0.774523

than that of rule base size of full feature set in all the datasets except forest fires dataset. The length of the rule has reduced with reduced feature set and achieved better RMSE value in the case of Auto MPG, Boston Housing and Forest Fires dataset. In the case of PSOAANN + CART hybrid the rule base size of Boston Housing, Forest Fires and Pollution dataset has reduced with reduced features set compared to full features set.

Table 3.10: Average rule base size of full and reduced features

Dataset	PSOAANN +DENFIS		PSOAANN + CART	
	Full Features	Reduced Features	Full Features	Reduced Features
Auto MPG	3.2	5.5	21	23
Body Fat	3	6.1	19.8	19.9
Boston Housing	2.4	3	8.1	4.3
Forest Fires	3.9	3.1	4.2	3.1
Pollution	3.8	3.9	4	3.8

The rules extracted by the proposed approach for regression problems of full features are presented below:

Rule extracted using DENFIS:

Auto MPG

(1) If Weight is GaussianMF(0.50 0.49) and Acceleration is GaussianMF(0.58 0.25) and Model Mile per Gallonear is GaussianMF(0.20 0.26) and Origin is

GaussianMF(0.60 0.26) then Mile per Gallon = 1.35- 0.83 × Weight- 0.33 × Acceleration+ 1.45 × Model Mile per Gallonear- 0.01 × Origin

(2) If Weight is GaussianMF(0.10 0.90) and Acceleration is GaussianMF(0.66 0.19) and Model Mile per Gallonear is GaussianMF(-3.29 0.41) and Origin is GaussianMF(0.12 0.21) then Mile per Gallon = 1.40-1.31 × Weight-0.58 × Acceleration+2.26 × Model Mile per Gallonear+0.11 × Origin

(3) If Weight is GaussianMF(0.10 0.73) and Acceleration is GaussianMF(0.42 0.48) and Model Mile per Gallonear is GaussianMF(0.41 0.94) and Origin is GaussianMF(0.16 0.66) then Mile per Gallon = 1.40 -1.42 × Weight+0.65 × Acceleration+2.65 × Model Mile per Gallonear-2.09 × Origin

(4) If Weight is GaussianMF(0.33 0.42) and Acceleration is GaussianMF(0.08 0.73) and Model Mile per Gallonear is GaussianMF(0.51 0.59) and Origin is GaussianMF(0.17 0.54) then Mile per Gallon = 1.90-1.74 × Weight+ 0.20 × Acceleration+2.49 × Model Mile per Gallonear- 1.98 × Origin

(5) If Weight is GaussianMF(0.43 0.85) and Acceleration is GaussianMF(0.14 0.72) and Model Mile per Gallonear is GaussianMF(1.53 0.65) and Origin is GaussianMF(0.38 0.34) then Mile per Gallon = 0.79-0.47 × Weight+0.60 × Acceleration+1.71 × Model Mile per Gallonear-0.79 × Origin

(6) If Weight is GaussianMF(0.14 0.06) and Acceleration is GaussianMF(0.46 0.81) and Model Mile per Gallonear is GaussianMF(0.46 0.53) and Origin is GaussianMF(0.15 0.93) then Mile per Gallon = 2.07-2.00 × Weight-0.11 × Acceleration+2.43 × Model Mile per Gallonear-1.76 × Origin

Body Fat

(1) If Weight is GaussianMF(0.45 0.61) and Height is GaussianMF(0.25 0.31) and Abdomen is GaussianMF(0.45 0.61) and Hip is GaussianMF(0.44 0.61) and Thigh is GaussianMF(0.44 0.61) andForearm is GaussianMF(0.39 0.38) and Wrist is GaussianMF(0.43 0.64) then Body fat = 7.21+35.46 × Weight-3.62 × Height-157.05 × Abdomen+165.24 × Hip-56.95 × Thigh-6.80 × Forearm+ 9.62 × Wrist

(2) If Weight is GaussianMF(0.42 0.25) and Height is GaussianMF(0.43 0.47) and Abdomen is GaussianMF(0.41 0.23) and Hip is GaussianMF(0.41 0.23) and Thigh is GaussianMF(0.42 0.26) and Forearm is GaussianMF(0.45 0.55) and Wrist is GaussianMF(0.43 0.34) then Body fat = 5.25-41.94 $\times$ Weight-1.32 $\times$ Height-17.82 $\times$ Abdomen+81.91 $\times$ Hip-47.99 $\times$ Thigh-5.32 $\times$ Forearm+23.19 $\times$ Wrist

(3) If Weight is GaussianMF(0.44 0.44) and Height is GaussianMF(0.36 0.03) and Abdomen is GaussianMF(0.43 0.46) and Hip is GaussianMF(0.43 0.46) and Thigh is GaussianMF(0.44 0.40) and Forearm is GaussianMF(0.24 0.75) and Wrist is GaussianMF(0.43 0.39) then Body fat = 5.83-19.30 $\times$ Weight-3.37 $\times$ Height+8.06 $\times$ Abdomen+29.27 $\times$ Hip-40.33 $\times$ Thigh-5.55 $\times$ Forearm+19.79 $\times$ Wrist

(4) If Weight is GaussianMF(0.52 0.07) and Height is GaussianMF(0.51 0.32) and Abdomen is GaussianMF(0.52 0.05) and Hip is GaussianMF(0.52 0.05) and Thigh is GaussianMF(0.52 0.06) and Forearm is GaussianMF(0.49 0.77) and Wrist is GaussianMF(0.52 0.12) then Body fat = 5.39-39.11 $\times$ Weight-0.35 $\times$ Height-46.36 $\times$ Abdomen+115.67 $\times$ Hip-55.58 $\times$ Thigh-5.74 $\times$ Forearm+22.36 $\times$ Wrist

(5) If Weight is GaussianMF(0.47 0.39) and Height is GaussianMF(0.18 0.90) and Abdomen is GaussianMF(0.44 0.31) and Hip is GaussianMF(0.45 0.32) and Thigh is GaussianMF(0.48 0.44) and Forearm is GaussianMF(0.53 0.08) and Wrist is GaussianMF(0.50 0.65) then Body fat = - 0.53-249.51 $\times$ Weight+2.05 $\times$ Height+420.11 $\times$ Abdomen-223.39 $\times$ Hip-8.78 $\times$ Thigh-0.22 $\times$ Forearm+62.81 $\times$ Wrist

(6) If Weight is GaussianMF(0.54 0.94) and Height is GaussianMF(0.51 0.47) and Abdomen is GaussianMF(0.55 0.94) and Hip is GaussianMF(0.55 0.94) and Thigh is GaussianMF(0.55 0.94) and Forearm is GaussianMF(0.55 0.06) and Wrist is GaussianMF(0.53 0.95) then Body fat = - 0.20-353.18 $\times$ Weight+7.35 $\times$ Height+397.33 $\times$ Abdomen-81.98 $\times$ Hip-57.90 $\times$ Thigh-2.77 $\times$ Forearm+94.30 $\times$ Wrist

Boston Housing

(1) If CRIM is GaussianMF(-0.26 0.34) and INDUS is GaussianMF(-0.09 0.62) and NOX is GaussianMF(0.09 0.38) and PTRATIO is GaussianMF(0.24 0.81) and LSTAT is GaussianMF(0.11 0.45) then MEDV = 1.20-4.92 $\times$ CRIM+0.74 $\times$ INDUS+3.24 $\times$ NOX-0.41 $\times$ PTRATIO+1.41 $\times$ LSTAT

(2) If CRIM is GaussianMF(0.35 0.06) and INDUS is GaussianMF(0.13 0.30) and NOX is GaussianMF(0.41 0.06) and PTRATIO is GaussianMF(0.19 0.33) and LSTAT is GaussianMF(0.34 0.07) then MEDV = 1.62-3.46 $\times$ CRIM+1.42 $\times$ INDUS+1.09 $\times$ NOX-1.63 $\times$ PTRATIO+2.15 $\times$ LSTAT

(3) If CRIM is GaussianMF(0.23 0.83) and INDUS is GaussianMF(0.19 0.81) and NOX is GaussianMF(0.24 0.85) and PTRATIO is GaussianMF(0.50 0.74) and LSTAT is GaussianMF(0.20 0.83) then MEDV = - 3.77-22.04 $\times$ CRIM-16.83 $\times$ INDUS+39.64 $\times$ NOX+ 19.26 $\times$ PTRATIO-13.43 $\times$ LSTAT

Forest fires

(1) If X-AXIS is GaussianMF(0.49 0.04) and Y-AXIS is GaussianMF(0.46 0.75) and MONTH is GaussianMF(0.45 0.89) and DMC is GaussianMF(0.49 0.30) and DC is GaussianMF(0.48 0.12) and RH is GaussianMF(0.45 0.14) then AREA = 2.27-0.03 $\times$ X-AXIS+ 2.41 $\times$ Y-AXIS-3.33 $\times$ MONTH-0.15 $\times$ DMC+0.41 $\times$ DC-0.54 $\times$ RH

(2) If X-AXIS is GaussianMF(0.49 0.69) and Y-AXIS is GaussianMF(0.39 0.85) and MONTH is GaussianMF(0.36 0.94) and DMC is GaussianMF(0.50 0.51) and DC is GaussianMF(0.49 0.45) and RH is GaussianMF(0.46 0.41) then AREA = 2.50-0.04 $\times$ X-AXIS+ 2.78 $\times$ Y-AXIS-3.91 $\times$ MONTH-0.04 $\times$ DMC+0.57 $\times$ DC-0.81 $\times$ RH

(3) If X-AXIS is GaussianMF(0.53 0.80) and Y-AXIS is GaussianMF(0.50 0.07) and MONTH is GaussianMF(0.51 0.20) and DMC is GaussianMF(0.51 0.46) and DC is GaussianMF(0.52 0.95) and RH is GaussianMF(0.50 0.90) then AREA = 1.05+0.00 $\times$ X-AXIS- 0.09 $\times$ Y-AXIS+0.07 $\times$ MONTH+0.06 $\times$ DMC+0.01 $\times$ DC-0.05 $\times$ RH

Pollution

(1) If JANT is GaussianMF(0.44 0.59) and OVER65 is GaussianMF(0.47 0.65) and HOUS is GaussianMF(0.45 0.68) and DENS is GaussianMF(0.27 0.58) and HC is GaussianMF(0.19 0.52) and SO is GaussianMF(0.27 0.58) and HUMID is GaussianMF(0.46 0.60) then MORT = 3.71+2.05 $\times$ JANT-8.27 $\times$ OVER65-2.18 $\times$ HOUS-3.40 $\times$ DENS-4.01 $\times$ HC +5.05 $\times$ SO +7.56 $\times$ HUMID

(2) If JANT is GaussianMF(0.45 0.04) 7 OVER65 is GaussianMF(0.31 0.13) and HOUS is GaussianMF(0.37 0.04) and DENS is GaussianMF(0.46 0.80) and HC is GaussianMF(0.61 0.17) and SO is GaussianMF(0.59 0.26) and HUMID is GaussianMF(0.38 0.16) then MORT = 14.68+36.47 $\times$ JANT+76.41 $\times$ OVER65-17.94 $\times$ HOUS-2.71 $\times$ DENS+21.04 $\times$ HC-35.19 $\times$ SO-102.83 $\times$ HUMID

(3) If JANT is GaussianMF(0.43 0.24) and OVER65 is GaussianMF(0.49 0.58) and HOUS is GaussianMF(0.44 0.67) and DENS is GaussianMF(0.24 0.06) and HC is GaussianMF(0.49 0.06) and SO is GaussianMF(0.32 0.05) and HUMID is GaussianMF(0.51 0.52) then MORT = - 1.17+3.08 $\times$ JANT+70.19 $\times$ OVER65+1.63 $\times$ HOUS+14.35 $\times$ DENS+34.44 $\times$ HC-45.35 $\times$ SO-77.50 $\times$ HUMID

PSOAANN + CART

Auto MPG:

(1) If (MODEL YEAR $\leq$ 0.363079 & ORIGIN $\leq$ 0.284024) then mean= 0.146703

(2) If (MODEL YEAR $\leq$ 0.363079 & ORIGIN $>$ 0.284024) then mean = 0.3524

(3) If (MODEL YEAR $>$ 0.363079 & MODEL YEAR $\leq$ 0.387028 & ORIGIN $\leq$ 0.282366) then mean = 0.276508

(4) If (MODEL YEAR $>$ 0.363079 & MODEL YEAR $\leq$ 0.387028 & ORIGIN $>$ 0.282366) then mean= 0.424724

(5) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ORIGIN $\leq$ 0.262815) then mean= 0.7713

(6) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ORIGIN $>$ 0.262815 & ACCELERATION $\leq$ 0.3677 & WEIGHT $\leq$ 0.580048) then mean= 0.240025

(7) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ORIGIN $>$ 0.262815 & ACCELERATION $\leq$ 0.3677 & WEIGHT $>$ 0.580048) then mean= 0.39839

(8) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ACCELERATION $>$ 0.3677 & ORIGIN $>$ 0.262815 & ORIGIN $\leq$ 0.28284) then mean = 0.441009

(9) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ACCELERATION $>$ 0.3677 & ORIGIN $>$ 0.28284 & ORIGIN $\leq$ 0.296171 & WEIGHT $\leq$ 0.573617) then mean = 0.475286

(10) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ACCELERATION $>$ 0.3677 & ORIGIN $>$ 0.28284 & ORIGIN $\leq$ 0.296171 & WEIGHT $>$ 0.573617) then mean= 0.6531

(11) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ACCELERATION $>$ 0.3677 & ORIGIN $>$ 0.296171 & ORIGIN $\leq$ 0.304303) then mean= 0.287683

(12) If (MODEL YEAR $>$ 0.387028 & MODEL YEAR $\leq$ 0.409424 & ACCELERATION $>$ 0.3677 & ORIGIN $>$ 0.304303) then mean = 0.45063

(13) If (MODEL YEAR $>$ 0.409424 & WEIGHT $\leq$ 0.567021) then mean = 0.605614

(14) If (MODEL YEAR $>$ 0.409424 & WEIGHT $>$ 0.567021 & WEIGHT $\leq$ 0.567886) then mean= 0.395367

(15) If (MODEL YEAR $>$ 0.409424 & WEIGHT $>$ 0.567886 & ACCELERATION $\leq$ 0.385558) then mean = 0.643297

(16) If (MODEL YEAR $>$ 0.409424 & WEIGHT $>$ 0.567886 & ACCELERATION $>$ 0.385558) then mean = 0.781033

Body Fat

(1) If (HIP ≤ 0.369299 & HEIGHT ≤ 0.616522 & ABDOMEN ≤ 0.335662) then mean $=0.270877$

(2) If (HIP ≤ 0.369299 & ABDOMEN > 0.335662 & HEIGHT ≤ 0.603009) then mean $= 0.541377$

(3) If (HIP ≤ 0.369299 & ABDOMEN > 0.335662 & HEIGHT > 0.603009 & HEIGHT ≤ 0.615732) then mean $= 0.430292$

(4) If (HIP ≤ 0.369299 & ABDOMEN > 0.335662 & HEIGHT > 0.615732 & HEIGHT ≤ 0.616522) then mean $= 0.661053$

(5) If (HIP ≤ 0.369299 & HEIGHT > 0.616522 & ABDOMEN ≤ 0.327975) then mean$= 0.13979$

(6) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & HEIGHT > 0.616522 & HEIGHT ≤ 0.640913 & FOREARM ≤ 0.567538) then mean$= 0.395684$

(7) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & HEIGHT > 0.616522 & HEIGHT $\leq .640913$ & FOREARM > 0.567538 & THIGH ≤ 0.389392) then mean $= 0.532632$

(8) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & HEIGHT > 0.616522 & HEIGHT ≤ 0.640913 & FOREARM > 0.567538 & THIGH > 0.389392) then mean $= 0.280982$

(9) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & HEIGHT > 0.640913 & FOREARM ≤ 0.49268) then mean$= 0.56$

(10) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & FOREARM > 0.49268 & FOREARM ≤ 0.552787 & HEIGHT > 0.640913 & HEIGHT ≤ 0.670625) then mean$= 0.353918$

(11) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & FOREARM > 0.49268 & FOREARM ≤ 0.552787 & HEIGHT > 0.670625) thenmean $= 0.203684$

(12) If (HIP ≤ 0.369299 & ABDOMEN > 0.327975 & HEIGHT > 0.640913 & FOREARM > 0.552787) then mean$= 0.226566$

(13) If (HIP > 0.369299 & HEIGHT ≤ 0.595437) then mean $= 0.753383$

(14) If (HIP $>$ 0.369299 & WEIGHT $\leq$ 0.416667 & HEIGHT $>$ 0.595437 & HEIGHT $\leq$ 0.632087 & FOREARM $\leq$ 0.540211) then mean= 0.638196

(15) If (HIP $>$ 0.369299 & WEIGHT $\leq$ 0.416667 & HEIGHT $>$ 0.595437 & HEIGHT $\leq$ 0.632087 & FOREARM $>$ 0.540211) then mean = 0.511292

(16) If (HIP $>$ 0.369299 & WEIGHT $\leq$ 0.416667 & HEIGHT $>$ 0.632087) then mean =0.428722

(17) If (HIP $>$ 0.369299 & HEIGHT $>$ 0.595437 & WEIGHT $>$ 0.416667) then mean= 0.71579

Boston Housing

(1) If (CRIM $\leq$ 0.29449) then mean = 0.55045

(2) If (CRIM $>$ 0.29449 & CRIM $\leq$ 0.31266) then mean= 0.426722

(3) If (CRIM $>$ 0.31266 & CRIM $\leq$ 0.364969) then mean = 0.324014

(4) If (CRIM $>$ 0.364969) thenmean = 0.183918

Forest Fires

(1) If (DC $\leq$ 0.339062) then mean= 0.00974743

(2) If (DC $>$ 0.339062 & DC $\leq$ 0.339085) then mean = 0.684146

(3) If (DC $>$ 0.339085) then mean = 0.00859632

Pollution

(1) If (SO $\leq$ 0.289196) then mean = 0.619402

(2) If (SO $>$ 0.289196) then mean = 0.391931

The rules extracted by the proposed approach for regression problems of reduced features are presented below:

Rule extracted using DENFIS:

Auto MPG

(1) If Weight is GaussianMF(0.29 0.71) and Acceleration is GaussianMF(0.43 0.53) and Model year is GaussianMF(0.09 0.89) and Origin is GaussianMF(0.09 0.55) then Auto MPG = 0.80 - 0.27 * Weight + 0.70 * Acceleration + 1.38 * Model year - 0.76 * Origin

(2) If Weight is GaussianMF(0.25 0.29) and Acceleration is GaussianMF(0.65 0.34) and Model year is GaussianMF(0.13 0.25) and Origin is GaussianMF(0.53 0.32) then Auto MPG = 1.20 - 0.43 * Weight + 0.06 * Acceleration + 0.92 * Model year - 0.11 * Origin

(3) If Weight is GaussianMF(0.13 0.94) and Acceleration is GaussianMF(0.61 0.08) and Model year is GaussianMF(0.42 0.46) and Origin is GaussianMF(0.17 0.06) then Auto MPG = 1.25 - 0.78 * Weight - 0.08 * Acceleration + 1.70 * Model year - 0.34 * Origin

(4) If Weight is GaussianMF(0.50 0.19) and Acceleration is GaussianMF(0.34 0.90) and Model year is GaussianMF(0.17 0.51) and Origin is GaussianMF(0.39 0.76) then Auto MPG = 1.58 - 1.32 * Weight + 0.33 * Acceleration + 2.07 * Model year - 1.51 * Origin

Body Fat

(1) If Weight is GaussianMF(0.44 0.44) and Height is GaussianMF(0.39 0.52) and Abdomen is GaussianMF(0.43 0.41) and Hip is GaussianMF(0.43 0.41) and Thigh is GaussianMF(0.45 0.44) and Forearm is GaussianMF(0.46 0.50) and Wrist is GaussianMF(0.46 0.55) then Body fat = 5.39 - 18.10 * Weight - 3.18 * Height + 2.47 * Abdomen + 32.27 * Hip - 37.08 * Thigh - 4.99 * Forearm + 18.28 * Wrist

(2) If Weight is GaussianMF(0.49 0.21) and Height is GaussianMF(0.52 0.32) and Abdomen is GaussianMF(0.49 0.20) and Hip is GaussianMF(0.49 0.20) and Thigh is GaussianMF(0.49 0.21) and Forearm is GaussianMF(0.48 0.65) and Wrist is GaussianMF(0.50 0.26) then Body fat = 5.22 - 38.72 * Weight - 1.23 * Height - 29.43 * Abdomen + 91.96 * Hip - 48.59 * Thigh - 5.34 * Forearm + 22.26 * Wrist

(3) If Weight is GaussianMF(0.48 0.73) and Height is GaussianMF(0.36 0.36) and Abdomen is GaussianMF(0.48 0.73) and Hip is GaussianMF(0.48 0.73) and

Thigh is GaussianMF(0.48 0.71) and Forearm is GaussianMF(0.46 0.39) and Wrist is GaussianMF(0.48 0.75) then Body fat = 9.59 + 106.49 * Weight - 5.09 * Height - 316.47 * Abdomen + 282.30 * Hip - 75.34 * Thigh - 8.96 * Forearm - 2.30 * Wrist

(4) If Weight is GaussianMF(0.43 0.52) and Height is GaussianMF(0.46 0.04) and Abdomen is GaussianMF(0.42 0.54) and Hip is GaussianMF(0.42 0.54) and Thigh is GaussianMF(0.44 0.48) and Forearm is GaussianMF(0.19 0.73) and Wrist is GaussianMF(0.44 0.47) then Body fat = 7.13 + 19.59 * Weight - 5.54 * Height - 6.56 * Abdomen + 10.43 * Hip - 40.03 * Thigh - 6.67 * Forearm + 13.80 * Wrist

(5) If Weight is GaussianMF(0.51 0.40) and Height is GaussianMF(0.51 0.88) and Abdomen is GaussianMF(0.52 0.32) and Hip is GaussianMF(0.52 0.33) and Thigh is GaussianMF(0.51 0.45) and Forearm is GaussianMF(0.52 0.08) and Wrist is GaussianMF(0.50 0.65) then Body fat = 1.42 - 183.77 * Weight + 1.88 * Height + 248.66 * Abdomen - 88.66 * Hip - 26.30 * Thigh - 2.08 * Forearm + 49.62 * Wrist

(6) If Weight is GaussianMF(0.55 0.94) and Height is GaussianMF(0.50 0.46) and Abdomen is GaussianMF(0.55 0.94) and Hip is GaussianMF(0.55 0.94) and Thigh is GaussianMF(0.55 0.94) and Forearm is GaussianMF(0.58 0.06) and Wrist is GaussianMF(0.53 0.95) then Body fat = 0.46 - 389.04 * Weight + 10.48 * Height + 279.54 * Abdomen + 92.79 * Hip - 93.30 * Thigh - 4.08 * Forearm + 106.44 * Wrist

Boston Housing

(1) If CRIM is GaussianMF(0.13 0.60) and INDUS is GaussianMF(0.61 0.53) and NOX is GaussianMF(0.13 0.63) and PTRATIO is GaussianMF(0.60 0.62) and LSTAT is GaussianMF(0.13 0.67) then MEDV = 0.98 - 3.48 * CRIM - 2.20 * INDUS + 5.88 * NOX + 2.40 * PTRATIO - 2.29 * LSTAT

(2) If CRIM is GaussianMF(0.40 0.06) and INDUS is GaussianMF(0.10 0.27) and NOX is GaussianMF(0.42 0.06) and PTRATIO is GaussianMF(0.20 0.31) and LSTAT is GaussianMF(0.32 0.07) then MEDV = 1.58 - 3.22 * CRIM + 1.89 * INDUS + 2.01 * NOX - 1.91 * PTRATIO + 1.18 * LSTAT

(3) If CRIM is GaussianMF(0.17 0.21) and INDUS is GaussianMF(0.25 0.76) and NOX is GaussianMF(0.62 0.29) and PTRATIO is GaussianMF(0.29 0.98) and LSTAT is GaussianMF(0.75 0.37) then MEDV = 1.02 - 5.51 * CRIM + 0.38 * INDUS + 3.16 * NOX + 0.02 * PTRATIO + 2.08 * LSTAT

Forest Fires

(1) If X-AXIS is GaussianMF(0.50 0.05) and DAAREA is GaussianMF(0.50 0.76) and FFMC is GaussianMF(0.51 0.87) and DC is GaussianMF(0.49 0.32) and RH is GaussianMF(0.50 0.12) and RAIN is GaussianMF(0.50 0.14) then AREA = 0.70 + 0.04 * X-AXIS - 0.60 * DAAREA + 0.92 * FFMC - 0.06 * DC - 0.28 * RH + 0.34 * RAIN

(2) If X-AXIS is GaussianMF(0.50 0.62) and DAAREA is GaussianMF(0.48 0.86) and FFMC is GaussianMF(0.48 0.92) and DC is GaussianMF(0.50 0.55) and RH is GaussianMF(0.50 0.45) and RAIN is GaussianMF(0.49 0.41) then AREA = 1.37 + 0.02 * X-AXIS + 0.51 * DAAREA - 0.80 * FFMC + 0.04 * DC + 0.06 * RH - 0.20 * RAIN

(3) If X-AXIS is GaussianMF(0.50 0.73) and DAAREA is GaussianMF(0.50 0.08) and FFMC is GaussianMF(0.50 0.20) and DC is GaussianMF(0.50 0.49) and RH is GaussianMF(0.50 0.95) and RAIN is GaussianMF(0.51 0.90) then AREA = 1.06 - 0.01 * X-AXIS - 0.01 * DAAREA - 0.01 * FFMC + 0.03 * DC + 0.02 * RH - 0.06 * RAIN

Pollution

(1) If JANT is GaussianMF(0.44 0.59) and OVER65 is GaussianMF(0.47 0.65) and HOUS is GaussianMF(0.45 0.68) and DENS is GaussianMF(0.27 0.58) and HC is GaussianMF(0.19 0.52) and SO is GaussianMF(0.27 0.58) and HUMID is GaussianMF(0.46 0.60) then MORT = 3.71 + 2.05 * JANT - 8.27 * OVER65 - 2.18 * HOUS - 3.40 * DENS - 4.01 * HC + 5.05 * SO + 7.56 * HUMID

(2) If JANT is GaussianMF(0.45 0.04) and OVER65 is GaussianMF(0.31 0.13) and HOUS is GaussianMF(0.37 0.04) and DENS is GaussianMF(0.46 0.80) and HC is GaussianMF(0.61 0.17) and SO is GaussianMF(0.59 0.26) and

HUMID is GaussianMF(0.38 0.16) then MORT = 14.68 + 36.47 * JANT + 76.41 * OVER65 - 17.94 * HOUS - 2.71 * DENS + 21.04 * HC - 35.19 * SO - 102.83 * HUMID

(3) If JANT is GaussianMF(0.43 0.24) and OVER65 is GaussianMF(0.49 0.58) and HOUS is GaussianMF(0.44 0.67) and DENS is GaussianMF(0.24 0.06) and HC is GaussianMF(0.49 0.06) and SO is GaussianMF(0.32 0.05) and HUMID is GaussianMF(0.51 0.52) then MORT = - 1.17 + 3.08 * JANT + 70.19 * OVER65 + 1.63 * HOUS + 14.35 * DENS + 34.44 * HC - 45.35 * SO - 77.50 * HUMID

PSOAANN + CART

Auto MPG:

(1) If (ORIGIN $\leq$ 0.284333 and MODEL YEAR $\leq$ 0.363637)mean = 0.150608

(2) If (ORIGIN $\leq$ 0.284333 and MODEL YEAR $>$ 0.363637 and MODEL YEAR $\leq$ 0.387028) mean = 0.279521

(3) If (MODEL YEAR $\leq$ 0.387028 and ORIGIN $>$ 0.284333) mean = 0.417045

(4) If (MODEL YEAR $>$ 0.387028 and MODEL YEAR $\leq$ 0.409424 and ORIGIN $\leq$ 0.295125) mean = 0.492516

(5) If (MODEL YEAR $>$ 0.387028 and MODEL YEAR $\leq$ 0.409424 and ORIGIN $>$ 0.295125 and ORIGIN $\leq$ 0.304303) mean = 0.2663

(6) If (MODEL YEAR $>$ 0.387028 and MODEL YEAR $\leq$ 0.409424 and ORIGIN $>$ 0.304303) mean = 0.45787

(7) If (MODEL YEAR $>$ 0.409424 and WEIGHT $\leq$ 0.567021) mean = 0.596762

(8) If (MODEL YEAR $>$ 0.409424 and WEIGHT $>$ 0.567021 and WEIGHT $\leq$ 0.567886) mean = 0.395367

(9) If (MODEL YEAR $>$ 0.409424 and WEIGHT $>$ 0.567886) mean = 0.657452

Body Fat

(1) If (HEIGHT $\leq$ 0.616522 and ABDOMEN $\leq$ 0.335662) mean = 0.26924

(2) If (HEIGHT $\leq$ 0.616522 and ABDOMEN $>$ 0.335662 and ABDOMEN $\leq$ 0.361097) mean = 0.462517

(3) If (HEIGHT $>$ 0.616522 and ABDOMEN $\leq$ 0.32746) mean = 0.127519

(4) If (ABDOMEN $>$ 0.32746 and ABDOMEN $\leq$ 0.361097 and HEIGHT $>$ 0.616522 and HEIGHT $\leq$ 0.635476) mean = 0.353294

(5) If (ABDOMEN $>$ 0.32746 and ABDOMEN $\leq$ 0.361097 and HEIGHT $>$ 0.635476) mean = 0.238338

(6) If (ABDOMEN $>$ 0.361097 and HEIGHT $\leq$ 0.595437) mean = 0.737105

(7) If (ABDOMEN $>$ 0.361097 and HEIGHT $>$ 0.595437 and HEIGHT $\leq$ 0.627203) mean = 0.544682

(8) If (ABDOMEN $>$ 0.361097 and HEIGHT $>$ 0.627203 and WEIGHT $\leq$ 0.418947) mean = 0.418526

(9) If (ABDOMEN $>$ 0.361097 and HEIGHT $>$ 0.627203 and WEIGHT $>$ 0.418947) mean = 0.733684

Boston Housing

(1) If (CRIM $\leq$ 0.29449) mean = 0.552222

(2) If (CRIM $>$ 0.29449 and CRIM $\leq$ 0.31266) mean = 0.425355

(3) If (CRIM $>$ 0.31266 and CRIM $\leq$ 0.364969) mean = 0.323907

(4) If (CRIM $>$ 0.364969) mean = 0.186583

Forest Fires

(1) If (DAY $\leq$ 0.616104) mean = 0.00836286

(2) If (DAY $>$ 0.616104 and DAY $\leq$ 0.616159) mean = 1.00002

(3) If (DAY $>$ 0.616159) mean = 0.0205862

Pollution

(1) If (SO $\leq$ 0.275384) mean = 0.366617

(2) If (SO $>$ 0.275384 and SO $\leq$ 0.283256) mean = 0.702171

(3) If (SO $>$ 0.283256 and JANT $\leq$ 0.420406) mean = 0.441483

(4) If (SO $>$ 0.283256 and JANT $>$ 0.420406) mean = 0

3.6 Conclusions

In this chapter, we propose a rule extraction method from a privacy preserving Neural Network trained by Particle Swarm Optimization by using DT and Ripper for classification problems. The efficiency of the proposed hybrid for the classification problems was tested on four benchmark datasets namely Iris, WBC, Wine and New Thyroid, and four bankruptcy prediction problems viz. Spanish banks, Turkish banks, US banks and UK banks. Results indicate that the proposed hybrid has yielded comparable accuracies to those of the original dataset. Also, we presented a hybrid method to extract rules using DENFIS and CART from PSOAANN to solve regression problems. We tested the efficiency of the proposed hybrid over the regression problems on five benchmark regression datasets viz. Auto MPG, Body Fat, Boston Housing, Forest Fires and Pollution. We proposed a new method of feature selection using PSOAANN. In the current study, PSOAANN accomplish in preserving the privacy of the input features and as well as feature selection. The rule base size obtained with and without feature selection overall the datasets is also compared. Since the obtained RMSE values over with and without feature selection are close to each other, we perform t-test at 1% level of significance to find whether they are statistically significant or not. In the case of PSOAANN + CART, it is observed that the results are statistically insignificant between with and without feature selection overall datasets except Body Fat. In the case of PSOAANN + DENFIS, it is observed that the results are statistically significant between with and without feature selection for four datasets. From the results, we also conclude that the proposed hybrid is viable in solving the regression problems with and without the proposed feature selection method.

Part II - Pedagogical Approach

Chapter 4

Rule Extraction from Differential Evolution trained Radial Basis Function Neural Networks in solving Classification problems

This chapter presents a *pedagogical* rule extraction approach which uses differential evolution trained radial basis function neural network as a *black box*. With the introduction and literature review in improving radial basis function network in the first section, the proposed hybrid approach is presented in detail in next section. In third section datasets analyzed is presented. Following section provides Results and discussions. Final section concludes the chapter.

4.1 Introduction

Artificial neural network (ANN) found applications in classification, function approximation, optimization and control. There are two most popular architectures of ANN such as multilayer perceptron (MLP) and radial basis function (RBF) network, which are universal function approximators. These are being used for a wide range of application and because they can approximate any function under mild conditions. MLP is trained by supervised learning. On the other hand, the training of RBF involves both unsupervised and supervised phases. Unsupervised part is less approximate, which is relatively estimation fast. Supervised part of learning consists of solving a linear problem, which is fast and also avoids the problem of local minima which is usually encountered in training of MLP. Hence the training of RBF is faster than that of MLP. RBF has two sets of parameters viz. centers and widths of clusters and weights connecting hidden layer (also called kernel layer) and output layer [Benoudjit and Verleysen, 2003].

An RBF network has two layers. For example consider a function $f(X) : R^d \rightarrow R$. In a regression context, RBF approximates f(X) by a weighted sum of d-dimensional radial activation functions (sometimes including linear and independent terms). The RBFs are centered on well-positioned data points called centroids. The centroids can be regarded as the nodes of the hidden layer. The centroids are obtained during the unsupervised learning phase. The network weights between the hidden layer and the output layer are estimated using least square technique.

Suppose we want to approximate the function $f(X)$ with a set of M RBFs $\phi_j(X)$, centered on the centroids C_j and is defined by [Benoudjit and Verleysen, 2003]:

$$\phi_j(X) : R^d \rightarrow R \tag{4.1}$$

$$\text{where}\phi_j = \phi_j(\| X - C_j \|) \tag{4.2}$$

where $\| \, . \, \|$ denotes the Euclidean distance, $C_j \in R^d$ and $1 \leq j \leq M$, M is the number of clusters.

The approximation of the function $f(X)$ may be expressed as a linear combination of the RBFs:

$$\sum_{j=1}^{M} \lambda_j \phi_j(\| X - C_j \|) + \sum_{i=1}^{d} a_i x_i + b \tag{4.3}$$

where λ_j is weights and a_i, b are the coefficients of the independent variables and constant term respectively.

A typical choice for the RBFs is a set of multi-dimensional Gaussian kernels:

$$\phi_j(\| X - C_j \|) = \exp[-\frac{1}{2}(X - C_j)^T (X - C_j)] \tag{4.4}$$

Moody and Darken [Moody and Darken, 1989], proposed the K-means clustering algorithm as a unsupervised part of the algorithm, to find the location of the centroid C_j. Once the basis function parameters are determined, the transformation between the input data and the corresponding outputs of the hidden units is fixed. Then, the supervised part of the algorithm commences where the weights connecting the nodes in the hidden layer and the nodes in the output layer are estimated using the linear least squares technique. Accordingly, the minimization of the average mean square error yields the least square solution for the weights as follows:

$$\lambda = \phi^{+}y = (\phi^T\phi)^{-1}\phi^T y \tag{4.5}$$

where λ and y are the row vectors of weights, λ_j and training data outputs, y_p, respectively, ϕ is the nM matrix elements $\phi_{ij} = \exp(-\frac{||X_i - C_j||^2}{2\sigma^2})$ values and $\phi^{+} = (\phi^T\phi)^{-1}\phi^T$ denotes the pseudo-inverse of ϕ and n is the number of samples.

The primary objective of this chapter is to develop a new architecture for the RBF using differential evolution (DE) (DERBF) and extract from DERBF using GATree.

4.1.1 Literature review in improving RBF

Since training of RBF involves both unsupervised and supervised phases, researchers have proposed several alternative methodologies in both phases. We now review earlier works where online training algorithms for RBF were suggested. Fung et al., [Fung et al., 1996] proposed a new recursive supervised learning algorithm, which combines the procedure of online candidate regressor selection with the conventional Givens QR based recursive parameter estimator to provide efficient adaptive supervised training. Mashor [Mashor, 2000] presented a new hybrid algorithm, moving K-means clustering algorithm for training RBF, which positions the RBF centres and Givens least square to estimate the weights. Sarimveis et al., [Sarimveis et al., 2002] proposed training methodology based on a fuzzy partition of the input space and combines self-organized and supervised learning. A new fast method for training RBF is used to model linear dynamical multi-input multi-output (MIMO) discrete-time systems. For a given fuzzy partition of the input space, the method is able to determine the proper network structure, without using a trial and error procedure. Sarimveis et al., [Sarimveis et al., 2003] presented an algorithm based on the subtractive clustering for training RBF, which is faster in training times and more accurate in prediction. Dumitrescu and Simon [Dumitrescu and Simon, 2003] proposed a dynamic clustering algorithm (GCDC) based on a new evolutionary optimization metaheuristics, the genetic chromo (GC) dynamics. This algorithm is used for designing RBF topologies. GCDC performed the clustering of training data thereby reducing the complexity of the network. Nabney [Nabney, 2004] showed that the RBF with logistic and softmax outputs can be trained efficiently using Fisher scoring algorithm. Han and Xi [Han and Xi, 2004] presented a new neural network called radial basis perceptron (RBP) for distinguishing different sets. RBP network is based on RBF and MLP. RBP has two hidden layers that are connected by selective connection. They presented an algorithm to train an RBP network. The algorithm alternately applied basis functions adaptation and back propagation training until a satisfactory error is achieved. Ravi et al., [Ravi et al., 2008] proposed evolving clustering method for the unsupervised phase and used linear least square technique for supervised phase. They called the resultant network semi-online RBF (SORBF). In the past, DE was used to train an MLP [Ilonen et al., 2003] and a wavelet neural network (WNN) [Chauhan et al., 2009].

4.2 Proposed Hybrid Approach

The proposed hybrid consists of two phases. In phase 1, we propose to employ DE for training the RBF in the supervised phase while keeping the unsupervised phase intact meaning that we used K-means clustering in that phase. In phase 2, the predicted outputs from DERBF network are taken along with input variables and fed as a new training set to the GATree to generate rules. The rules generated by GATree were tested on the validation set of the given classification problem. The hybrid is tested using 10-FCV.

In the proposed hybrid approach, we used the training data of every fold to train the DERBF. After the network is trained efficiently it is tested on the test data. It is only after getting high accuracies in the 1st phase (i.e. DERBF) then we invoked the 2nd phase. This scheme ensures that the rules extracted by GATree are truly representing the knowledge learnt by the DERBF in 1st phase. The outputs predicted by DERBF for the training data is fed to the GATree to generate the rules which are tested on the validation data.

4.2.1 DERBF - Training Algorithm

In unsupervised phase, the number of clusters and cluster centers are determined by using K-means clustering technique. The weights between the kernel layer and the output layer are determined by DE algorithm. Thus, the supervised part of training is treated as an optimization problem. DERBF architecture is depicted in Figure 4.1.

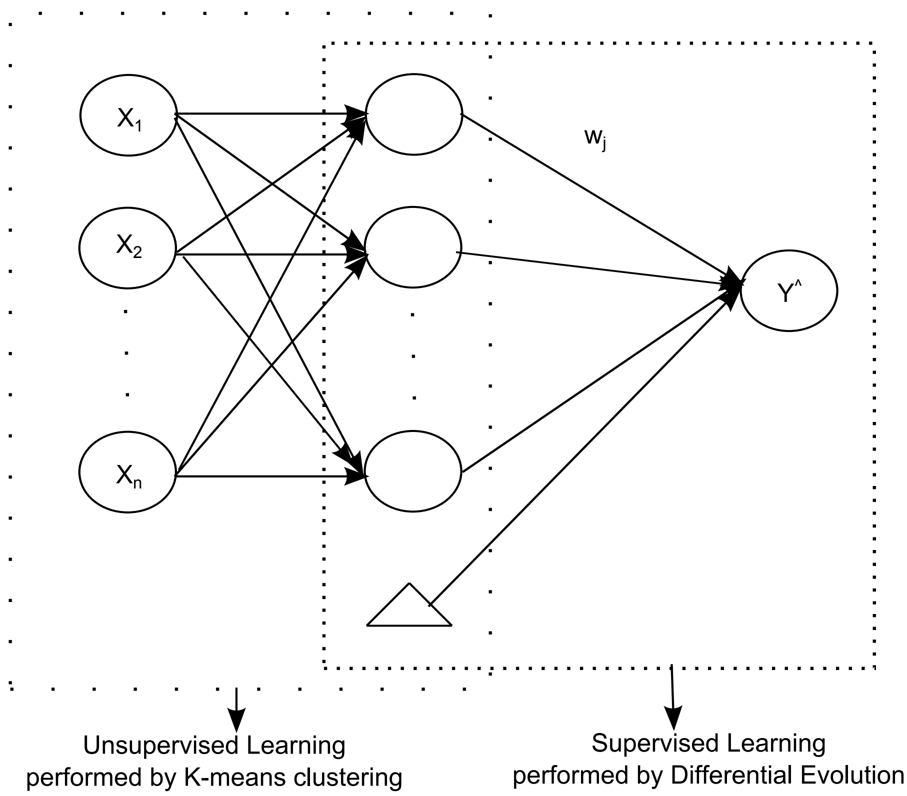

Figure 4.1: DERBF Architecture

The training algorithm for DERBF is as follows:

1. Compute the cluster centers by using K-means clustering. Initialize randomly the weights between the hidden and output layer in the range [10 10].

2. The output $\hat{y}_k$ of the samples is calculated in supervised part, where $\hat{y}_k = \sum_{j=1}^{M} w_j \phi_{ij}$, where where $k = 1, ..., n$ and n is the number of samples, M is the number of hidden nodes, w_j is the weights between hidden layer and output layer. ϕ_{ij} is calculated by using Gaussian activation function, $\phi_{ij} = \exp(-\frac{||X_i - C_j||^2}{2\sigma_j^2})$ where $i = 1, 2, ..., n$, $j = 1, 2, ..., M$ and scaling factor $\sigma_j = \frac{1}{p}\sqrt{\sum_{i=1}^{p}(C_j - C_i)^2}$, where p=2, C_i is the p-nearest neighbors of centroid C_j [Benoudjit and Verleysen, 2003].

3. By using DE, minimize the error E of prediction by estimating the w_j in supervised part, where the error function E is taken as a normalized room mean squared error defined as follows:

$$E = \sqrt{\frac{\sum_{k=1}^{n}(y_k - \hat{y}_k)^2}{\sum_{k=1}^{n} y_k^2}} \tag{4.6}$$

y_k is the actual output.

4. Repeat step 3, till maximum number of iterations is completed and/or the difference of E over successive generations is less than or equal to a pre-specified small value. Then the training process is completed.

During the training phase the input vector X and the output vector y are known, where $y = f(X, R)$, $R = (w_1, w_2, ..., w_M)$ and w_j is estimated by minimizing the error function E, which essentially approximates the relation between the inputs and the outputs.

DE is used to update the weights. DE is discussed in Appendix A.3.3. GATree is used to extract rule from DE trained RBF, which is completely described in Appendix B.5.4. The data flow diagram of the proposed architecture is show in Figure 4.2.

Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data. Generated rules are then tested against the validation set and the results are presented in Results and discussions section. Prediction accuracy of the rules is determined in terms of accuracy for classification problems.

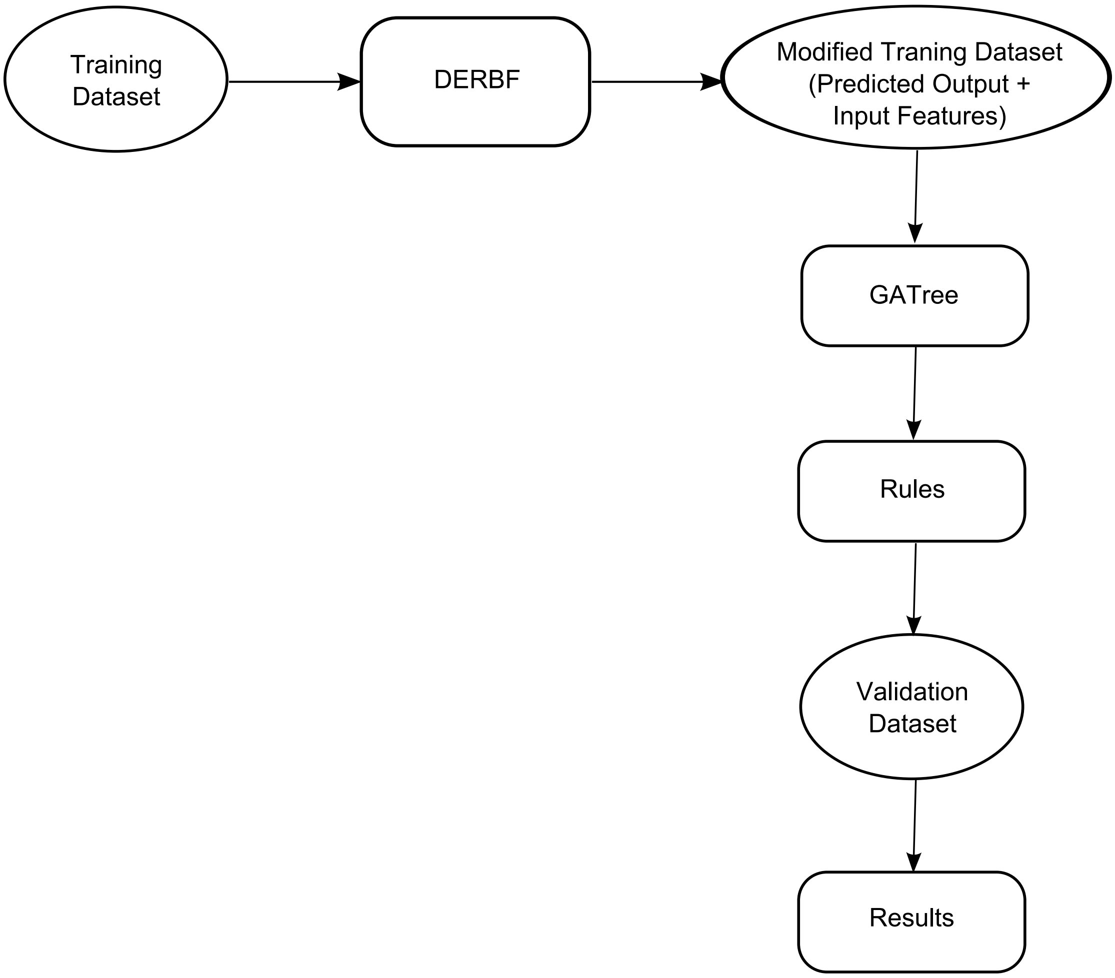

Figure 4.2: DERBF + GATree data flow diagram

4.3 Datasets Analyzed

We chose publicly available benchmark datasets from UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`) such as IRIS, WINE and WBC datasets.The effectiveness of the proposed model is tested over classification datasets of bank bankruptcy datasets such as Spanish banks [Olmeda and Fernndez, 1997], Turkish banks [Canbas et al., 2005], UK banks [Beynon and Peel, 2001] and US banks [Rahimian et al., 1996]. Information about the classification datasets analyzed is presented in Appendix C, out of the listed datasets in Table 1.1 for demonstration purpose.

4.4 Results and Discussions

We developed the DERBF in C language in Windows 7 environment on a desktop with 2 GB RAM.The user defined parameters related to DERBF are number of clusters, number of hidden nodes, scaling factor F, crossover factor CR, population size, number of generations, weights range and error tolerance. F in the range of 0.4 to 0.95, CR in the range 0.4 to 0.99, population size is taken as 30 and an error tolerance is taken as 10^6. Number of hidden nodes is equal to number of clusters plus one bias node. Number of clusters are varied between 3 and 6.

The user defined parameter related to GATree are maximum number of generations, maximum population size, crossover rate, mutation rate, percentage of genome replacement and error rate and finally crossover and mutation heuristics are taken as the standard ones only. Maximum number of generations is taken as 100, population size is also taken as 100 and crossover rate is taken in the range of 0.6 to 0.99 and mutation rate is taken in the range of 0.001 to 0.7. It is observed that the rules generated with this parameter combination yields the best tree size.

The average accuracies of 10-fold cross validation on classification datasets are presented in the Table 4.1. In the case of iris and wine datasets DERBF yielded better results than the proposed hybrid. In the case of Spanish dataset proposed hybrid achieved 89.56% which is close to the result yielded by stand alone DERBF. In the case of Turkish dataset DERBF yielded 100% sensitivity which is superior than the result achived by GATree upon DERBF. In the case of UK dataset DERBF achieved better sensitivity of 96.66% than that of the proposed hybrid which got 88.33%. However, in the case of US dataset GATree upon DERBF yielded better sensitivity of 55.36% than DERBF of 45.38%. Further, t-test is carried out whether the standalone and hybrid approach are statistically significant at 1% level or not. The t-test values are presented in Table 4.1. From the t-test values it is observed that in the case of bank datasets the proposed hybrid and stand alone DERBF are statistically insignificant. But in the case of iris and wine datasets they are statistically significant at 1% level of significance. Although, the accuracies are less when hybrid is compared with DERBF, we represent the knowledge in the form if-then rules. Hence our proposed approach is advantageous.

Table 4.1: Average results of 10FCV of DERBF + GATree

Dataset	DERBF			DERBF+GATree			t-Test values
	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*	
Iris			85.99			52.49	7.69
Wine			63.99			45.43	4.23
Spanish	93.33	45.71	67.68	89.56	25.26	57.66	0.06
Turkish	100	80	90	95	47.5	70.24	1.5
UK	96.66	74.99	85.82	88.33	43.33	65.82	1.98
US	45.38	97.41	70.38	55.36	97.69	76.54	2.46

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

Rules extracted using the proposed approach for classification problems are presented below:

IRIS DATASET

(1) If Petal width $>$ 0.041667 Then IRIS SETOSA

(2) If Petal width $>$ 0.375 and Petal width $>$ 0.166667 then IRIS SETOSA

(3) If Petal width $>$ 0.375 and Sepal Width $\leq$ 0.333333 then IRIS VERSICOLOR

(4) If Petal width $\leq$ 0.375 then IRIS VERGINICA

WINE DATASET

(1) If Proline $>$ 0.672043 then Class A

(2) If OD280/OD315 of diluted wines $>$ 0.836842 then Class A

(3) If OD280/OD315 of diluted wines $\leq$ 0.836842 and Nonflavanoid phenols $>$ 0.849582 then Class A

(4) If OD280/OD315 of diluted wines $\leq$ 0.836842 and Nonflavanoid phenols $\leq$ 0.849582 then Class B

SPANISH DATASET

(1) If CATA $>$ 0.284492 then Bankrupt

(2) If CATA $\leq$ 0.284492 and CSS $>$ 0.845858 then Bankrupt

(3) If CATA $\leq$ 0.284492 and CSS $\leq$ 0.845858 then NonBankrupt

TURKISH DATASET

(1) If (SEB+RR)/P $\geq$ 0.505291 then NonBankrupt

(2) If (SEB+RR)/P $<$ 0.505291 and IE/ANA $\geq$ 0.706522 then NonBankrupt

(3) If (SEB+RR)/P $<$ 0.505291 and IE/ANA $<$ 0.706522 then Bankrupt

(4) If (SEB+RR)/P $<$ 0.505291 and IE/ANA $<$ 0.706522 and LA/(D+NF) $\geq$ 0.249578 then Bankrupt

(5) If (SEB+RR)/P $<$ 0.505291 and IE/ANA $<$ 0.706522 and LA/(D+NF) $<$ 0.249578 and (SE+TI)/TA $\geq$ 0.470588 then NonBankrupt

UK DATASET

(1) If AGE $\geq$ 0.159091 then Bankrupt

(2) If AGE $<$ 0.159091 and CL/TA $\geq$ 0.466784 then NonBankrupt

(3) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $\geq$ 0.159091 and SALES $\geq$ 0.040275 and CA-CL/TA $\geq$ 0.81762 then Bankrupt

(4) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $\geq$ 0.159091 and SALES $\geq$ 0.040275 and CA-CL/TA $<$ 0.81762 then Bankrupt

(5) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $\geq$ 0.159091 and SALES $<$ 0.040275 then Bankrupt

(6) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $<$ 0.159091 and FF/TL $\geq$ 0.156573 then Bankrupt

(7) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $<$ 0.159091 and FF/TL $<$ 0.156573 and PBT/CE $\geq$ 0.430377 then NonBankrupt

(8) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $<$ 0.159091 and FF/TL $<$ 0.156573 and PBT/CE $<$ 0.430377 and AGE $\geq$ 0.386364 then NonBankrupt

(9) If AGE $<$ 0.159091 and CL/TA $<$ 0.466784 and AGE $<$ 0.159091 and FF/TL $<$ 0.156573 and PBT/CE $<$ 0.430377 and AGE $<$ 0.386364 and LAG $\geq$ 0.672131 then NonBankrupt

(10) If AGE < 0.159091 and CL/TA < 0.466784 and AGE < 0.159091 and FF/TL < 0.156573 and PBT/CE < 0.430377 and AGE < 0.386364 and LAG < 0.672131 and FF/TL ≥ 0.373439 then NonBankrupt

(11) If AGE < 0.159091 and CL/TA < 0.466784 and AGE < 0.159091 and FF/TL < 0.156573 and PBT/CE < 0.430377 and AGE < 0.386364 and LAG < 0.672131 and FF/TL < 0.373439 then Bankrupt

US DATASET

(1) If RE/TA ≥ 0.550543 then Bankrupt

(2) If RE/TA < 0.550543 and RE/TA ≥ 0.786827 then NonBankrupt

(3) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA ≥ 0.678122 and S/TA ≥ 0.152347 then NonBankrupt

(4) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA ≥ 0.678122 and S/TA < 0.152347 then NonBankrupt

(5) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA ≥ 0.566553 then Bankrupt

(6) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA < 0.566553 and ME/TA ≥ 0.026319 then NonBankrupt

(7) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA < 0.566553 and ME/TA < 0.026319 and ME/TA ≥ 0.032871 then NonBankrupt

(8) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA < 0.566553 and ME/TA < 0.026319 and ME/TA < 0.032871 and S/TA ≥ 0.159278 then NonBankrupt

(9) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA < 0.566553 and ME/TA < 0.026319 and ME/TA < 0.032871 and S/TA < 0.159278 and RE/TA ≥ 0.673432 then Bankrupt

(10) If RE/TA < 0.550543 and RE/TA < 0.786827 and EIT/TA < 0.678122 and WC/TA < 0.566553 and ME/TA < 0.026319 and ME/TA < 0.032871

and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $\geq$ 0.343019 then Bankrupt

(11) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $\geq$ 0.166727 then Bankrupt

(12) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $<$ 0.166727 and S/TA $\geq$ 0.31727 then Bankrupt

(13) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $<$ 0.166727 and S/TA $<$ 0.31727 and S/TA $\geq$ 0.175101 then NonBankrupt

(14) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $<$ 0.166727 and S/TA $<$ 0.31727 and S/TA $<$ 0.175101 and WC/TA $\geq$ 0.534228 then NonBankrupt

(15) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $<$ 0.166727 and S/TA $<$ 0.31727 and S/TA $<$ 0.175101 and WC/TA $<$ 0.534228 and RE/TA $\geq$ 0.742896 then NonBankrupt

(16) If RE/TA $<$ 0.550543 and RE/TA $<$ 0.786827 and EIT/TA $<$ 0.678122 and WC/TA $<$ 0.566553 and ME/TA $<$ 0.026319 and ME/TA $<$ 0.032871 and S/TA $<$ 0.159278 and RE/TA $<$ 0.673432 and S/TA $<$ 0.343019 and S/TA $<$ 0.166727 and S/TA $<$ 0.31727 and S/TA $<$ 0.175101 and WC/TA $<$ 0.534228 and RE/TA $<$ 0.742896 then Bankrupt

4.5 Conclusions

In this Chapter, we presented rule extraction from DE trained radial basis function network using GATree. From the numerical experiments conducted by us, we concluded that the proposed DERBF-GATree hybrid performed well in analyzing the bankruptcy datasets. Moreover, the GATree removes the black-box stigma attached to DERBF neural network by extracting *if-then* rules from the trained DERBF network. Even though the accuracy of the rules is less than that of the DERBF, our idea is to extract the rules from the trained DERBF network using GATree. The rules extracted by the proposed hybrid yielded good accuracies in all the datasets.

Chapter 5

Rule Extraction from GMDH Neural Network: Application to Banking

This Chapter presents a *pedagogical* rule extraction approach proposed which analyzes medium scale dataset pertaining to finance. Introduction is presented in the first section followed by customer relationship management in the second section. Churn prediction problems was presented in third section. In fourth section, proposed rule extraction method and then in fifth section description about the group method of data handling presented. Data description, data imbalance problems and datasets analyzed are presented in section six, seven and eight. finally results and discussion and conclusions of the chapter are presented in ninth and tenth sections.

5.1 Introduction

It is observed that rule extraction from Neural Networks has been successfully applied for small scale problems. On the other hand Neural Networks has shown superior performance dealing with large scale datasets. Further, Decision Tree (DT) employed to generate rules from Group Method of Data Handling (GMDH). 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 pedagogical rule extraction technique which analyzes medium scale 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 behavior. Rules extracted using GMDH for churn prediction problem can also be used as an early warning system that alerts the management about about-to-churn customers behavior. This is a very important application of analytical Customer Relationship Management (CRM) in finance.

5.2 Customer Relationship Management

CRM is a process or methodology used to learn more about customers need and behaviors 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:

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

Various definitions of CRM put emphasis on different perspectives. CRMs technological perspective was stressed in [Yuan and Chang, 2001]; [Peppers and Rogers, 1995]; [Shaw et al., 2001]; [Verhoef and Donkers, 2001], its knowledge management perspective was emphasized in [Massey et al., 2001].

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 organization 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 [Edwards, 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 organizations 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 behavior could help the organizations 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 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.

5.4 Overview of Group Method of Data Handling

GMDH, a self-organizing network, was introduced by Ivakhnenko in 1966 [Ivakhnenko, 1996] as an inductive learning algorithm for complex systems. Its main advantage is that it accurately estimates the parameters from its original structure. In fact, the GMDH network is not like regular feed-forward networks and was not originally represented as a network. The GMDH network is implemented with polynomial terms in the links and a genetic component to decide how many layers are built. The result of training at the output layer can be represented as a polynomial function of all or some of inputs. GMDH builds hierarchical solutions by trying for many easy models and by retaining the best and constructing on them to get the composition function.

The polynomial nodes are generally in quadratic form $z = w_0 + w_1x_1 + w_2x_2 + w_3x_1 + w_4x_2 + w_5x_1x_2$ for the inputs x_1 and x_2 , weight vector wand output node z . The weights are found by solving the linear regressing equation $z = y$, the response vector.

The learning process of GMDH is as follows: The GMDH develops on dataset with independent feature X_1, X_2,. . ., X_n and dependent variable y before learning processing is initiated. First the dataset is split into training and test set.

5.4.1 Description of algorithm

GMDH is a heuristic self organizing method that models the input-output relationship of a complex system using a multilayered Rosenblatts perception-type network structure. Each element in the network implements a non-linear equation of two inputs and its coefficients are determined by a regression analysis. Self selection thresholds are given at each layer in the network to delete those useless elements which can not estimate the correct output. Only those elements whose performance indices exceed the threshold are allowed to pass to succeeding layers, where more complex combination is formed. These steps are repeated until the convergence criterion is satisfied or a predetermined number of layers are reached. GMDH approach can be useful because:

− A small training set of data is required.

− The computational burden is reduced.

− The procedure automatically filters out input properties that provide little information about the location and shape of hyper surface.

− A multilayer structure is a computationally feasible way to implement multinomials of high degree.

GMDH algorithm can be represented as a set of neurons in which different pairs of them in each layer are connected through a quadratic polynomial and thus produce new neuron in the next layer. The input-output relationship perfectly and has been widely used as a complete description of the system model.

By combining the so called partial polynomial of two feature in the multilayer, the GMDH can easily solve these problems. In fact, the GMDH network is not like regular feed forward network and was not originally represented as a network.

The input-output relationship perfectly and has been widely used as a complete description of the system model. By combining the so called partial polynomial of two feature in the multilayer, the GMDH can easily solve these problems. In fact, the GMDH network is not like regular feed forward network and was not originally represented as a network. In Neuroshell2 [Karunanithi et al., 1992], the GMDH network is implemented with polynomial terms in the links and a genetic component to decide how many layers are built. The result of training at the output layer can be represented as a polynomial function of all or some of inputs [Ivakhnenko, 1996]. The main idea behind GMDH is that it is trying to build a function (called a polynomial model) that would behave in such a way that the predicted value of the output would be as close as possible to the actual value of output (Neuroshell2 tool: urlhttp://www.inf.kiew.ua/gmdh-home).

During the learning process GMDH is developed as follows:

1. Input layer, as usual, consists of independent feature.
2. When constructing the hidden layer, initial population of units is created. Each unit is in the form of Ivakhnenko polynomial form: $y = a + bx_1 + cx_2 + dx_1^2 + ex_1x_2 + fx_2^2$ or $y = a + bx_1 + cx_2 + dx_1x_2$, where y is the dependent variable, x_1, x_2 are independent feature and a, b, c, d, e, f are parameters. Parameters are estimated using training set.
3. The mean squared error (MSE) is calculated for the test set.
4. Units are sorted using the MSE values. The units with good approximation are taken to the next construction layers and the remaining are deleted.
5. Next layers are constructed while the MSE value decreases for the best units.
6. The response of the unit having minimum MSE value is considered as the output of GMDH.

In this network, the important input feature, number of hidden layers, neurons in each hidden layer are determined automatically. Majority of GMDH network implementations use regression analysis for solving the problem. The first step is to decide the type of polynomial that regression should find. General connection between input and output feature can be expressed by Volterra functional series, discrete analog of which is KolmogorovGabor polynomial. The next step is to construct a linear combination of all of the polynomial terms with variable coefficients. The algorithm determines values of these coefficients by minimizing the squared sum (over all samples) of differences between sample outputs and model predictions. GMDH architecture is shown in Figure 5.1.

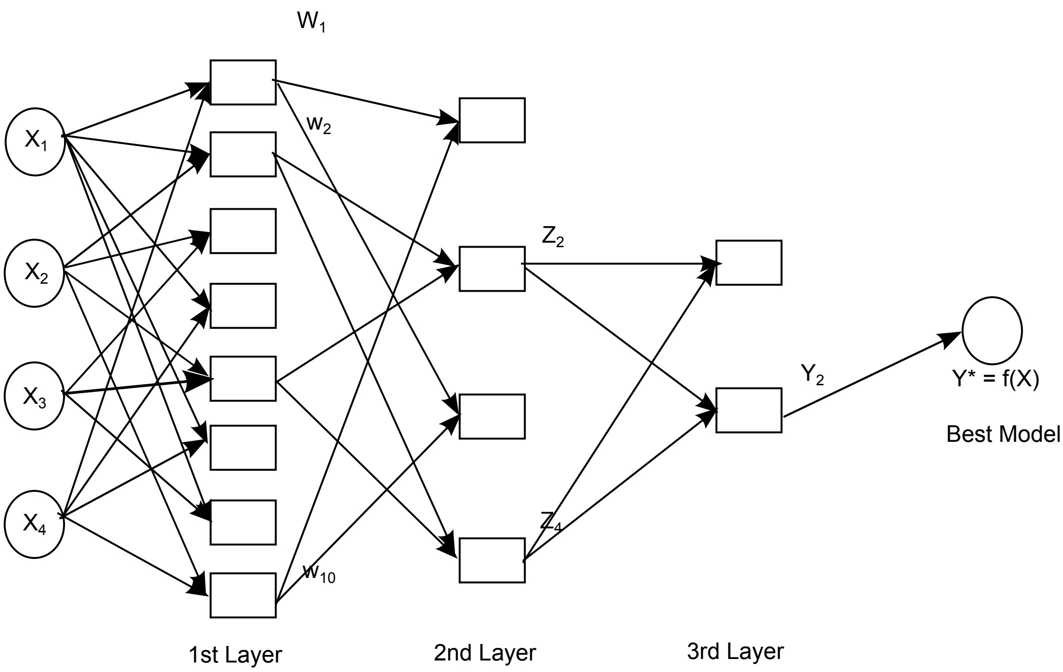

Figure 5.1: GMDH Architecture

5.4.2 Application of GMDH Algorithm

The GMDH algorithm (Ivakhnenko and Osipenko, 1982) can solve the problem of

– Long term forecasting

– Short term forecasting of processes and events

– Identification of physical regularities

– Approximation of multivariate processes

– Physical field extrapolation

– Data sampling clusterization

– Pattern recognition in case of continuous valued or discrete feature

– Diagnostics and recognition by probabilistic sorting out algorithms

– Model less processes forecasting using analogous complexing

– Self organization of twice multilayered neuronet with active neurons

5.5 Data Imbalance Problem

In many real time problems, almost all the instances belong to one class, while far fewer instances are labeled 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]; [Japkowicz and Stephen, 2002]; [Kubat et al., 1998]; [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., 1998], learning word pronunciation [Davel and Barnard, 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 [Snchez et al., 2009]. DT is discussed in Appendix B.5.4.

5.5.1 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., 1998].

In the earliest stages of this research, undersampling using condensed nearest neighbor (CNN) [Hart, 1968], Edited Nearest Neighbor (ENN) [Wilson, 1972], Selective undersampling using Tomak-Links concept [Kubat and Matwin, 1997], ENN with Neighborhood cleaning rule [Laurikkala, 2001] are proposed. Further, Chawla et al. [Chawla et al., 2002] proposed SMOTE (Synthetic Minority Oversampling Technique), where synthetic (artificial) samples are generated rather than oversampling by replacement. Maloof [Maloof et al., 2003] reported that sampling has the same results as moving the decision threshold or adjusting the cost matrix. Estabrooks et al., [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 and Pintelas, 2006]; [Kumar and Ravi, 2008]; [Guo et al., 2008]), 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 [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 [Guo et al., 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 [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.

Han et al., [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., [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., [Liu et al., 2006b] proposed SMOTE-Bootstrap hybrid (SMOTE-BU) method to deal with unbalanced data. Where, SMOTE is applied to oversample the minority instances and Bootstrap is applied to under-sample the majority instances.

5.5.2 Synthetic Minority Oversampling Technique

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 neighbors. SMOTE is used to widen the data region that corresponds to minority samples. This approach effectively forces the decision region of the minority class to become more general [Chawla et al., 2002].

5.5.3 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.

5.5.4 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 Proposed Rule Extraction Technique

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 [Chile, 2004], information about the dataset features is presented in Table C.5 in Appendix C. Balancing technique SMOTE is employed.

The proposed hybrid consists of two phases. The block diagram of the hybrid is depicted in Figure 5.2. In phase 1, GMDH was trained using training set. In phase 2, the predicted outputs from GMDH along with the independent feature are fed as new training set to the DT to generate rules. The rules generated were tested on the test data and also on validation set of the given classification problem.

Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data (see Section1.6 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 Section1.6 in Chapter 1. Prediction accuracy of the rules is determined in terms of accuracy for classification problems.

In the proposed hybrid approach, we trained the GMDH using SMOTE data.

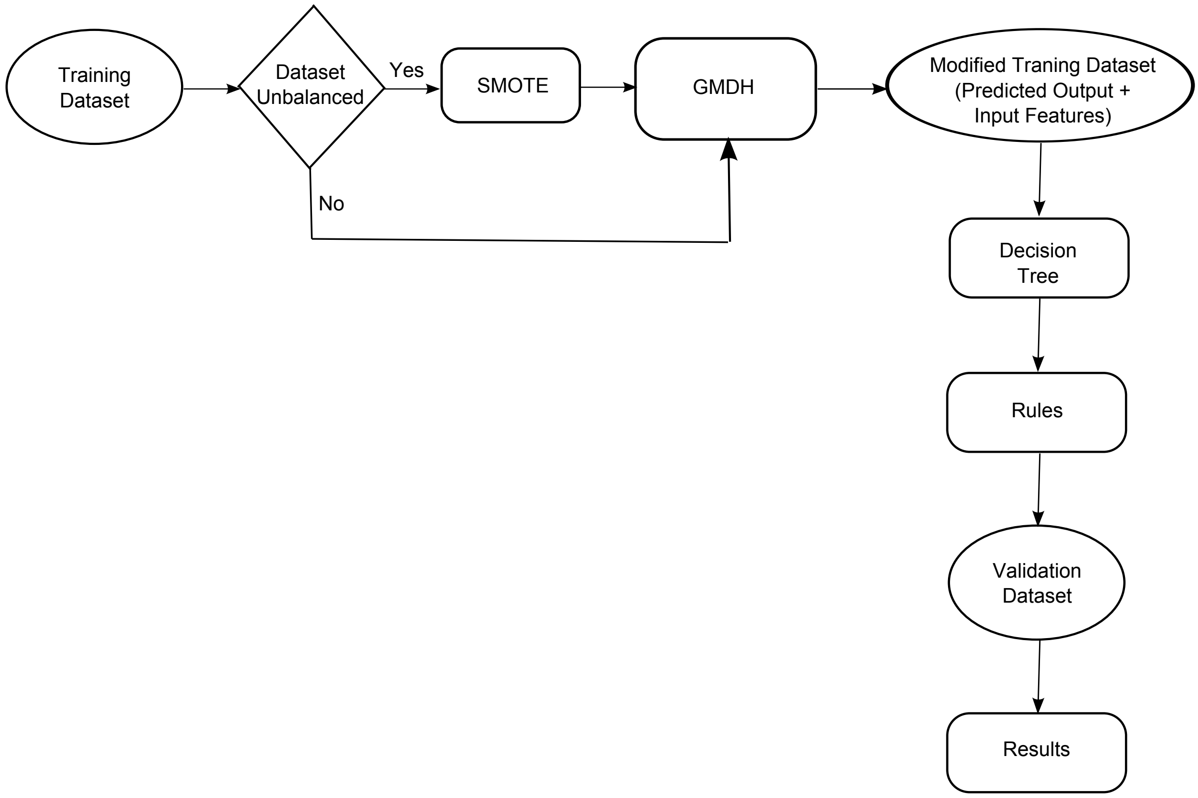

Figure 5.2: GMDH + DT data flow diagram

Once the network is trained well, it is tested on the test data and also on the validation set (20% dataset). After getting high accuracies in the 1st phase, we invoke 2nd phase. The rules generated by decision tree ensure that they represent the knowledge learnt by the GMDH network. The accuracy of rules is tested against the validation set. The difference between the DT and the proposed hybrid approach is that with DT we can extract rules directly from dataset. But with proposed approach, we extract rules from GMDH using DT. Rules extracted using the hybrid can be used as an early warning system. This feature removes the black box stigma over the GMDH.

5.7 Datasets Analyzed

We chose publicly available benchmark datasets from UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`) such as IRIS, WINE US Congressional and New Thyroid datasets. Further, a small scale data mining problem namely churn prediction datasets [Chile, 2004]. Information about the classification datasets analyzed is presented in Appendix C, out of the listed datasets in Table 1.1 for demonstration purpose.

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 feature are available for each customer: sociodemographic and behavioral data, which are described in Table C.10 in Appendix C. The dataset comprises of 22 feature, with 21 predictor feature 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.

5.8 Results and Discussion

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.

The average results of the 10-FCV of the benchmark datasets are presented in Table 5.1. In case of Iris dataset, hybrid yielded 93.33% of accuracy over the validation set which is higher when compared to the GMDH and DT alone. For Wine dataset, hybrid yields 98.00% which is very much close to the GMDH. In case of New Thyroid dataset hybrid yielded 88.80%, accuracy which is lesser when compared to GMDH and DT standalone techniques on validation set. In case of US Congressional dataset the GMDH and proposed hybrid yielded 73.33% accuracy over the validation set. t-Test performed indicates that on wine and US Congressional datasets, the difference between the hybrid and GMDH is statistically insignificant. However, our main motive is to represent the knowledge learnt by the GMDH in the form if-then rules. But in the case of Iris and New Thyroid t-test indicates that hybrid is better than GMDH. Although, the accuracies are less when hybrid is compared with GMDH, we represent the knowledge in the form if-then rules. Hence our proposed approach is advantageous. On the other hand, Fujimoto and Nakabayashi [Fujimoto and Nakabayashi, 2003] also used the US congressional dataset and extracted rules from GMDH which is not comparable with our technique because their experimental setup was different and also they extracted rules directly from the GMDH.

Table 5.1: Average accuracies for benchmark datasets

Dataset	Techniques	Accuracies	t-test values
IRIS	GMDH	91	3.45
	DT	93.33	
	Hybrid	92.66	
Wine	GMDH	98.85	0.84
	DT	98.00	
	Hybrid	97.42	
US Congressional	GMDH	73.33	0
	DT	73.33	
	Hybrid	95.55	
New Thyroid	GMDH	95.81	4.87
	DT	88.8	
	Hybrid	91.39	

Rules extracted using the proposed approach for benchmark problems are presented below:

IRIS DATASET

(1) If PW $\leq$ 0.5 then IRIS SETOSA

(2) If PW $>$ 0.5 AND PW $\leq$ 1.5 then IRIS VERSICOLOR

(3) If PW $>$ 0.5 AND PW $>$ 1.5 then IRIS VERGINICA

WINE DATASET

(1) If FLAVANOIDS $\leq$ 1.57 and COLOR $\leq$ 3.8 then Class B

(2) If FLAVANOIDS $\leq$ 1.57 and COLOR $>$ 3.8 then Class C

(3) If FLAVANOIDS $>$ 1.57 and ALCOHOL $\leq$ 12.72 then Class B

(4) If FLAVANOIDS $>$ 1.57 and ALCOHOL $>$ 12.72 and PROLINE $\leq$ 650 then Class B

(5) If FLAVANOIDS $>$ 1.57 and ALCOHOL $>$ 12.72 and PROLINE $>$ 650 then Class A

US Congressional DATASET

(1) If DUTY FREE EXPORTS $\leq$ 0 then REPUBLIC

(2) If DUTY FREE EXPORTS $>$ 0 then DEMOCRAT

New Thyroid DATASET

(1) If Thyroid-stimulating hormone $\leq$ 5.8 and Serumthyroxin $\leq$ 13.8 and Serumthyroxin $\leq$ 5.3 then HyperThyroid

(2) If Thyroid-stimulating hormone $\leq$ 5.8 and Serumthyroxin $\leq$ 13.8 and Serumthyroxin $>$ 5.3 then Normal

(3) If Thyroid-stimulating hormone $\leq$ 5.8 and Serumthyroxin $>$ 13.8 then HyperThyroid

(4) If Thyroid-stimulating hormone $>$ 5.8 then HypoThyroid

As regards churn prediction dataset, we note that many business decision makers place more emphasis on the sensitivity rather than the specificity because higher sensitivity leads to more accurately identifying the churners, thereby achieving the chief objective of CRM viz., retaining old loyal customers. Similar kind of arguments apply to some real-world problems like fraud detection in bank credit cards and telecom services, bankruptcy prediction, cancer detection in humans based on their genetic profiles etc. Table 5.2 presents the average results of the 10-fold cross-validation performed on standalone GMDH, proposed hybrid and standalone DT on the churn prediction dataset.

The standalone GMDH yields 80.89% sensitivity. The hybrid generated rules with 81.55% accuracy. t-Test performed on sensitivity indicates the difference

Table 5.2: Average accuracies for churn prediction dataset

Dataset	Techniques	Sens*	Spec*	Acc*	t-test values
Churn prediction	GMDH	80.89	88.74	88.21	1.70
	Hybrid	81.55	87.92	87.49	
	DT	81.65	93.50	92.70	

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

between GMDH and the hybrid is statistically insignificant. Even though the accuracy of the hybrid is reduced by 0.72% compared to the GMDH, the hybrid yielded the rules which can be easily interpreted and can be used as an early warning system. DT yielded 81.65% sensitivity and 92.70% accuracy on the validation set. Even though the DT yielded high accuracy, our main idea is to extract rules from the GMDH neural network. The rules of the best fold for churn prediction dataset are presented below.

(1) If NCC_T $\leq$ 0.932959 and T_WEB_T $\leq$ 5.977072 and CRED_T $\leq$ 630.21 and NCC_T $\leq$ 0.412727 then *Churner.*

(2) If NCC_T $\leq$ 0.932959 and T_WEB_T $\leq$ 5.977072 and CRED_T $\leq$ 630.21 and NCC_T $>$ 0.412727 and SX $\leq$ 0.966218 and SX $\leq$ 0.099431 then *Loyal customer.*

(3) If NCC_T $\leq$ 0.932959 and T_WEB_T $\leq$ 5.977072 and CRED_T $\leq$ 630.21 and NCC_T $>$ 0.412727 and SX $\leq$ 0.966218 and SX $>$ 0.099431 then *Churner.*

(4) If NCC_T $\leq$ 0.932959 and T_WEB_T $\leq$ 5.977072 and CRED_T $\leq$ 630.21 and NCC_T $>$ 0.412727 and SX $\leq$ 0.966218 and SX $>$ 0.966218 then *Loyal customer.*

(5) If NCC_T $\leq$ 0.932959 and T_WEB_T $\leq$ 5.977072 and CRED_T $>$ 630.21 then *Loyal Customer.*

(6) If NCC_T $\leq$ 0.932959 and T_WEB_T $>$ 5.977072 and CRED_T $\leq$ 592.937096 and NCC_T-2 $\leq$ 0.499881 then *Loyal Customer.*

(7) If NCC_T $\leq$ 0.932959 and T_WEB_T $>$ 5.977072 and CRED_T $\leq$ 592.937096 and NCC_T-2 $>$ 0.499881 then *Churner.*

(8) If NCC_T $\leq$ 0.932959 and T_WEB_T $>$ 5.977072 and CRED_T $>$ 592.937096 then *Loyal Customer*.

(9) If NCC_T $\leq$ 0.932959 and NCC_T $>$ 0.932959 and SX $\leq$ 0.899029 and SX $\leq$ 0.093913 then *Loyal Customer*.

(10) If NCC_T $\leq$ 0.932959 and NCC_T $>$ 0.932959 and SX $\leq$ 0.899029 and SX $>$ 0.093913 then *Churner*.

(11) If NCC_T $\leq$ 0.932959 and NCC_T $>$ 0.932959 and SX $>$ 0.899029 then *Loyal Customer*.

Table 5.3 presents the rule base size of the DT and proposed hybrid for the experiments conducted corresponding to the rules obtained for the best fold on the validation set. In the case of Churn Prediction dataset hybrid has yielded 11 rules whereas the DT yielded 18 rules. From this it is clearly shown that the hybrid approach has reduced the number of rule base size. In the case of Iris dataset also the number of rules is less for the hybrid model compared to DT. In other datasets the number of rules for both the hybrid and DT are equal.

Table 5.3: Average rule base size of DT and proposed hybrid approach for the best fold

Dataset	DT	Proposed Hybrid
Iris	4	3
Wine	5	3
US Congressional	2	2
New Thyroid	6	6
Churn Prediction	18	11

5.9 Conclusion

In this chapter a hybrid rule extraction approach from GMDH using DT to solve data mining problem like churn prediction in bank credit cards and also the benchmark datasets viz. Iris, Wine, US Congressional and New Thyroid datasets. Since the churn prediction dataset at hand is a highly unbalanced with 93.24% loyal and

6.76% churned customers, balancing technique SMOTE is employed to balance the data. Since identifying churner is most important from business perspective, by considering sensitivity alone the proposed hybrid performed well. The rules extracted by proposed hybrid yielded high accuracies.

Part III - Eclectic Approach

Chapter 6

Rule Extraction from DEWNN for solving Classification and Regressions problems

This Chapter presents an *eclectic* rule extraction approach proposed for solving classification and regression problems. First section presents the introduction about the wavelet followed by proposed rule extraction using wavelet neural network and feature selection. In third section, datasets analyzed by the proposed method are presented. Results and discussions are presented in fourth section. Finally section concludes the chapter.

6.1 Introduction

The word wavelet is due to Grossmann and Morlet [Grossmann and Morlet, 1984]. Wavelets are a class of functions used to localize a given function in both space and scaling (`http://mathworld.wolfram.com/Wavelet.html`). They have advantages over traditional Fourier methods in analyzing physical situations where the signal contains discontinuities and sharp spikes. Wavelets were developed independently in the fields of mathematics, quantum physics, electrical engineering and seismic geology. Interchanges between these fields during the last few years have led to many new wavelet applications such as image compression, turbulence, human vision, radar, chemistry and earthquake prediction.

Taking cue from the locally supported basis functions such as Radial basis function networks (RBFN), a class of neural networks called WNN, which originate from wavelet decomposition in signal processing, have become more popular recently [Zhang, 1997]. Wavelet networks employ activation functions that are dilated and translated versions of a single function $\psi : R^d \rightarrow R$, where d is the input dimension as stated in Zhang and Benvniste [Zhang and Benveniste, 1992] and Zhang [Zhang, 1997]. This function called the mother wavelet is localized both in the space and frequency domains [Becerra et al., 2005]. Based on wavelet theory, the WNN was proposed as a universal tool for functional approximation, which shows surprising effectiveness in solving the conventional problem of poor convergence or even divergence encountered in other kinds of neural networks. It can dramatically increase convergence speed compared to other networks as stated in Zhang et al., [Zhang et al., 2001].

WNN uses a gradient descent technique for training. Traditional gradient descent method suffers from well known drawbacks such as entrapment in local minimum, long convergence times and the need of differentiability of the objective function that are associated with calculus based optimization techniques. Consequently, WNN also suffers from these disadvantages. Therefore, Guangbin et al., [Guangbin et al., 2007] proposed the use of improved chaotic particle swarm optimization (ICPSO) and improved particle swarm optimization (IPSO) to tune both the structure and parameters of the WNN. Recently, Pan et al., [Pan and Chen, 2008] used genetic algorithm to optimize the WNN. Vinay Kumar et al., [Vinay Kumar et al., 2008] proposed TAWNN for estimating software development cost. Most recently, Chauhan et al., [Chauhan et al., 2009] proposed DE based training algorithm for WNN and call the resulting network as DEWNN.

6.2 Proposed Rule Extraction approach

In this study, proposed hybrid consists of two phases working in tandem. In phase 1, DEWNN is trained by using a training set. Then, feature selection [Chauhan et al., 2009] is performed to identify the salient features. We thoroughly exploited DEWNN [Chauhan et al., 2009] by using Gaussian and Morlet wavelet functions separately as activation functions for classification and regression problems. Once DEWNN is trained, its predicted output label along with the actual input features is fed to the phase 2. This results in a modified dataset. In phase 2, rule generation methods like DT, Ripper for classification and CART, DENFIS for regression problems respectively are invoked used to extract rules from the modified dataset. These rules signify the knowledge learnt by the DEWNN. These rules are applied on the validation dataset. Prediction accuracy of the rules is determined in terms of accuracy for classification problems and RMSE for regression problems on validation set. The data flow diagram of the proposed hybrid is depicted in Figure 6.1.

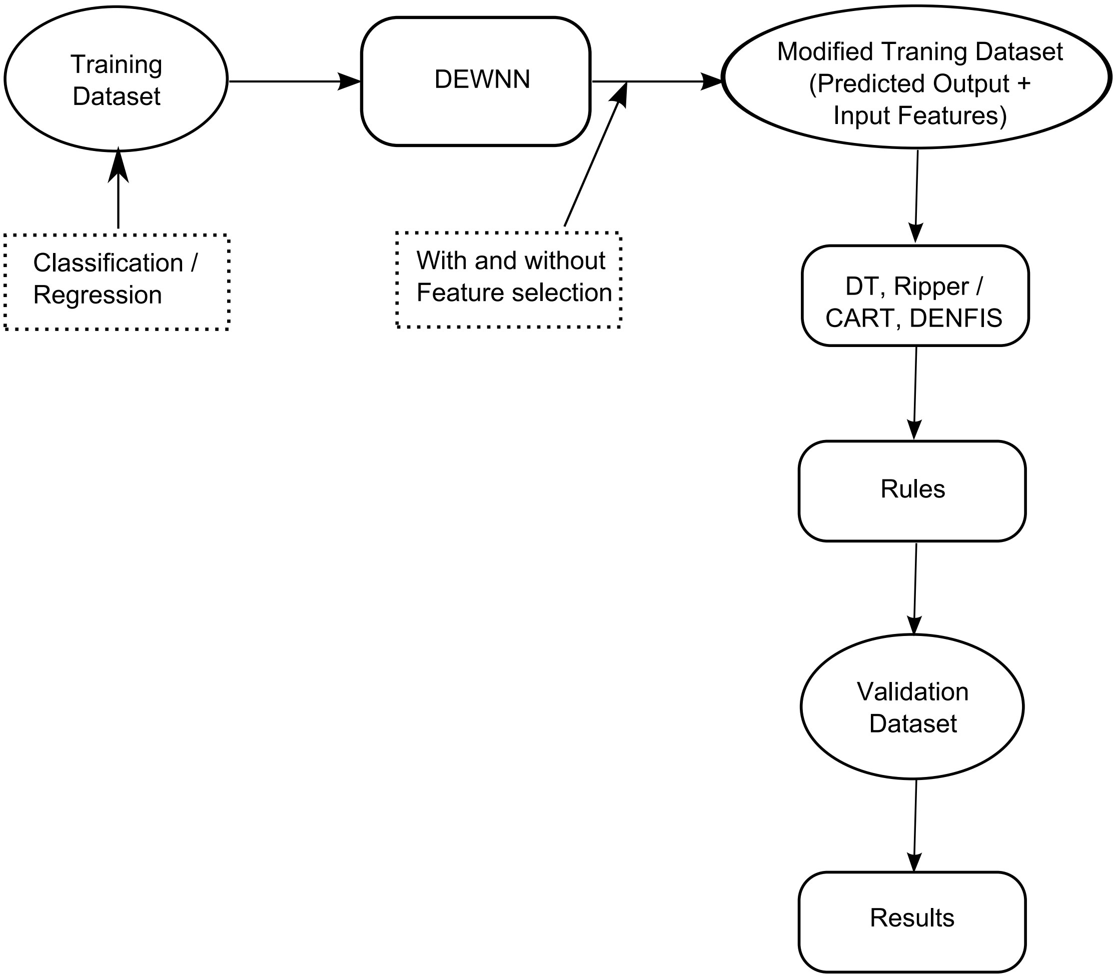

Figure 6.1: DEWNNRE data flow diagram

6.2.1 Wavelet Neural Network

A family of wavelets can be constructed from a function $\psi(x)$, sometimes known as a mother wavelet, which is confined in a finite interval. Daughter wavelets $\psi^{a,b}(x)$ are then formed by dilation (a) and translation (b). Wavelets are especially useful for compressing image data, since a wavelet transform has properties that are in some ways superior to a conventional Fourier transform (`http://mathworld.wolfram.com/Wavelet.html`). An individual wavelet is defined by

$$\psi^{a,b}(X) = \mid \alpha \mid^{-\frac{1}{2}} \psi \frac{x-b}{a} \tag{6.1}$$

Industrial processes can impose a number of problems upon the structures adopted for neural network dynamic modeling due to varying sampling times, sparse and dense data in different operating regions and the inherent presence of both large and small dynamics. In the case of non-uniformly distributed training data, an efficient way of solving this problem is by learning at multiple resolutions. A higher resolution of input space is used if the data is dense and a lower resolution when it is sparse.

Wavelets, in addition to forming an orthogonal basis, are capable of explicitly representing the behavior of a function at various resolutions of input variables. Consequently, a wavelet network is first trained to learn the mapping at the coarsest resolution level. In subsequent stages, the network is trained to incorporate elements of the mapping at higher and higher resolutions. Such hierarchical, multi resolution training has many attractive features for solving engineering problems, resulting in a more meaningful interpretation of the resulting mapping and more efficient training and adaptation of the network compared to conventional methods. The wavelet theory provides useful guidelines for the construction and initialization of networks and consequently, the training times are significantly reduced.The structure of the WNN with two input and six hidden nodes is shown in Figure 6.2.

The WNN consists of three layers: input layer, hidden layer and output layer. All the units in each layer are fully connected to the nodes in the next layer. The output layer contains a single unit. WNN is implemented here with the Gaussian wavelet function. The training algorithm [Zhang et al., 2001] for a WNN is as follows:

1. Select the number of hidden nodes required. Initialize the dilation and translation parameters for these nodes to some random values. Also initialize the

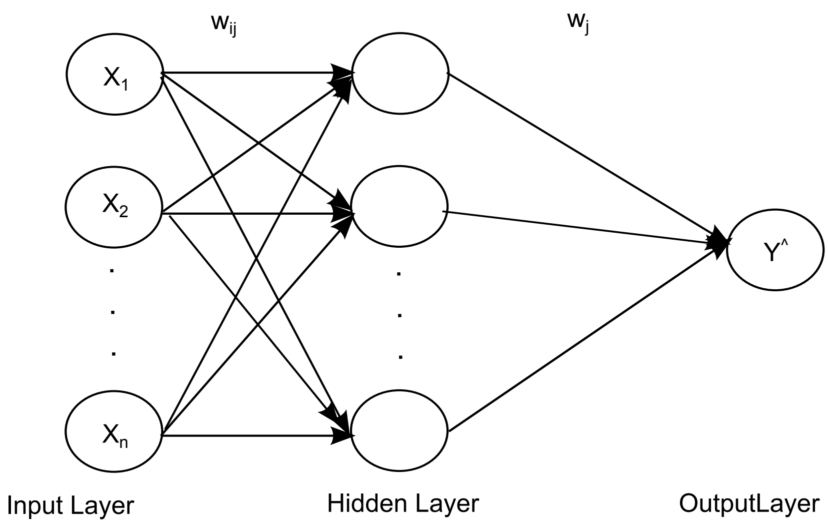

Figure 6.2: Wavelet Neural Network

weights for the connections between the input and hidden layer and also for the connections between the hidden and the output layer. It should be taken note that the random values should be limited to the interval (0, 1).

2. The output of the sample $V_k, k = 1, ..., np$, where np is the number of samples, is calculated with the following formula:

$$V_k = \sum_{j=1}^{nhn} w_j f\left(\frac{\sum_{i=1}^{nin} w_{ij} x_{ki} - b_j}{a_j}\right) \tag{6.2}$$

where $k = 1, ..., np, nin$ = number of input nodes and nhn = number of hidden nodes.In the above equation when $f(t)$ is taken as a Gaussian wavelet it has the following form:

$$f(t) = \exp(-t^2) \tag{6.3}$$

and when taken as Morlet wavelt it becomes

$$f(t) = cos(1.75t)\exp(-\frac{t^2}{2}) \tag{6.4}$$

3. Reduce the error of prediction by adjusting W_j, w_{ij}, a_j, b_j using $\triangle W_j$, $\triangle w_{ij}$, $\triangle a_j$, $\triangle b_j$. In the WNN, the gradient descend algorithm is employed.

$$\triangle W_j(t+1) = -\eta \frac{\partial E}{\partial W_j(t)} + \alpha \triangle W_j(t) \tag{6.5}$$

$$\triangle w_{ij}(t+1) = -\eta \frac{\partial E}{\partial w_{ij}(t)} + \alpha \triangle w_{ij}(t) \tag{6.6}$$

$$\triangle a_j(t+1) = -\eta \frac{\partial E}{\partial a_j(t)} + \alpha \triangle a_j(t) \tag{6.7}$$

$$\triangle b_j(t+1) = -\eta \frac{\partial E}{\partial b_j(t)} + \alpha \triangle b_j(t) \tag{6.8}$$

where the error function E is taken as normalized root mean squared deviation (NRMSE) as follows:

$$E = \sqrt{\frac{\sum_{j=1}^{n}(V_k - \hat{V}_k)^2}{\sum_{k=1} nV_k^2}} \tag{6.9}$$

where η and α are the learning and the momentum rates respectively.

4. Return to step (2), the process is continued until E satisfies the given error criteria, and the whole training of the WNN is completed.

Some problems exist in WNN such as slow convergence, searching space tapping in local minima and oscillation [Pan and Chen, 2008]. Chauhan et al., [Chauhan et al., 2009] employed DE to overcome these drawbacks. DE is used to update the weights between input and hidden layer, and hidden and output layer and also to update the dilation and translation parameters. DE is completely described in Appendix A.3.3.

6.2.2 Feature Selection

Feature selection is a process by which a sample in the measurement space is described by a finite and usually smaller set of number classed features. The features become components of the pattern space. Feature selection is regarded as a procedure to determine which variables (attributes) are to be measured either first or last. Guyon and Elisseeff [Guyon and Elisseeff, 2003] indicated that there are many potential benefits of feature selection: facilitating data visualization and data understanding, reducing the measurement and storage requirements, reducing training and utilization times, and defying the curse of dimensionality to improve prediction performance.

Nath et al., [Nath et al., 1997] applied the Garson [Garson, 1991] algorithm for feature selection while training MLP. We adopted the same algorithm and applied it in the context of WNN. Garsons algorithm for feature selection is presented as follows.

For a two-group classification, consider a neural network with q input nodes, r hidden nodes and one output node. Let $w_{ij}(i = 1, 2, ..., q; j = 1, 2, ..., r)$ represent the weight of the connection from ith input node to jth hidden node. Let $w_{ko}(k = 1, 2, ..., r)$ be the weight of the connection from kth hidden node to the output node.

The method to measure the importance of an input variable is to partition the hidden to output connection weights of each hidden node into components associated with each input node. The resulting weight associated with each input is a reflection of its importance.

1. Each hidden to output weight w_{ko}, irrespective of its sign, is incorporated into the input to hidden weights W_{ij} using the following expression:

$$W^*_{ij} = \frac{\mid W_{ij} \mid}{S_j} * (\mid w_{ko} \mid) \tag{6.10}$$

   Where $S_j = \sum_{i=1}^{p} \mid W_{ij} \mid$, i=1...q and j=1...r || represents absolute value.

2. For each hidden node j, the sum of weights over all input nodes is equal to the hidden to output node weight w_{jo}.

$$W_{jo} = \sum_{i=1}^{q} w^q_{ij} \tag{6.11}$$

3. For each input node, the adjusted weights W^*_{ij} are summed over all hidden nodes and converted to a percentage of the total for all input nodes.

$$r_i = \frac{W^*_{in}}{w_{jo}} * 100 \tag{6.12}$$

This percentage serves as a measure of the importance of the variable represented by the input node.

The features are thus ranked according to their importance and the top 50% features in each fold are considered. Once we select the top 50% features in ten folds, we calculate the frequency of occurrence of each feature across all folds and the features with highest frequency of occurrence are considered as important ones. Thus, the 50% most important features are selected and an optimal feature subset is formed.

The full features and selected features are fed to the DT / Ripper and DENFIS / CART for the rule generation purpose of classification and regression problems respectively. DT / Ripper and DENFIS / CART rule generation methods which are completely described in Appendix B.5.4. Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data. Generated rules are then tested against the validation set. Prediction accuracy of the rules is determined in terms of accuracy for classification problems and MSE for regression problems on validation set.

6.3 Datasets Analyzed

We chose the classification problems of bank bankruptcy datasets such as Spanish banks [Olmeda and Fernndez, 1997], Turkish banks [Canbas et al., 2005], UK banks [Beynon and Peel, 2001] and US banks [Rahimian et al., 1996] and publicly availabale benchmark datasets from UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`) such as IRIS and WINE datasets.

In addition to the classification problems, we also analyzed regression problems viz., Auto MPG dataset [Asuncion and Newman, 2007], Boston Housing dataset [Asuncion and Newman, 2007], Forest Fires dataset [Asuncion and Newman, 2007] and Body Fat dataset [Penrose et al., 1985], Pollution dataset [McDonald and Schwing, 1973]. Information about the classification and regression datasets analyzed is presented in Appendix C, out of the listed datasets in Table

1.1 for demonstration purpose.

6.4 Results and Discussions

The parameters of the DEWNN viz., crossover rate and mutation rate are varied between 0.3 and 0.8, 0.3 and 0.65 respectively. Population size is fixed at 30 and the number of hidden neurons is varied between 3 and 7. The important feature of the classification datasets are selected by using Garsons algorithm in DEWNN are tabulated in Table 6.1 with respect to the Gaussian and Morlet function in DEWNN.

Also, the important feature of the regression datasets are selected by using garsons algorithm in DEWNN are tabulated in Table 6.2 with respect to the Gaussian and Morlet function in DEWNN.

The average results of 10FCV of the proposed hybrid with full features and reduced features datasets using Gaussian and Morlet wavelet activation function for classification problems are presented in Tables 6.3 and 6.4 respectively. From Tables 6.3 and 6.4, it is observed that the proposed hybrids DEWNN + DT and DEWNN + Ripper performed well against the validation dataset. t-test is performed on the sensitivity between the hybrids and t-statistic value is tabulated in Tables 6.3 and 6.4. From the t-statistic values, it is observed that there is no statistically significant difference between the hybrids at 1% level of significance. Further, from the t-test performed separately, it is observed that in majority of the cases in classification problems without feature selection, there is no statistically significant difference between the hybrids and the DEWNN at 1% level of significance. The same is observed in the case of reduced feature set also.

Rules extracted using the proposed approach for classification problems over full features when Gaussian function is used in DEWNN are presented below:

Rule extracted using DEWNN+DT:

IRIS DATASET

(1) If Petal width ≤ 0.5 then IRIS SETOSA

(2) If Petal width > 0.5 and Petal length ≤ 4.7 then IRIS VERSICOLOR

(3) If Petal width > 0.5 and Petal length > 4.7 then IRIS VERGINICA

Table 6.1: Important features selected in classification datasets

Dataset	Feature selected	
	Gaussian Function	**Morlet Function**
Wine	ASH	ASH
	ALA-ASH	ALA-ASH
	PHENOLS	FLAVANOIDS
	FLAVANOIDS	HUE
	HUE	OD280
	PROLINE	PROLINE
Spanish	CAcash/totalA	Current assets/total assets
	Netincome/totalA	Current assets-cash/total assets
	Net income/TEquityCapital	Net income/total assets
	Net income/loans	Cost of sales/sales
	Cost of sales/sales	Cash flow/loans
Turkish	Interest expenses/average profitable assets	(Share holders equity + total income)/ (deposits + non-deposit funds)
	Interest income/interest expenses	Interest income/interest expenses
	Networking capital/total assets	(Share holders equity + total income)/total assets
	Liquid assets/(deposits + non-deposit funds)	Liquid assets/(deposits + non-deposit funds)
	Interest expenses/total expenses	Interest expenses/total expenses
	Liquid assets/total assets	Standard capital ratio
UK	Profit before tax/capital employed	Sales
	Funds flow/total liabilities	Funds flow/total liabilities
	Current assets - stock/ current liabilities	(Current liabilities + long-term debit)/total assets
	Current assets - current liabilities /total assets	Current assets/ current liabilities
	LAG (number of days between account year end and the date of annual report)	Current assets - current l liabilities/total assets

WINE DATASET

(1) If Nonflavanoid phenols ≤ 1.39 and Hue ≤ 3.4 then Class B

(2) If Nonflavanoid phenols ≤ 1.39 and Hue > 3.4 then Class c

(3) If Nonflavanoid phenols > 1.39 and Proline ≤ 735 then Class B

(4) If Nonflavanoid phenols > 1.39 and Proline > 735 and Hue ≤ 3.58 then Class B

Table 6.2: Important features selected in regression datasets

Dataset	Feature selected	
	Gaussian Function	Morlet Function
AutoMPG	Cylinders	displacement
	displacement	horsepower
	horsepower	weight
	weight	model year
Bodyfat	density	density
	height	weight
	chest	neck
	abdomen	chest
	thigh	thigh
	ankle	knee
	forearm	biceps
BostonHousing	crim	crim
	nox	age
	rm	dis
	tax	tax
	istat	istat
ForestFires	x-axis	ffmc
	ffmc	dc
	dmc	isi
	temperature	temperature
	rh	wind
	rain	rain
Pollution	prec	prec
	jult	ovr
	ovr	educ
	nonw	nonw
	wwdrk	poor
	nox	nox
	humid	humid

(5) If Nonflavanoid phenols $>$ 1.39 and Proline $>$ 735 and Hue $>$ 3.58 then Class A

SPANISH DATASET

(1) If CS/S $\leq$ 0.906 then NonBankrupt

(2) If CS/S $>$ 0.906 then Bankrupt

TURKISH DATASET

Table 6.3: Average results of 10FCV of full features datasets for classification problems

Dataset	DEWNN			DEWNN +DT			DEWNN+RIPPER			t-TEST VALUES
Gaussian function										
	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*	
Iris	NA	NA	92.99	NA	NA	91.99	NA	NA	93.99	2.01
Wine	NA	NA	92.56	NA	NA	92.56	NA	NA	93.99	0.46
Spanish	86.66	48.56	66.14	64.99	98.57	83.07	68.33	98.57	84.60	0.40
Turkish	84.16	97.5	90	50	100	75	50	100	75	0
US	93.84	83.07	88.45	97.69	83.07	90.37	94.61	85.38	89.99	1.32
UK	74.99	74.99	74.99	66.66	71.66	69.16	60.83	63.33	65.83	0.51
Morlet function										
IRIS	NA	NA	93.99	NA	NA	85.99	NA	NA	89.32	0.44
WINE	NA	NA	90.56	NA	NA	96.56	NA	NA	95.13	0.73
SPANISH	89.99	54.23	70.76	71.66	100	86.91	71.66	100	86.91	0.00
TURKISH	85	95	90	50	100	75	55	100	77.5	1.5
US	92.23	76.14	83.84	97.69	85.37	91.53	96.92	83.84	90.37	0.32
UK	79.99	72.06	75.83	71.66	69.99	70.83	66.66	71.66	69.16	0.50

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

(1) If NC/TA $\leq$ -4.8 then Bankrupt

(2) If NC/TA $>$ -4.8 then NonBankrupt

US DATASET

(1) If RE/TA $\leq$ 0.2862 and EIT/TA $\leq$ 0.1015 then Bankrupt

(2) If RE/TA $\leq$ 0.2862 and EIT/TA $>$ 0.1015 then NonBankrupt

(3) If RE/TA $>$ 0.2862 then NonBankrupt

UK DATASET

(1) If LAG $\leq$ 147 then Bankrupt

(2) If LAG $>$ 147 and CA-S/CL $\leq$ 0.5667 then NonBankrupt

(3) If LAG $>$ 147 and CA-S/CL $>$ 0.5667 then Bankrupt

Rule extracted using DEWNN+Ripper:

IRIS DATASET

(1) If Petal width $\geq$ 0.5 and Petal length $\geq$ 4.8 then IRIS VERGINICA

(2) If Sepal width $\leq$ 2.5 and Petal length $\leq$ 4.5 then IRIS VERGINICA

(3) If Petal length $\geq$ 3 then IRIS VERSICOLOR

(4) If Petal lengthPL < 3 then IRIS SETOSA

WINE DATASET

(1) If Nonflavanoid phenols $\leq$ 1.39 and Hue $\leq$ 3.85 then Class C

(2) If Malic acid $\leq$ 13.03 then Class B

(3) If Color intensity $\leq$ 0.73 then Class B

(4) If Color intensity > 0.73 then Class A

SPANISH DATASET

(1) If CF/L $\leq$ 0.0092 then Bankrupt

(2) If CF/L > 0.0092 then NonBankrupt

TURKISH DATASET

(1) If NC/TA $\leq$ -4.8 then Bankrupt

(2) If NC/TA > -4.8 then NonBankrupt

US DATASET

(1) If RE/TA $\geq$ 0.1635 and EIT/TA $\geq$ 0.0543 then NonBankrupt

(2) If RE/TA < 0.1635 and EIT/TA < 0.0543 then Bankrupt

UK DATASET

(1) If PBT/CE $\geq$ 4.408 then Bankrupt

(2) If CA-S/CL $\geq$ 1.0913 then Bankrupt

(3) If CA-S/CL < 1.0913 then NonBankrupt

Rules extracted using the proposed approach for classification problems over full features when Morlet function is used in DEWNN are presented below:

Rule extracted using DEWNN+DT:

IRIS DATASET

(1) If PW $\leq$ 0.5 then IRIS SETOSA

(2) If PW $>$ 0.5 and PL $\leq$ 4.7 then IRIS VERGINICA

(3) If PW $>$ 0.5 and PL $>$ 4.7 then IRIS VERSICOLOR

WINE DATASET

(1) If Nonflavanoid phenols $\leq$ 2.17 and Hue $\leq$ 3.8 then Class B

(2) If Nonflavanoid phenols $\leq$ 2.17 and Hue $>$ 3.8 and Alcalinity of ash $\leq$ 2.02 then Class B

(3) If Nonflavanoid phenols $\leq$ 2.17 and Hue $>$ 3.8 and Alcalinity of ash $>$ 2.02 then Class C

(4) If Nonflavanoid phenols $>$ 2.17 and Proline $\leq$ 720 then Class B

(5) If Nonflavanoid phenols $>$ 2.17 and Proline $>$ 720 then Class A

SPANISH DATASET

(1) If CF/L $\leq$ 0.0092 then Bankrupt

(2) If CF/L $>$ 0.0092 then NonBankrupt

TURKISH DATASET

(1) If IE/APA $\leq$ 29.4 then NonBankrupt

(2) If IE/APA $>$ 29.4 then Bankrupt

US DATASET

(1) If RE/TA $\leq$ 0.166 then Bankrupt

(2) If RE/TA $>$ 0.166 and EIT/TA $\leq$ 0.0695 and RE/TA $\leq$ 0.3484 then Bankrupt

(3) If RE/TA $>$ 0.166 and EIT/TA $\leq$ 0.0695 and RE/TA $>$ 0.3484 then Non-Bankrupt

(4) If RE/TA $>$ 0.166 and EIT/TA $>$ 0.0695 then NonBankrupt

UK DATASET

(1) If CA-CL/TA $\leq$ 0.1169 and PBT/CE $\leq$ 9.4607 then NonBankrupt

(2) If CA-CL/TA $\leq$ 0.1169 and PBT/CE $>$ 9.4607 then Bankrupt

(3) If CA-CL/TA $>$ 0.1169 then Bankrupt

Rule extracted using DEWNN+Ripper:

IRIS DATASET

(1) If PW $\geq$ 1.7 then IRIS VERSICOLOR

(2) If PL $\leq$ 1.9 then IRIS SETOSA

(3) If PL $>$ 1.9 then IRIS VERGINICA

WINE DATASET

(1) If OD280/OD315 of diluted wines $\leq$ 0.89 and Nonflavanoid phenols $\leq$ 1.84 then Class C

(2) If Proline $\geq$ 760 and Malic acid $\geq$ 12.85 then Class A

(3) If Proline $<$ 760 and Malic acid $<$ 12.85 then Class B

SPANISH DATASET

(1) If CF/L $\leq$ 0.0092 then Bankrupt

(2) If CF/L $>$ 0.0092 then NonBankrupt

TURKISH DATASET

(1) If IE/APA $\geq$ 31.799 then Bankrupt

(2) If IE/APA $<$ 31.799 then NonBankrupt

US DATASET

(1) If EIT/TA $\geq$ 0.1076 then NonBankrupt

(2) If RE/TA $\geq$ 0.2366 and ME/TA $\geq$ 0.5863 then NonBankrupt

(3) If RE/TA $<$ 0.2366 and ME/TA $<$ 0.5863 then Bankrupt

UK DATASET

Table 6.4: Average results of 10FCV of reduced features datasets for classification problems

Dataset	DEWNN			DEWNN +DT			DEWNN+RIPPER			t-TEST VALUES
Gaussian function										
	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*	Sens*	Spec*	Acc*	
Wine	NA	NA	96.28	NA	NA	93.42	NA	NA	91.99	0.67
Spanish	93.33	88.56	90.76	83.33	98.57	91.53	83.33	98.57	91.53	0
Turkish	95	95	95	50	100	75	50	100	75	0
UK	71.66	56.66	64.16	75	53.33	64.16	68.33	51.66	59.99	0.82
Morlet function										
WINE	NA	NA	92.85	NA	NA	92.56	NA	NA	91.42	0.30
SPANISH	88.33	69.995	78.45	73.33	100	87.68	75	100	88.45	0.26
TURKISH	87.5	92.5	90	62.5	100	81.25	52.5	100	76.25	2.05
UK	78.32	63.329	70.83	76.66	66.66	71.66	78.33	65	71.66	0.29

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

(1) If CA/CL $\leq$ 1.0696 and LAG $\geq$ 183 then NonBankrupt

(2) If CA/CL $\leq$ 1.0696 and LAG $<$ 183 then Bankrupt

Rules extracted using the proposed approach for classification problems over reduced features when Gaussian function is used in DEWNN are presented below:

Rule extracted using DEWNN+DT:

WINE DATASET

(1) If Flavanoids $\leq$ 0.43367 and Total phenols $\leq$ 0.30081 and Magnesium $\leq$ 0.31646 then Class C

(2) If Flavanoids $\leq$ 0.43367 and Total phenols $\leq$ 0.30081 and Magnesium $>$ 0.31646 then Class B

(3) If Flavanoids $\leq$ 0.43367 and Total phenols $>$ 0.30081 then Class B

(4) If Flavanoids $>$ 0.43367 then Class A

SPANISH DATASET

(1) If NI/TA $\leq$ 0.52975 then NonBankrupt

(2) If NI/TA $>$ 0.52975 then Bankrupt

TURKISH DATASET

(1) If (SE+TI)/(D+NF) $\leq$ 0.88973 then Bankrupt

(2) If (SE+TI)/(D+NF) $>$ 0.88973 then NonBankrupt

UK DATASET

(1) If PBT/CE $\leq$ 0.53937 and FF/TL $\leq$ 0.20862 then NonBankrupt

(2) If PBT/CE $\leq$ 0.53937 and FF/TL $>$ 0.20862 then Bankrupt

(3) If PBT/CE $>$ 0.53937 then Bankrupt

Rule extracted using DEWNN+Ripper:

WINE DATASET

(1) If Magnesium $\leq$ 0.18565 and Alcalinity of ash $\leq$ 0.32759 then Class C

(2) If Total phenols $\leq$ 0.22764 then Class C

(3) If Flavanoids $\geq$ 0.42939 then Class A

(4) If Magnesium $\geq$ 0.55907 then Class A

(5) If Magnesium $<$ 0.55907 then Class B

SPANISH DATASET

(1) If CAC/TA $\leq$ 0.81298 then Bankrupt

(2) If CAC/TA $>$ 0.81298 then NonBankrupt

TURKISH DATASET

(1) If IE/APA $\geq$ 0.18992 then Bankrupt

(2) If IE/APA $<$ 0.18992 then NonBankrupt

UK DATASET

(1) If Sales $\leq$ 0.52566 then NonBankrupt

(2) If Sales $>$ 0.52566 then Bankrupt

Rules extracted using the proposed approach for classification problems over reduced features when Morlet function is used in DEWNN are presented below:

Rule extracted using DEWNN+DT:

WINE DATASET

(1) If Total phenols ≤ 0.32234 and Malic acid ≤ 0.31551 then Class B

(2) If Total phenols ≤ 0.32234 and Malic acid > 0.31551 then Class C

(3) If Total phenols > 0.32234 and Flavanoids ≤ 0.33666 then Class B

(4) If Total phenols > 0.32234 and Flavanoids > 0.33666 then Class A

SPANISH DATASET

(1) If CA/L ≤ 0.81298 then Bankrupt

(2) If CA/L > 0.81298 then NonBankrupt

TURKISH DATASET

(1) If (SE+TI)/(D+NF) ≤ 0.67283 then Bankrupt

(2) If (SE+TI)/(D+NF) > 0.67283 and IE/ANA ≤ 0.36759 then Bankrupt

(3) If (SE+TI)/(D+NF) > 0.67283 and IE/ANA > 0.36759 then NonBankrupt

UK DATASET

(1) If PBT/CE ≤ 0.51839 and (CL+LTD)/TA ≤ 0.16933 then NonBankrupt

(2) If PBT/CE ≤ 0.51839 and (CL+LTD)/TA > 0.16933 then Bankrupt

(3) If PBT/CE > 0.51839 then Bankrupt

Rule extracted using DEWNN+Ripper:

WINE DATASET

(1) If Total phenols ≤ 0.32234 and Magnesium ≤ 0.33333 then Class C

(2) If Flavanoids ≥ 0.34379 and Alcalinity of ash ≥ 0.41983 then Class A

(3) If Flavanoids ≥ 0.34379 and Alcalinity of ash < 0.41983 then Class B

SPANISH DATASET

(1) If CA/L $\leq$ 0.81298 then Bankrupt

(2) If CA/L $>$ 0.81298 then NonBankrupt

TURKISH DATASET

(1) If IE/APA $\leq$ 0.43767 then Bankrupt

(2) If IE/ANA $\leq$ 0.36759 then Bankrupt

(3) If IE/ANA $>$ 0.36759 then NonBankrupt

UK DATASET

(1) If PBT/CE $\geq$ 0.51641 then Bankrupt

(2) If (CL+LTD)/TA $\geq$ 0.1765 and PBT/CE $\leq$ 0.33163 then Bankrupt

(3) If (CL+LTD)/TA $\geq$ 0.1765 and PBT/CE $>$ 0.33163 then NonBankrupt

Further, the average mean squared error (MSE) of 10FCV of full features and reduced features datasets using Gaussian and Morlet functions for regression problems is presented in Tables 6.5 and 6.6 respectively. From Tables 6.5 and 6.6, it is observed that DEWNN+CART and DEWNN+DENFIS performed well on the validation set. t-test is performed on sensitivity between the hybrids and values are presented in Tables 6.5 and 6.6. From the t-statistic values, it is observed that there is no statistically significant difference between the hybrids at 1% level of significance. Further, from the t-test performed separately, it is observed that the performance of DEWNN is statistically significant between compared to hybrids at 1% level of significance.

Table 6.5: Average MSE values of 10FCV of full features datasets for regression problems

Dataset	DEWNN	DEWNN+CART	DEWNN+DENFIS	t-TEST VALUES
Gaussian function				
AUTOMPG	0.0111	0.109873	0.10819	0.49
BODYFAT	0.0056	0.075554	0.06196	1.28
BOSTONHOUSING	0.0179	0.189722	0.12261	1.03
FORESTFIRES	0.009	0.060594	0.08172	0.25
POLLUTION	0.0125	0.116408	0.10526	1.11
Morlet function				
AUTOMPG	0.0161	0.120471	0.1209	0.05
BODYFAT	0.0106	0.104273	0.08386	1.09
BOSTONHOUSING	0.0256	0.199331	0.14498	1.02
FORESTFIRES	0.0089	0.059748	0.09381	0.37
POLLUTION	0.0167	0.11909	0.12551	0.54

Table 6.6: Average MSE values of 10FCV of reduced features datasets for regression problems

Dataset	DEWNN	DEWNN+CART	DEWNN+DENFIS	t-TEST VALUES
Gaussian function				
AUTOMPG	0.0512	0.192196	0.19426	0.03
BODYFAT	0.0849	0.222867	0.21294	0.10
BOSTONHOUSING	0.0509	0.192392	0.19028	0.03
FORESTFIRES	0.002	0.028401	0.02865	0.01
POLLUTION	0.024	0.129867	0.1426	0.50
Morlet function				
AUTOMPG	0.0252	0.154617	0.15394	0.06
BODYFAT	0.0237	0.160615	0.1315	2.74
BOSTON HOUSING	0.0323	0.176827	0.17214	0.44
FORESTFIRES	0.0024	0.028749	0.02845	0.01
POLLUTION	0.0367	0.15929	0.16188	0.06

Rules extracted using the proposed approach for regression problems over full features when Gaussian function is used in DEWNN are presented below:

Rule extracted using DEWNN+DENFIS:

Auto MPG DATASET

(1) If Cylinders is GaussianMF(0.41 0.22) and Displacement is GaussianMF(0.44 0.09) and Horsepower is GaussianMF(0.47 0.16) and Weight is GaussianMF(0.45 0.15) and Acceleration is GaussianMF(0.53 0.41) and Model year is GaussianMF(0.50 0.65) and Origin is GaussianMF(0.53 0.50) then Miles per Gallon

= 1.71 - 0.16 * Cylinders + 0.12 * Displacement - 0.62 * Horsepower - 0.71 * Weight - 0.03 * Acceleration + 0.36 * Model year + 0.04 * Origin

(2) If Cylinders is GaussianMF(0.58 0.61) and Displacement is GaussianMF(0.52 0.51) and Horsepower is GaussianMF(0.60 0.31) and Weight is GaussianMF(0.56 0.49) and Acceleration is GaussianMF(0.57 0.55) and Model year is GaussianMF(0.55 0.48) and Origin is GaussianMF(0.43 0.04) then Miles per Gallon = 1.71 - 0.11 * Cylinders + 0.12 * Displacement - 0.34 * Horsepower - 0.53 * Weight - 0.19 * Acceleration + 0.28 * Model year - 0.05 * Origin

(3) If Cylinders is GaussianMF(0.47 0.96) and Displacement is GaussianMF(0.35 0.97) and Horsepower is GaussianMF(0.25 0.96) and Weight is GaussianMF(0.48 0.79) and Acceleration is GaussianMF(0.44 0.06) and Model year is GaussianMF(0.38 0.03) and Origin is GaussianMF(0.46 0.04) then Miles per Gallon = 0.88 + 0.54 * Cylinders + 0.18 * Displacement - 0.20 * Horsepower - 0.33 * Weight - 0.01 * Acceleration + 0.14 * Model year - 4.92 * Origin

Bodyfat DATASET

(1) If Density is GaussianMF(0.50 0.34) and Age is GaussianMF(0.50 0.71) and Weight is GaussianMF(0.50 0.31) and Height is GaussianMF(0.50 0.73) and Neck is GaussianMF(0.50 0.48) and Chest is GaussianMF(0.50 0.48) and Abdomen is GaussianMF(0.50 0.40) and Hip is GaussianMF(0.50 0.27) and Thigh is GaussianMF(0.50 0.32) and Knee is GaussianMF(0.50 0.33) and Ankle is GaussianMF(0.50 0.35) and Biceps is GaussianMF(0.50 0.53) and Forearm is GaussianMF(0.50 0.67) and Wrist is GaussianMF(0.50 0.58) then Body Fat = 1.37 - 0.85 * Density + 0.20 * Age + 0.64 * Weight + 0.29 * Height + 0.17 * Neck - 0.75 * Chest + 0.61 * Abdomen + 0.24 * Hip - 0.18 * Thigh + 0.08 * Knee - 0.07 * Ankle + 0.27 * Biceps + 0.04 * Forearm - 0.13 * Wrist

(2) If Density is GaussianMF(0.50 0.95) and Age is GaussianMF(0.50 0.32) and Weight is GaussianMF(0.50 0.05) and Height is GaussianMF(0.50 0.77) and Neck is GaussianMF(0.50 0.17) and Chest is GaussianMF(0.50 0.05) and Abdomen is GaussianMF(0.50 0.05) and Hip is GaussianMF(0.50 0.05) and Thigh is GaussianMF(0.50 0.05) and Knee is GaussianMF(0.50 0.07) and Ankle is GaussianMF(0.50 0.11) and Biceps is GaussianMF(0.50 0.16) and Forearm is

GaussianMF(0.50 0.28) and Wrist is GaussianMF(0.50 0.16) and then Body Fat = 1.40 - 0.83 * Density + 0.20 * Age + 0.74 * Weight + 0.27 * Height + 0.17 * Neck - 0.84 * Chest + 0.69 * Abdomen + 0.08 * Hip - 0.16 * Thigh - 0.00 * Knee - 0.09 * Ankle + 0.26 * Biceps + 0.05 * Forearm - 0.09 * Wrist

(3) If Density is GaussianMF(0.50 0.25) and Age is GaussianMF(0.50 0.42) and Weight is GaussianMF(0.50 0.95) and Height is GaussianMF(0.50 0.85) and Neck is GaussianMF(0.50 0.95) and Chest is GaussianMF(0.50 0.95) and Abdomen is GaussianMF(0.50 0.95) and Hip is GaussianMF(0.50 0.95) and Thigh is GaussianMF(0.50 0.95) and Knee is GaussianMF(0.50 0.95) and Ankle is GaussianMF(0.50 0.69) and Biceps is GaussianMF(0.50 0.95) and Forearm is GaussianMF(0.50 0.57) and Wrist is GaussianMF(0.50 0.95) and then Body Fat = 1.60 - 0.48 * Density + 0.04 * Age + 0.82 * Weight + 0.94 * Height - 0.48 * Neck - 0.39 * Chest + 0.33 * Abdomen + 0.12 * Hip - 0.53 * Thigh - 0.31 * Knee + 0.05 * Ankle - 0.58 * Biceps + 0.15 * Forearm + 0.03 * Wrist

Boston Housing DATASET

(1) If CRIM is GaussianMF(0.50 0.05) and ZN is GaussianMF(0.50 0.05) and INDUS is GaussianMF(0.51 0.35) and CHAS is GaussianMF(0.56 0.09) and NOX is GaussianMF(0.53 0.41) and RM is GaussianMF(0.50 0.36) and AGE is GaussianMF(0.50 0.71) and DIS is GaussianMF(0.50 0.15) and RAD is GaussianMF(0.47 0.24) and TAX is GaussianMF(0.48 0.40) and PTRATIO is GaussianMF(0.52 0.68) and B:1000 is GaussianMF(0.45 0.95) and LSTAT is GaussianMF(0.51 0.38) then MEDV = 1.62 - 0.12 * CRIM - 0.05 * ZN - 0.23 * INDUS + 0.16 * CHAS - 0.35 * NOX + 0.69 * RM - 0.35 * AGE - 0.73 * DIS - 0.04 * RAD + 0.10 * TAX - 0.20 * PTRATIO + 0.19 * B:1000 - 0.02 * LSTAT

(2) If CRIM is GaussianMF(0.48 0.44) and ZN is GaussianMF(0.50 0.05) and INDUS is GaussianMF(0.49 0.63) and CHAS is GaussianMF(0.50 0.08) and NOX is GaussianMF(0.48 0.62) and RM is GaussianMF(0.50 0.34) and AGE is GaussianMF(0.50 0.96) and DIS is GaussianMF(0.50 0.07) and RAD is GaussianMF(0.33 0.96) and TAX is GaussianMF(0.44 0.88) and PTRATIO is GaussianMF(0.44 0.79) and B:1000 is GaussianMF(0.55 0.95) and LSTAT

is GaussianMF(0.48 0.77) then MEDV = 1.57+ 0.06 * CRIM + 8.78 * ZN + 0.65 * INDUS + 0.26 * CHAS - 0.27 * NOX+ 0.38 * RM - 0.32 * AGE - 0.20 * DIS+ 0.19 * RAD - 0.41 * TAX - 0.99 * PTRATIO + 0.10 * B:1000 - 0.03 * LSTAT

(3) If CRIM is GaussianMF(0.50 0.05) and ZN is GaussianMF(0.54 0.95) and INDUS is GaussianMF(0.50 0.08) and CHAS is GaussianMF(0.53 0.08) and NOX is GaussianMF(0.50 0.07) and RM is GaussianMF(0.50 0.65) and AGE is GaussianMF(0.51 0.15) and DIS is GaussianMF(0.51 0.58) and RAD is GaussianMF(0.50 0.13) and TAX is GaussianMF(0.50 0.42) and PTRATIO is GaussianMF(0.50 0.47) and B:1000 is GaussianMF(0.50 0.93) and LSTAT is GaussianMF(0.50 0.11) then MEDV = 1.57 + 5.56 * CRIM - 0.06 * ZN - 0.18 * INDUS + 0.34 * CHAS - 0.32 * NOX + 0.54 * RM - 0.31 * AGE - 0.58 * DIS - 0.02 * RAD - 0.02 * TAX - 0.25 * PTRATIO + 0.04 * B:1000 - 0.22 * LSTAT

(4) If CRIM is GaussianMF(0.50 0.10) and ZN is GaussianMF(0.48 0.04) and INDUS is GaussianMF(0.47 0.64) and CHAS is GaussianMF(0.46 0.90) and NOX is GaussianMF(0.48 0.50) and RM is GaussianMF(0.50 0.57) and AGE is GaussianMF(0.47 0.93) and DIS is GaussianMF(0.48 0.06) and RAD is GaussianMF(0.39 0.96) and TAX is GaussianMF(0.41 0.88) and PTRATIO is GaussianMF(0.50 0.78) and B:1000 is GaussianMF(0.49 0.91) and LSTAT is GaussianMF(0.50 0.09) then MEDV = 0.14 + 0.05 * CRIM + 13.10 * ZN + 1.09 * INDUS + 0.23 * CHAS - 0.29 * NOX + 0.36 * RM - 0.35 * AGE - 0.23 * DIS - 0.11 * RAD + 0.49 * TAX - 0.38 * PTRATIO + 0.13 * B:1000 - 0.07 * LSTAT

Forest fires DATASET

(1) If X-axis is GaussianMF(0.50 0.17) and Y-axis is GaussianMF(0.50 0.87) and Month is GaussianMF(0.48 0.35) and Day is GaussianMF(0.50 0.76) and FFMC is GaussianMF(0.50 0.05) andDMC is GaussianMF(0.50 0.15) and DC is GaussianMF(0.49 0.09) and ISI is GaussianMF(0.50 0.32) and Temperature is GaussianMF(0.51 0.22) and RH is GaussianMF(0.50 0.43) and Wind is GaussianMF(0.50 0.05) and Rain is GaussianMF(0.50 0.05) then AREA =

1.11 - 0.08 * X-axis + 0.15 * Y-axis - 0.07 * Month - 0.00 * Day + 0.37 * FFMC - 0.25 * DMC + 0.14 * DC - 0.14 * ISI + 0.03 * Temperature - 0.02 * RH + 9.91 * Wind - 0.14 * Rain

(2) If X-axis is GaussianMF(0.51 0.18) and Y-axis is GaussianMF(0.50 0.71) and Month is GaussianMF(0.39 0.96) and Day is GaussianMF(0.50 0.90) and FFMC is GaussianMF(0.50 0.34) and DMC is GaussianMF(0.50 0.84) and DC is GaussianMF(0.48 0.40) and ISI is GaussianMF(0.49 0.63) and Temperature is GaussianMF(0.49 0.39) and RH is GaussianMF(0.48 0.37) and Wind is GaussianMF(0.43 0.04) and Rain is GaussianMF(0.51 0.05) then AREA = 1.12 - 0.12 * X-axis + 0.28 * Y-axis - 0.03 * Month + 0.16 * Day + 0.26 * FFMC - 0.25 * DMC + 0.09 * DC - 0.02 * ISI + 0.04 * Temperature + 0.02 * RH + 3.85 * Wind - 0.20 * Rain

(3) If X-axis is GaussianMF(0.52 0.56) and Y-axis is GaussianMF(0.50 0.21) and Month is GaussianMF(0.54 0.80) and Day is GaussianMF(0.51 0.90) and FFMC is GaussianMF(0.51 0.15) and DMC is GaussianMF(0.52 0.12) and DC is GaussianMF(0.52 0.43) and ISI is GaussianMF(0.50 0.22) and Temperature is GaussianMF(0.55 0.92) and RH is GaussianMF(0.51 0.38) and Wind is GaussianMF(0.50 0.07) and Rain is GaussianMF(0.50 0.05) then AREA = 1.47 - 0.01 * X-axis + 0.24 * Y-axis - 0.12 * Month + 0.01 * Day + 0.35 * FFMC - 0.40 * DMC + 0.18 * DC + 0.03 * ISI + 0.24 * Temperature + 0.16 * RH - 3.68 * Wind + 0.26 * Rain

(4) If X-axis is GaussianMF(0.51 0.43) and Y-axis is GaussianMF(0.50 0.46) and Month is GaussianMF(0.51 0.05) and Day is GaussianMF(0.50 0.92) and FFMC is GaussianMF(0.50 0.61) and DMC is GaussianMF(0.51 0.50) and DC is GaussianMF(0.50 0.52) and ISI is GaussianMF(0.50 0.63) and Temperature is GaussianMF(0.52 0.40) and RH is GaussianMF(0.54 0.09) and Wind is GaussianMF(0.52 0.05) and Rain is GaussianMF(0.50 0.05) then AREA = 1.47 - 0.07 * X-axis + 0.30 * Y-axis - 0.07 * Month + 0.18 * Day + 0.30 * FFMC - 0.36 * DMC + 0.08 * DC - 0.14 * ISI + 0.00 * Temperature - 0.00 * RH - 0.85 * Wind - 0.09 * Rain

(5) If X-axis is GaussianMF(0.50 0.44) and Y-axis is GaussianMF(0.50 0.04) and

Month is GaussianMF(0.49 0.04) and Day is GaussianMF(0.33 0.04) and FFMC is GaussianMF(0.50 0.05) and DMC is GaussianMF(0.49 0.22) and DC is GaussianMF(0.49 0.04) and ISI is GaussianMF(0.50 0.13) and Temperature is GaussianMF(0.48 0.96) and RH is GaussianMF(0.46 0.04) and Wind is GaussianMF(0.50 0.05) and Rain is GaussianMF(0.50 0.05) then AREA = 0.75 - 0.15 * X-axis + 3.06 * Y-axis - 0.36 * Month - 0.02 * Day - 1.97 * FFMC - 0.63 * DMC + 1.29 * DC - 1.65 * ISI - 0.25 * Temperature + 0.24 * RH + 28.32 * Wind + 1.24 * Rain

(6) If X-axis is GaussianMF(0.49 0.43) and Y-axis is GaussianMF(0.50 0.62) and Month is GaussianMF(0.48 0.35) and Day is GaussianMF(0.50 0.95) and FFMC is GaussianMF(0.49 0.61) and DMC is GaussianMF(0.50 0.76) and DC is GaussianMF(0.49 0.66) and ISI is GaussianMF(0.49 0.78) and Temperature is GaussianMF(0.49 0.56) and RH is GaussianMF(0.50 0.47) and Wind is GaussianMF(0.30 0.96) and Rain is GaussianMF(0.50 0.05) then AREA = 2.10 - 0.58 * X-axis + 2.26 * Y-axis + 0.11 * Month - 2.24 * Day + 0.93 * FFMC - 0.84 * DMC + 0.57 * DC + 0.17 * ISI + 0.36 * Temperature - 0.14 * RH - 0.38 * Wind - 1.20 * Rain

Pollution DATASET

(1) If PREC is GaussianMF(0.50 0.83) and JANT is GaussianMF(0.50 0.59) and JULT is GaussianMF(0.50 0.78) and OVR65 is GaussianMF(0.50 0.35) and POPN is GaussianMF(0.50 0.89) and EDUC is GaussianMF(0.50 0.27) and HOUS is GaussianMF(0.50 0.05) and DENS is GaussianMF(0.50 0.25) and NONW is GaussianMF(0.50 0.95) and WWDRK is GaussianMF(0.50 0.27) and POOR is GaussianMF(0.50 0.91) and HC is GaussianMF(0.50 0.13) and NOX is GaussianMF(0.50 0.21) and SO is GaussianMF(0.50 0.28) and HUMID is GaussianMF(0.50 0.46) then MORT = 1.20 + 0.55 * PREC + 0.23 * JANT - 0.37 * JULT - 0.04 * OVR65 - 0.19 * POPN - 0.05 * EDUC - 0.13 * HOUS + 0.12 * DENS + 0.76 * NONW - 0.11 * WWDRK - 0.36 * POOR - 1.15 * HC + 0.87 * NOX + 0.18 * SO + 0.45 * HUMID

(2) If PREC is GaussianMF(0.50 0.05) and JANT is GaussianMF(0.50 0.76) and JULT is GaussianMF(0.50 0.35) and OVR65 is GaussianMF(0.50 0.29) and

POPN is GaussianMF(0.50 0.35) and EDUC is GaussianMF(0.50 0.89) and HOUS is GaussianMF(0.50 0.89) and DENS is GaussianMF(0.50 0.20) and NONW is GaussianMF(0.50 0.17) and WWDRK is GaussianMF(0.50 0.60) and POOR is GaussianMF(0.50 0.28) and HC is GaussianMF(0.50 0.46) and NOX is GaussianMF(0.50 0.39) and SO is GaussianMF(0.50 0.11) and HUMID is GaussianMF(0.50 0.64) then MORT = 1.72 + 0.18 * PREC + 0.34 * JANT - 0.38 * JULT + 0.19 * OVR65 + 0.10 * POPN - 0.01 * EDUC - 0.59 * HOUS - 0.04 * DENS + 0.91 * NONW - 0.15 * WWDRK - 0.76 * POOR - 0.97 * HC + 0.79 * NOX + 0.33 * SO - 0.17 * HUMID

(3) If PREC is GaussianMF(0.50 0.68) and JANT is GaussianMF(0.50 0.24) and JULT is GaussianMF(0.50 0.35) and OVR65 is GaussianMF(0.50 0.95) and POPN is GaussianMF(0.50 0.57) and EDUC is GaussianMF(0.50 0.57) and HOUS is GaussianMF(0.50 0.54) and DENS is GaussianMF(0.50 0.31) and NONW is GaussianMF(0.50 0.05) and WWDRK is GaussianMF(0.50 0.34) and POOR is GaussianMF(0.50 0.28) and HC is GaussianMF(0.50 0.06) and NOX is GaussianMF(0.50 0.06) and SO is GaussianMF(0.50 0.07) and HUMID is GaussianMF(0.50 0.51) then MORT = 1.87 + 0.46 * PREC - 0.14 * JANT - 0.41 * JULT + 0.03 * OVR65 - 0.13 * POPN - 0.01 * EDUC - 0.50 * HOUS + 0.11 * DENS + 0.91 * NONW - 0.24 * WWDRK - 0.59 * POOR - 0.43 * HC + 0.72 * NOX + 0.17 * SO - 0.22 * HUMID

(4) If PREC is GaussianMF(0.50 0.46) and JANT is GaussianMF(0.50 0.28) and JULT is GaussianMF(0.50 0.61) and OVR65 is GaussianMF(0.50 0.49) and POPN is GaussianMF(0.50 0.49) and EDUC is GaussianMF(0.50 0.50) and HOUS is GaussianMF(0.50 0.68) and DENS is GaussianMF(0.50 0.73) and NONW is GaussianMF(0.50 0.42) and WWDRK is GaussianMF(0.50 0.35) and POOR is GaussianMF(0.50 0.10) and HC is GaussianMF(0.50 0.30) and NOX is GaussianMF(0.50 0.38) and SO is GaussianMF(0.50 0.95) and HUMID is GaussianMF(0.50 0.57) then MORT = 1.08 + 0.58 * PREC + 0.12 * JANT - 0.26 * JULT + 0.10 * OVR65 + 0.03 * POPN + 0.02 * EDUC - 0.30 * HOUS + 0.06 * DENS + 0.65 * NONW - 0.10 * WWDRK - 0.36 * POOR - 0.27 * HC - 0.13 * NOX + 0.45 * SO + 0.35 * HUMID

(5) If PREC is GaussianMF(0.50 0.95) and JANT is GaussianMF(0.50 0.95) and

JULT is GaussianMF(0.50 0.87) and OVR65 is GaussianMF(0.50 0.69) and POPN is GaussianMF(0.50 0.14) and EDUC is GaussianMF(0.50 0.69) and HOUS is GaussianMF(0.50 0.87) and DENS is GaussianMF(0.50 0.48) and NONW is GaussianMF(0.50 0.35) and WWDRK is GaussianMF(0.50 0.45) and POOR is GaussianMF(0.50 0.74) and HC is GaussianMF(0.50 0.05) and NOX is GaussianMF(0.50 0.05) and SO is GaussianMF(0.50 0.05) and HUMID is GaussianMF(0.50 0.62) then MORT = 2.31 + 0.39 * PREC + 1.32 * JANT - 0.78 * JULT + 0.13 * OVR65 + 0.05 * POPN - 0.06 * EDUC - 1.50 * HOUS + 0.42 * DENS + 0.78 * NONW - 0.03 * WWDRK - 1.25 * POOR - 1.62 * HC + 1.23 * NOX - 0.18 * SO - 0.51 * HUMID

Rule extracted using DEWNN+CART:
Auto MPG DATASET

(1) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $\leq$ 0.458333 and HORSEPOWER $\leq$ 0.0923915) mean = 0.538541

(2) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $\leq$ 0.458333 and HORSEPOWER $>$ 0.0923915 and HORSEPOWER $\leq$ 0.138587) mean = 0.436451

(3) If (CYLINDERS $\leq$ 0.5 and HORSEPOWER $\leq$ 0.138587 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.708333 and ACCELERATION $\leq$ 0.375) mean = 0.454463

(4) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.708333 and ACCELERATION $>$ 0.375 and ACCELERATION $\leq$ 0.627976 and HORSEPOWER $\leq$ 0.084239) mean = 0.628446

(5) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.708333 and ACCELERATION $>$ 0.375 and ACCELERATION $\leq$ 0.627976 and HORSEPOWER $>$ 0.084239 and HORSEPOWER $\leq$ 0.138587) mean = 0.576495

(6) If (CYLINDERS $\leq$ 0.5 and HORSEPOWER $\leq$ 0.138587 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.708333 and ACCELERATION $>$ 0.627976) mean = 0.655603

(7) If (CYLINDERS ≤ 0.5 and HORSEPOWER > 0.138587 and HORSEPOWER ≤ 0.220109 and MODEL YEAR ≤ 0.25) mean $= 0.344237$

(8) If (CYLINDERS ≤ 0.5 and HORSEPOWER > 0.138587 and HORSEPOWER ≤ 0.220109 and MODEL YEAR > 0.25 and MODEL YEAR ≤ 0.458333) mean $= 0.3883$

(9) If (CYLINDERS ≤ 0.5 and MODEL YEAR ≤ 0.458333 and HORSEPOWER > 0.220109 and WEIGHT ≤ 0.340375 and ACCELERATION ≤ 0.476191) mean $= 0.318815$

(10) If (CYLINDERS ≤ 0.5 and MODEL YEAR ≤ 0.458333 and HORSEPOWER > 0.220109 and WEIGHT ≤ 0.340375 and ACCELERATION > 0.476191) mean $= 0.361013$

(11) If (CYLINDERS ≤ 0.5 and MODEL YEAR ≤ 0.458333 and HORSEPOWER > 0.220109 and WEIGHT > 0.340375) mean $= 0.293827$

(12) If (CYLINDERS ≤ 0.5 and HORSEPOWER > 0.138587 and MODEL YEAR > 0.458333 and MODEL YEAR ≤ 0.708333 and WEIGHT ≤ 0.232918 and ACCELERATION ≤ 0.458333) mean $= 0.447117$

(13) If (CYLINDERS ≤ 0.5 and HORSEPOWER > 0.138587 and MODEL YEAR > 0.458333 and MODEL YEAR ≤ 0.708333 and WEIGHT ≤ 0.232918 and ACCELERATION > 0.458333 and ACCELERATION ≤ 0.592262) mean $=$ 0.486585

(14) If (CYLINDERS ≤ 0.5 and HORSEPOWER > 0.138587 and MODEL YEAR > 0.458333 and MODEL YEAR ≤ 0.708333 and WEIGHT ≤ 0.232918 and ACCELERATION > 0.592262) mean $= 0.559671$

(15) If (CYLINDERS ≤ 0.5 and MODEL YEAR > 0.458333 and MODEL YEAR ≤ 0.708333 and HORSEPOWER > 0.138587 and HORSEPOWER ≤ 0.285326 and WEIGHT > 0.232918 and WEIGHT ≤ 0.337964) mean $= 0.381937$

(16) If (CYLINDERS ≤ 0.5 and MODEL YEAR > 0.458333 and MODEL YEAR ≤ 0.708333 and HORSEPOWER > 0.138587 and HORSEPOWER ≤ 0.285326 and WEIGHT > 0.337964) mean $= 0.421391$

(17) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.708333 and WEIGHT $>$ 0.232918 and HORSEPOWER $>$ 0.285326) mean = 0.351983

(18) If (CYLINDERS $\leq$ 0.5 and ACCELERATION $\leq$ 0.425596 and MODEL YEAR $>$ 0.708333 and MODEL YEAR $\leq$ 0.958333 and DISPLACEMENT $\leq$ 0.116279) mean = 0.578989

(19) If (CYLINDERS $\leq$ 0.5 and ACCELERATION $\leq$ 0.425596 and MODEL YEAR $>$ 0.708333 and MODEL YEAR $\leq$ 0.958333 and DISPLACEMENT $>$ 0.116279 and DISPLACEMENT $\leq$ 0.135659) mean = 0.507946

(20) If (CYLINDERS $\leq$ 0.5 and DISPLACEMENT $\leq$ 0.135659 and ACCELERATION $\leq$ 0.425596 and MODEL YEAR $>$ 0.958333) mean = 0.648922

(21) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $>$ 0.708333 and DISPLACEMENT $\leq$ 0.135659 and ACCELERATION $>$ 0.425596 and HORSEPOWER $\leq$ 0.054348) mean = 0.512448

(22) If (CYLINDERS $\leq$ 0.5 and MODEL YEAR $>$ 0.708333 and DISPLACEMENT $\leq$ 0.135659 and ACCELERATION $>$ 0.425596 and HORSEPOWER $>$ 0.054348 and WEIGHT $\leq$ 0.154097) mean = 0.667176

(23) If (CYLINDERS $\leq$ 0.5 and DISPLACEMENT $\leq$ 0.135659 and ACCELERATION $>$ 0.425596 and HORSEPOWER $>$ 0.054348 and WEIGHT $>$ 0.154097 and MODEL YEAR $>$ 0.708333 and MODEL YEAR $\leq$ 0.958333) mean = 0.612538

(24) If (CYLINDERS $\leq$ 0.5 and DISPLACEMENT $\leq$ 0.135659 and ACCELERATION $>$ 0.425596 and HORSEPOWER $>$ 0.054348 and WEIGHT $>$ 0.154097 and MODEL YEAR $>$ 0.958333) mean = 0.648929

(25) If (CYLINDERS $\leq$ 0.5 and DISPLACEMENT $>$ 0.135659 and MODEL YEAR $>$ 0.708333 and MODEL YEAR $\leq$ 0.875)

mean = 0.487934

(26) If (CYLINDERS $\leq$ 0.5 and DISPLACEMENT $>$ 0.135659 and MODEL YEAR $>$ 0.875) mean = 0.547744

(27) If (CYLINDERS > 0.5 and MODEL YEAR ≤ 0.25 and WEIGHT ≤ 0.427842) mean $= 0.241484$

(28) If (CYLINDERS > 0.5 and MODEL YEAR ≤ 0.25 and WEIGHT > 0.427842 and WEIGHT ≤ 0.574) mean $= 0.188237$

(29) If (CYLINDERS > 0.5 and CYLINDERS ≤ 0.8 and WEIGHT ≤ 0.452935 and MODEL YEAR > 0.25 and MODEL YEAR ≤ 0.541667) mean $= 0.271063$

(30) If (CYLINDERS > 0.5 and CYLINDERS ≤ 0.8 and WEIGHT ≤ 0.452935 and MODEL YEAR > 0.541667 and MODEL YEAR ≤ 0.708333) mean $= 0.308952$

(31) If (MODEL YEAR > 0.25 and MODEL YEAR ≤ 0.708333 and CYLINDERS > 0.5 and CYLINDERS ≤ 0.8 and WEIGHT > 0.452935 and WEIGHT ≤ 0.574) mean $= 0.250273$

(32) If (WEIGHT ≤ 0.574 and MODEL YEAR > 0.25 and MODEL YEAR ≤ 0.708333 and CYLINDERS > 0.8) mean $= 0.206666$

(33) If (CYLINDERS > 0.5 and MODEL YEAR > 0.708333 and WEIGHT ≤ 0.43649) mean $= 0.396744$

(34) If (CYLINDERS > 0.5 and MODEL YEAR > 0.708333 and WEIGHT > 0.43649 and WEIGHT ≤ 0.574) mean $= 0.316756$

(35) If (CYLINDERS > 0.5 and WEIGHT > 0.574 and WEIGHT ≤ 0.665012 and HORSEPOWER ≤ 0.375) mean $= 0.211639$

(36) If (CYLINDERS > 0.5 and WEIGHT > 0.574 and WEIGHT ≤ 0.665012 and HORSEPOWER > 0.375) mean $= 0.178843$

(37) If (CYLINDERS > 0.5 and WEIGHT > 0.665012 and WEIGHT ≤ 0.79898 and HORSEPOWER ≤ 0.673913) mean $= 0.146128$

(38) If (CYLINDERS > 0.5 and WEIGHT > 0.665012 and WEIGHT ≤ 0.79898 and HORSEPOWER > 0.673913) mean $= 0.171466$

(39) If (CYLINDERS > 0.5 and WEIGHT > 0.79898 and HORSEPOWER ≤ 0.853261) mean $= 0.121835$

(40) If (CYLINDERS $>$ 0.5 and WEIGHT $>$ 0.79898 and HORSEPOWER $>$ 0.853261) mean = 0.159515

Bodyfat DATASET

(1) If (CHEST $\leq$ 0.559754 and DENSITY $\leq$ 0.424934) mean = 0.510571

(2) If (CHEST $\leq$ 0.559754 and DENSITY $>$ 0.424934 and DENSITY $\leq$ 0.5518) mean = 0.420974

(3) If (CHEST $>$ 0.559754 and DENSITY $\leq$ 0.40079) mean = 0.655681

(4) If (CHEST $>$ 0.559754 and DENSITY $>$ 0.40079 and DENSITY $\leq$ 0.5518) mean = 0.544287

(5) If (ABDOMEN $\leq$ 0.174714 and DENSITY $>$ 0.5518 and DENSITY $\leq$ 0.835382) mean = 0.192939

(6) If (ABDOMEN $\leq$ 0.174714 and DENSITY $>$ 0.835382) mean = 0.084981

(7) If (ABDOMEN $>$ 0.174714 and DENSITY $>$ 0.5518 and DENSITY $\leq$ 0.644425) mean = 0.319957

(8) If (ABDOMEN $>$ 0.174714 and DENSITY $>$ 0.644425) mean = 0.245206

Boston Housing DATASET

(1) If (INDUS $\leq$ 0.223424 and PTRATIO $\leq$ 0.335106 and RM $\leq$ 0.45823) mean = 0.507719

(2) If (INDUS $\leq$ 0.223424 and PTRATIO $\leq$ 0.335106 and RM $>$ 0.45823) mean = 0.657551

(3) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $\leq$ 0.533531 and RAD $\leq$ 0.021739) mean = 0.537651

(4) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $\leq$ 0.533531 and RAD $>$ 0.021739 and LSTAT $\leq$ 0.170806) mean = 0.428368

(5) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RAD $>$ 0.021739 and LSTAT $>$ 0.170806 and RM $\leq$ 0.467235) mean = 0.344452

(6) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RAD $>$ 0.021739 and LSTAT $>$ 0.170806 and RM $>$ 0.467235 and RM $\leq$ 0.533531) mean = 0.386581

(7) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and CHAS $\leq$ 0.5 and ZN $\leq$ 0.275 and RM $>$ 0.533531 and RM $\leq$ 0.641406) mean = 0.451287

(8) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and CHAS $\leq$ 0.5 and ZN $\leq$ 0.275 and RM $>$ 0.641406 and RM $\leq$ 0.817111) mean = 0.508499

(9) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and CHAS $\leq$ 0.5 and ZN $\leq$ 0.275 and RM $>$ 0.817111) mean = 0.561884

(10) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $>$ 0.533531 and CHAS $\leq$ 0.5 and ZN $>$ 0.275 and AGE $\leq$ 0.340886 and DIS $\leq$ 0.375497) mean = 0.485593

(11) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $>$ 0.533531 and CHAS $\leq$ 0.5 and ZN $>$ 0.275 and AGE $\leq$ 0.340886 and DIS $>$ 0.375497) mean = 0.580005

(12) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $>$ 0.533531 and CHAS $\leq$ 0.5 and ZN $>$ 0.275 and AGE $>$ 0.340886) mean = 0.516436

(13) If (INDUS $\leq$ 0.223424 and PTRATIO $>$ 0.335106 and RM $>$ 0.533531 and CHAS $>$ 0.5) mean = 0.308789

(14) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and DIS $\leq$ 0.39372 and PTRATIO $\leq$ 0.776596 and LSTAT $\leq$ 0.122792) mean = 0.378099

(15) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and DIS $\leq$ 0.39372 and PTRATIO $\leq$ 0.776596 and LSTAT $>$ 0.122792 and LSTAT $\leq$ 0.239928) mean = 0.33437

(16) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $\leq$ 0.239928 and DIS $\leq$ 0.39372 and PTRATIO $>$ 0.776596) mean = 0.280799

(17) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $\leq$ 0.239928 and DIS $>$ 0.39372) mean = 0.439315

(18) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $\leq$ 0.845745 and CHAS $\leq$ 0.5 and AGE $\leq$ 0.580329) mean = 0.31773

(19) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $\leq$ 0.845745 and CHAS $\leq$ 0.5 and AGE $>$ 0.580329) mean = 0.285139

(20) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $\leq$ 0.845745 and CHAS $>$ 0.5 and DIS $\leq$ 0.229992) mean = 0.277434

(21) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $\leq$ 0.845745 and CHAS $>$ 0.5 and DIS $>$ 0.229992) mean = 0.183654

(22) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $>$ 0.845745 and B $\leq$ 0.953553) mean = 0.19137

(23) If (INDUS $>$ 0.223424 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.239928 and PTRATIO $>$ 0.845745 and B $>$ 0.953553) mean = 0.232888

(24) If (INDUS $>$ 0.743218) mean = 0.150074

Forest fires DATASET

(1) If (TEMPERATURE $\leq$ 0.733119) mean = 0.000152651

(2) If (TEMPERATURE $>$ 0.733119 and TEMPERATURE $\leq$ 0.73955) mean = 0.28478

(3) If (TEMPERATURE $>$ 0.73955) mean = 0.00163878

Pollution DATASET

(1) If (NONW $\leq$ 0.381963) mean = 0.408978

(2) If (NONW $>$ 0.381963) mean = 0.657968

Rules extracted using the proposed approach for regression problems over full features when Morlet function is used in DEWNN are presented below:

Rule extracted using DEWNN+DENFIS:

Auto MPG DATASET

(1) If Cylinders is GaussianMF(0.41 0.22) and Displacement is GaussianMF(0.45 0.09) and Horsepower is GaussianMF(0.48 0.16) and Weight is GaussianMF(0.45 0.15) and Acceleration is GaussianMF(0.52 0.41) and Modle year is GaussianMF(0.48 0.65) and Origin is GaussianMF(0.52 0.50) then Miles per Gallon = 1.72 - 0.20 * CMiles per Gallonlinders + 0.44 * Displacement - 0.55 * Horsepower - 0.96 * Weight + 0.02 * Acceleration + 0.41 * Modle Miles per Gallonear + 0.04 * Origin

(2) If Cylinders is GaussianMF(0.58 0.61) and Displacement is GaussianMF(0.52 0.51) and Horsepower is GaussianMF(0.60 0.30) and Weight is GaussianMF(0.55 0.49) and Acceleration is GaussianMF(0.58 0.55) and Modle year is GaussianMF(0.56 0.49) and Origin is GaussianMF(0.32 0.03) then Miles per Gallon = 1.69 - 0.12 * CMiles per Gallonlinders - 0.12 * Displacement - 0.12 * Horsepower - 0.46 * Weight - 0.13 * Acceleration + 0.33 * Modle Miles per Gallonear - 0.14 * Origin

(3) If Cylinders is GaussianMF(0.51 0.95) and Displacement is GaussianMF(0.47 0.96) and Horsepower is GaussianMF(0.43 0.94) and Weight is GaussianMF(0.50 0.79) and Acceleration is GaussianMF(0.49 0.07) and Modle year is GaussianMF(0.39 0.03) and Origin is GaussianMF(0.49 0.05) then Miles per Gallon = 0.95 + 0.31 * CMiles per Gallonlinders - 0.54 * Displacement - 0.06 * Horsepower + 0.31 * Weight - 0.16 * Acceleration + 0.10 * Modle Miles per Gallonear - 1.21 * Origin

Bodyfat DATASET

(1) If Density is GaussianMF(0.49 0.34) and Age is GaussianMF(0.48 0.71) and Weight is GaussianMF(0.49 0.31) and Height is GaussianMF(0.50 0.73) and Neck is GaussianMF(0.50 0.48) and Chest is GaussianMF(0.49 0.48) and Abdomen is GaussianMF(0.49 0.40) and Hip is GaussianMF(0.49 0.27) and Thigh is GaussianMF(0.49 0.32) and Knee is GaussianMF(0.50 0.33) and Ankle is

GaussianMF(0.50 0.35) and Biceps is GaussianMF(0.50 0.54) and Frearm is GaussianMF(0.49 0.67) and Wrist is GaussianMF(0.49 0.58) then Boda fat = 1.34 - 0.53 * DensitBoda fat + 0.25 * Age + 0.01 * Weight + 0.12 * Height - 0.24 * Neck + 0.43 * Chest + 0.17 * Abdomen - 0.04 * Hip + 0.47 * Thigh - 0.18 * Knee - 0.02 * Ankle - 0.03 * Biceps - 0.01 * Frearm + 0.04 * Wrist

(2) If Density is GaussianMF(0.51 0.95) and Age is GaussianMF(0.50 0.32) and Weight is GaussianMF(0.50 0.05) and Height is GaussianMF(0.50 0.77) and Neck is GaussianMF(0.50 0.17) and Chest is GaussianMF(0.50 0.05) and Abdomen is GaussianMF(0.50 0.05) and Hip is GaussianMF(0.50 0.05) and Thigh is GaussianMF(0.50 0.05) and Knee is GaussianMF(0.50 0.07) and Ankle is GaussianMF(0.50 0.11) and Biceps is GaussianMF(0.50 0.16) and Frearm is GaussianMF(0.50 0.28) and Wrist is GaussianMF(0.50 0.16) then Boda fat = 1.49 - 0.55 * DensitBoda fat + 0.23 * Age + 0.08 * Weight - 0.07 * Height - 0.27 * Neck + 0.49 * Chest + 0.14 * Abdomen + 0.30 * Hip + 0.56 * Thigh - 0.40 * Knee + 0.00 * Ankle - 0.11 * Biceps - 0.12 * Frearm + 0.17 * Wrist

(3) If Density is GaussianMF(0.50 0.25) and Age is GaussianMF(0.50 0.41) and Weight is GaussianMF(0.52 0.95) and Height is GaussianMF(0.50 0.85) and Neck is GaussianMF(0.52 0.95) and Chest is GaussianMF(0.50 0.95) and Abdomen is GaussianMF(0.51 0.95) and Hip is GaussianMF(0.52 0.95) and Thigh is GaussianMF(0.51 0.95) and Knee is GaussianMF(0.52 0.95) and Ankle is GaussianMF(0.51 0.69) and Biceps is GaussianMF(0.52 0.95) and Frearm is GaussianMF(0.51 0.57) and Wrist is GaussianMF(0.50 0.95) then Boda fat = 1.80 - 0.37 * DensitBoda fat + 0.06 * Age - 0.57 * Weight - 0.03 * Height - 0.62 * Neck + 0.30 * Chest + 0.60 * Abdomen - 0.18 * Hip - 0.14 * Thigh + 0.15 * Knee - 0.36 * Ankle + 0.10 * Biceps + 0.22 * Frearm + 0.13 * Wrist

Boston Housing DATASET

(1) If CRIM is GaussianMF(0.50 0.05) and ZN is GaussianMF(0.51 0.05) and INDUS is GaussianMF(0.46 0.35) and CHAS is GaussianMF(0.44 0.09) and NOX is GaussianMF(0.57 0.42) and RM is GaussianMF(0.49 0.39) and AGE is GaussianMF(0.46 0.71) and DIS is GaussianMF(0.47 0.14) and RAD is GaussianMF(0.41 0.24) and TAX is GaussianMF(0.39 0.39) and PTRATIO

is GaussianMF(0.52 0.67) and B is GaussianMF(0.48 0.95) and LSTAT is GaussianMF(0.51 0.39) then MEDV = 1.78 - 0.11 * CRIM + 0.09 * ZN - 0.38 * INDUS + 0.12 * CHAS - 0.12 * NOX + 0.05 * RM + 0.25 * AGE + 0.30 * DIS + 0.19 * RAD - 0.34 * TAX - 0.14 * PTRATIO - 0.01 * B - 0.37 * LSTAT

(2) If CRIM is GaussianMF(0.57 0.43) and ZN is GaussianMF(0.50 0.05) and INDUS is GaussianMF(0.58 0.63) and CHAS is GaussianMF(0.44 0.11) and NOX is GaussianMF(0.49 0.59) and RM is GaussianMF(0.49 0.38) and AGE is GaussianMF(0.37 0.96) and DIS is GaussianMF(0.51 0.08) and RAD is GaussianMF(0.67 0.94) and TAX is GaussianMF(0.54 0.88) and PTRATIO is GaussianMF(0.16 0.81) and B is GaussianMF(0.45 0.96) and LSTAT is GaussianMF(0.57 0.76) then MEDV = 1.86 + 0.11 * CRIM + 12.71 * ZN + 0.18 * INDUS + 0.67 * CHAS + 0.40 * NOX + 0.53 * RM - 0.67 * AGE + 0.67 * DIS - 0.11 * RAD + 0.03 * TAX - 1.39 * PTRATIO - 0.13 * B - 0.04 * LSTAT

(3) If CRIM is GaussianMF(0.50 0.05) and ZN is GaussianMF(0.72 0.94) and INDUS is GaussianMF(0.53 0.08) and CHAS is GaussianMF(0.55 0.11) and NOX is GaussianMF(0.51 0.07) and RM is GaussianMF(0.51 0.66) and AGE is GaussianMF(0.48 0.15) and DIS is GaussianMF(0.50 0.58) and RAD is GaussianMF(0.50 0.12) and TAX is GaussianMF(0.51 0.42) and PTRATIO is GaussianMF(0.51 0.47) and B is GaussianMF(0.50 0.93) and LSTAT is GaussianMF(0.50 0.12) then MEDV = 0.68 + 6.18 * CRIM + 0.29 * ZN - 0.30 * INDUS - 0.15 * CHAS - 0.13 * NOX - 0.29 * RM + 0.43 * AGE - 0.08 * DIS + 0.46 * RAD - 0.28 * TAX + 0.47 * PTRATIO + 0.63 * B - 0.22 * LSTAT

(4) If CRIM is GaussianMF(0.50 0.07) and ZN is GaussianMF(0.49 0.05) and INDUS is GaussianMF(0.45 0.68) and CHAS is GaussianMF(0.57 0.88) and NOX is GaussianMF(0.58 0.46) and RM is GaussianMF(0.52 0.78) and AGE is GaussianMF(0.49 0.94) and DIS is GaussianMF(0.49 0.12) and RAD is GaussianMF(0.57 0.21) and TAX is GaussianMF(0.48 0.42) and PTRATIO is GaussianMF(0.40 0.24) and B is GaussianMF(0.56 0.93) and LSTAT is GaussianMF(0.58 0.06) then MEDV = 1.81 + 2.62 * CRIM - 0.22 * ZN - 0.50 * INDUS - 0.08 * CHAS - 0.47 * NOX - 0.27 * RM + 0.54 * AGE + 0.46 *

DIS + 1.02 * RAD - 0.60 * TAX - 0.18 * PTRATIO - 0.07 * B - 0.31 * LSTAT

Forest fires DATASET

(1) If X-AXIS is GaussianMF(0.50 0.18) and Y-AXIS is GaussianMF(0.52 0.87) and MONTH is GaussianMF(0.46 0.34) and DAAREA is GaussianMF(0.52 0.76) and FFMC is GaussianMF(0.49 0.05) and DMC is GaussianMF(0.47 0.15) and DC is GaussianMF(0.49 0.09) and ISI is GaussianMF(0.49 0.32) and Temperature is GaussianMF(0.52 0.22) and RH is GaussianMF(0.50 0.43) and Wind is GaussianMF(0.50 0.05) and Rain is GaussianMF(0.47 0.04) then AREA = 1.09 - 0.11 * X-AXIS - 0.15 * AREA-AXIS - 0.04 * MONTH + 0.06 * DAAREA - 0.04 * FFMC - 0.04 * DMC - 0.15 * DC + 0.35 * ISI + 0.07 * Temperature + 0.35 * RH + 6.60 * Wind - 0.02 * Rain

(2) If X-AXIS is GaussianMF(0.48 0.17) and Y-AXIS is GaussianMF(0.50 0.71) and MONTH is GaussianMF(0.32 0.97) and DAAREA is GaussianMF(0.50 0.90) and FFMC is GaussianMF(0.50 0.34) and DMC is GaussianMF(0.49 0.84) and DC is GaussianMF(0.48 0.40) and ISI is GaussianMF(0.50 0.63) and Temperature is GaussianMF(0.50 0.40) and RH is GaussianMF(0.51 0.38) and Wind is GaussianMF(0.37 0.04) and Rain is GaussianMF(0.57 0.05) then AREA = 1.48 + 0.01 * X-AXIS - 0.20 * AREA-AXIS - 0.01 * MONTH - 0.32 * DAAREA - 0.06 * FFMC + 0.06 * DMC + 0.00 * DC + 0.24 * ISI + 0.05 * Temperature + 0.28 * RH + 5.16 * Wind - 0.31 * Rain

(3) If X-AXIS is GaussianMF(0.50 0.56) and Y-AXIS is GaussianMF(0.51 0.21) and MONTH is GaussianMF(0.61 0.80) and DAAREA is GaussianMF(0.61 0.89) and FFMC is GaussianMF(0.50 0.15) and DMC is GaussianMF(0.50 0.12) and DC is GaussianMF(0.53 0.43) and ISI is GaussianMF(0.50 0.22) and Temperature is GaussianMF(0.51 0.92) and RH is GaussianMF(0.52 0.37) and Wind is GaussianMF(0.50 0.07) and Rain is GaussianMF(0.50 0.05) then AREA = 1.22 - 0.09 * X-AXIS - 0.11 * AREA-AXIS - 0.00 * MONTH + 0.08 * DAAREA + 0.18 * FFMC - 0.05 * DMC - 0.06 * DC + 0.19 * ISI - 0.06 * Temperature + 0.25 * RH + 4.34 * Wind - 0.01 * Rain

(4) If X-AXIS is GaussianMF(0.50 0.44) and Y-AXIS is GaussianMF(0.50 0.46) and MONTH is GaussianMF(0.41 0.04) and DAAREA is GaussianMF(0.50

0.92) and FFMC is GaussianMF(0.51 0.61) and DMC is GaussianMF(0.50 0.50) and DC is GaussianMF(0.50 0.52) and ISI is GaussianMF(0.50 0.63) and Temperature is GaussianMF(0.50 0.40) and RH is GaussianMF(0.49 0.09) and Wind is GaussianMF(0.53 0.05) and Rain is GaussianMF(0.45 0.04) then AREA = 1.58 - 0.04 * X-AXIS - 0.20 * AREA-AXIS - 0.00 * MONTH - 0.22 * DAAREA - 0.08 * FFMC + 0.08 * DMC - 0.10 * DC + 0.31 * ISI + 0.09 * Temperature + 0.41 * RH - 0.60 * Wind + 0.15 * Rain

(5) If X-AXIS is GaussianMF(0.49 0.43) and Y-AXIS is GaussianMF(0.49 0.04) and MONTH is GaussianMF(0.50 0.05) and DAAREA is GaussianMF(0.42 0.04) and FFMC is GaussianMF(0.50 0.05) and DMC is GaussianMF(0.50 0.22) and DC is GaussianMF(0.49 0.04) and ISI is GaussianMF(0.49 0.13) and Temperature is GaussianMF(0.49 0.96) and RH is GaussianMF(0.51 0.05) and Wind is GaussianMF(0.50 0.05) and Rain is GaussianMF(0.50 0.05) then AREA = 0.80 - 0.47 * X-AXIS + 0.03 * AREA-AXIS - 0.21 * MONTH - 0.56 * DAAREA - 0.64 * FFMC + 0.13 * DMC + 0.34 * DC + 0.27 * ISI + 0.04 * Temperature + 0.62 * RH + 38.77 * Wind - 3.69 * Rain

(6) If X-AXIS is GaussianMF(0.50 0.44) and Y-AXIS is GaussianMF(0.50 0.62) and MONTH is GaussianMF(0.48 0.35) and DAAREA is GaussianMF(0.50 0.95) and FFMC is GaussianMF(0.50 0.61) and DMC is GaussianMF(0.50 0.76) and DC is GaussianMF(0.49 0.66) and ISI is GaussianMF(0.50 0.78) and Temperature is GaussianMF(0.50 0.56) and RH is GaussianMF(0.50 0.47) and Wind is GaussianMF(0.50 0.95) and Rain is GaussianMF(0.50 0.05) then AREA = 1.82 + 0.12 * X-AXIS - 1.00 * AREA-AXIS - 0.02 * MONTH - 0.28 * DAAREA - 0.00 * FFMC + 0.35 * DMC - 0.35 * DC + 0.45 * ISI + 0.05 * Temperature + 0.56 * RH - 0.16 * Wind - 0.17 * Rain

Pollution DATASET

(1) If PREC is GaussianMF(0.50 0.83) and JANT is GaussianMF(0.50 0.59) and JULT is GaussianMF(0.50 0.78) and OVR65 is GaussianMF(0.50 0.35) and POPN is GaussianMF(0.50 0.89) and EDUC is GaussianMF(0.50 0.27) and HOUS is GaussianMF(0.50 0.05) and DENS is GaussianMF(0.50 0.25) and NONW is GaussianMF(0.50 0.95) and WWDRK is GaussianMF(0.50 0.27)

and POOR is GaussianMF(0.50 0.91) and HC is GaussianMF(0.50 0.13) and NOX is GaussianMF(0.50 0.21) and SO is GaussianMF(0.50 0.28) and HUMID is GaussianMF(0.50 0.46) then MORT = 1.35 + 0.25 * PREC + 0.17 * JANT - 0.19 * JULT - 0.26 * OVR65 - 0.47 * POPN - 0.12 * EDUC + 0.25 * HOUS + 0.03 * DENS + 0.70 * NONW - 0.40 * WWDRK - 0.03 * POOR + 1.58 * HC - 1.13 * NOX + 0.41 * SO + 0.46 * HUMID

(2) If PREC is GaussianMF(0.50 0.05) and JANT is GaussianMF(0.50 0.76) and JULT is GaussianMF(0.50 0.35) and OVR65 is GaussianMF(0.50 0.29) and POPN is GaussianMF(0.50 0.35) and EDUC is GaussianMF(0.50 0.89) and HOUS is GaussianMF(0.50 0.89) and DENS is GaussianMF(0.50 0.20) and NONW is GaussianMF(0.50 0.17) and WWDRK is GaussianMF(0.50 0.60) and POOR is GaussianMF(0.50 0.28) and HC is GaussianMF(0.51 0.46) and NOX is GaussianMF(0.51 0.39) and SO is GaussianMF(0.50 0.11) and HUMID is GaussianMF(0.50 0.64) then MORT = 1.10 + 0.56 * PREC - 0.45 * JANT - 0.02 * JULT - 0.28 * OVR65 - 0.22 * POPN + 0.09 * EDUC + 0.32 * HOUS + 0.13 * DENS + 0.25 * NONW - 0.25 * WWDRK + 0.51 * POOR - 0.04 * HC + 0.04 * NOX + 0.50 * SO + 0.14 * HUMID

(3) If PREC is GaussianMF(0.50 0.68) and JANT is GaussianMF(0.49 0.24) and JULT is GaussianMF(0.50 0.35) and OVR65 is GaussianMF(0.50 0.96) and POPN is GaussianMF(0.50 0.57) and EDUC is GaussianMF(0.50 0.57) and HOUS is GaussianMF(0.50 0.54) and DENS is GaussianMF(0.50 0.31) and NONW is GaussianMF(0.50 0.05) and WWDRK is GaussianMF(0.50 0.34) and POOR is GaussianMF(0.50 0.28) and HC is GaussianMF(0.50 0.06) and NOX is GaussianMF(0.50 0.06) and SO is GaussianMF(0.50 0.07) and HUMID is GaussianMF(0.50 0.51) then MORT = 0.99 + 0.43 * PREC - 0.43 * JANT - 0.02 * JULT - 0.19 * OVR65 - 0.12 * POPN + 0.02 * EDUC + 0.32 * HOUS + 0.23 * DENS + 0.40 * NONW - 0.18 * WWDRK + 0.40 * POOR + 0.53 * HC + 0.02 * NOX + 0.15 * SO + 0.18 * HUMID

(4) If PREC is GaussianMF(0.50 0.46) and JANT is GaussianMF(0.50 0.28) and JULT is GaussianMF(0.49 0.61) and OVR65 is GaussianMF(0.50 0.49) and POPN is GaussianMF(0.50 0.49) and EDUC is GaussianMF(0.49 0.50) and HOUS is GaussianMF(0.50 0.68) and DENS is GaussianMF(0.50 0.73) and

NONW is GaussianMF(0.50 0.42) and WWDRK is GaussianMF(0.50 0.35) and POOR is GaussianMF(0.50 0.10) and HC is GaussianMF(0.49 0.30) and NOX is GaussianMF(0.49 0.38) and SO is GaussianMF(0.48 0.96) and HUMID is GaussianMF(0.50 0.56) then MORT = 0.91 + 0.42 * PREC - 0.05 * JANT + 0.01 * JULT - 0.35 * OVR65 - 0.12 * POPN - 0.26 * EDUC + 0.27 * HOUS + 0.21 * DENS + 0.33 * NONW + 0.00 * WWDRK + 0.35 * POOR - 0.47 * HC + 1.50 * NOX - 0.27 * SO + 0.57 * HUMID

(5) If PREC is GaussianMF(0.50 0.95) and JANT is GaussianMF(0.50 0.95) and JULT is GaussianMF(0.50 0.87) and OVR65 is GaussianMF(0.50 0.69) and POPN is GaussianMF(0.50 0.14) and EDUC is GaussianMF(0.50 0.69) and HOUS is GaussianMF(0.50 0.87) and DENS is GaussianMF(0.50 0.48) and NONW is GaussianMF(0.50 0.35) and WWDRK is GaussianMF(0.50 0.45) and POOR is GaussianMF(0.50 0.74) and HC is GaussianMF(0.50 0.05) and NOX is GaussianMF(0.50 0.05) and SO is GaussianMF(0.50 0.05) and HUMID is GaussianMF(0.50 0.62) then MORT = 1.78 + 0.32 * PREC - 0.22 * JANT - 0.20 * JULT - 0.30 * OVR65 - 0.15 * POPN - 0.18 * EDUC - 0.19 * HOUS + 0.07 * DENS + 0.44 * NONW - 0.09 * WWDRK - 0.11 * POOR + 0.45 * HC + 1.56 * NOX - 0.26 * SO - 0.06 * HUMID

Rule extracted using DEWNN+CART:
Auto MPG DATASET

(1) If (DISPLACEMENT $\leq$ 0.373385 and MODEL YEAR $\leq$ 0.25) mean = 0.400068

(2) If (DISPLACEMENT $\leq$ 0.373385 and MODEL YEAR $>$ 0.25 and MODEL YEAR $\leq$ 0.458333) mean = 0.436418

(3) If (DISPLACEMENT $\leq$ 0.187339 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.541667 and HORSEPOWER $\leq$ 0.222826) mean = 0.485848

(4) If (DISPLACEMENT $\leq$ 0.187339 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.541667 and HORSEPOWER $>$ 0.222826) mean = 0.414214

(5) If (DISPLACEMENT $\leq$ 0.187339 and MODEL YEAR $>$ 0.541667 and MODEL YEAR $\leq$ 0.791667 and HORSEPOWER $\leq$ 0.214674 and ORIGIN $\leq$ 0.75) mean = 0.554886

(6) If (DISPLACEMENT $\leq$ 0.187339 and MODEL YEAR $>$ 0.541667 and MODEL YEAR $\leq$ 0.791667 and HORSEPOWER $\leq$ 0.214674 and ORIGIN $>$ 0.75) mean = 0.514148

(7) If (DISPLACEMENT $\leq$ 0.187339 and MODEL YEAR $>$ 0.541667 and MODEL YEAR $\leq$ 0.791667 and HORSEPOWER $>$ 0.214674) mean = 0.490263

(8) If (MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.791667 and DISPLACEMENT $>$ 0.187339 and DISPLACEMENT $\leq$ 0.236434) mean = 0.444202

(9) If (MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.791667 and DISPLACEMENT $>$ 0.236434 and DISPLACEMENT $\leq$ 0.373385) mean = 0.377861

(10) If (MODEL YEAR $>$ 0.791667 and DISPLACEMENT $\leq$ 0.108527) mean = 0.6046

(11) If (MODEL YEAR $>$ 0.791667 and DISPLACEMENT $>$ 0.108527 and DISPLACEMENT $\leq$ 0.179587) mean = 0.561178

(12) If (DISPLACEMENT $>$ 0.179587 and DISPLACEMENT $\leq$ 0.242894 and MODEL YEAR $>$ 0.791667 and MODEL YEAR $\leq$ 0.958333) mean = 0.483438

(13) If (DISPLACEMENT $>$ 0.179587 and DISPLACEMENT $\leq$ 0.242894 and MODEL YEAR $>$ 0.958333) mean = 0.525602

(14) If (MODEL YEAR $>$ 0.791667 and DISPLACEMENT $>$ 0.242894 and DISPLACEMENT $\leq$ 0.373385) mean = 0.409168

(15) If (DISPLACEMENT $>$ 0.373385 and DISPLACEMENT $\leq$ 0.498708 and MODEL YEAR $\leq$ 0.25) mean = 0.367933

(16) If (MODEL YEAR $>$ 0.25 and DISPLACEMENT $>$ 0.373385 and DISPLACEMENT $\leq$ 0.48062 and WEIGHT $\leq$ 0.517579) mean = 0.308321

(17) If (DISPLACEMENT $>$ 0.373385 and DISPLACEMENT $\leq$ 0.48062 and WEIGHT $>$ 0.517579 and MODEL YEAR $>$ 0.25 and MODEL YEAR $\leq$ 0.791667) mean = 0.265338

(18) If (DISPLACEMENT $>$ 0.373385 and DISPLACEMENT $\leq$ 0.48062 and WEIGHT $>$ 0.517579 and MODEL YEAR $>$ 0.791667) mean = 0.360234

(19) If (MODEL YEAR $>$ 0.25 and DISPLACEMENT $>$ 0.48062 and DISPLACEMENT $\leq$ 0.498708) mean = 0.225413

(20) If (DISPLACEMENT $>$ 0.498708 and MODEL YEAR $\leq$ 0.541667 and HORSEPOWER $\leq$ 0.524457 and WEIGHT $\leq$ 0.44854) mean = 0.077249

(21) If (MODEL YEAR $\leq$ 0.541667 and HORSEPOWER $\leq$ 0.524457 and WEIGHT $>$ 0.44854 and DISPLACEMENT $>$ 0.498708 and DISPLACEMENT $\leq$ 0.614987) mean = 0.174923

(22) If (MODEL YEAR $\leq$ 0.541667 and HORSEPOWER $\leq$ 0.524457 and WEIGHT $>$ 0.44854 and DISPLACEMENT $>$ 0.614987) mean = 0.237584

(23) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.524457 and HORSEPOWER $\leq$ 0.660326 and MODEL YEAR $\leq$ 0.458333) mean = 0.139722

(24) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.524457 and HORSEPOWER $\leq$ 0.660326 and MODEL YEAR $>$ 0.458333 and MODEL YEAR $\leq$ 0.541667) mean = 0.0812607

(25) If (DISPLACEMENT $>$ 0.498708 and MODEL YEAR $>$ 0.541667 and HORSEPOWER $\leq$ 0.478261) mean = 0.112165

(26) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.478261 and HORSEPOWER $\leq$ 0.660326 and MODEL YEAR $>$ 0.541667 and MODEL YEAR $\leq$ 0.708333) mean = 0.0751784

(27) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.478261 and HORSEPOWER $\leq$ 0.660326 and MODEL YEAR $>$ 0.708333) mean = 0.0360015

(28) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.660326 and HORSEPOWER $\leq$ 0.755435 and MODEL YEAR $\leq$ 0.125) mean = 0.120294

(29) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.660326 and HORSEPOWER $\leq$ 0.755435 and MODEL YEAR $>$ 0.125) mean = 0.0261756

(30) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.755435 and MODEL YEAR $\leq$ 0.208333) mean = 0.010766

(31) If (DISPLACEMENT $>$ 0.498708 and HORSEPOWER $>$ 0.755435 and MODEL YEAR $>$ 0.208333) mean = -0.0306088

Bodyfat DATASET

(1) If (DENSITY $\leq$ 0.615891 and WRIST $\leq$ 0.758929 and FOREARM $\leq$ 0.615108 and BICEPS $\leq$ 0.576733 and NECK $\leq$ 0.298508) mean = 0.348155

(2) If (DENSITY $\leq$ 0.615891 and WRIST $\leq$ 0.758929 and FOREARM $\leq$ 0.615108 and BICEPS $\leq$ 0.576733 and NECK $>$ 0.298508) mean = 0.437185

(3) If (DENSITY $\leq$ 0.615891 and WRIST $\leq$ 0.758929 and FOREARM $\leq$ 0.615108 and BICEPS $>$ 0.576733) mean = 0.056782

(4) If (DENSITY $\leq$ 0.615891 and WRIST $\leq$ 0.758929 and FOREARM $>$ 0.615108 and THIGH $\leq$ 0.50374) mean = 0.527513

(5) If (DENSITY $\leq$ 0.615891 and WRIST $\leq$ 0.758929 and FOREARM $>$ 0.615108 and THIGH $>$ 0.50374) mean = 0.401612

(6) If (DENSITY $\leq$ 0.615891 and WRIST $>$ 0.758929) mean = -0.0369583

(7) If (DENSITY $>$ 0.615891 and FOREARM $\leq$ 0.86331 and AGE $\leq$ 0.237288 and ANKLE $\leq$ 0.72973) mean = 0.205922

(8) If (DENSITY $>$ 0.615891 and FOREARM $\leq$ 0.86331 and AGE $\leq$ 0.237288 and ANKLE $>$ 0.72973) mean = -0.116319

(9) If (DENSITY $>$ 0.615891 and FOREARM $\leq$ 0.86331 and AGE $>$ 0.237288) mean = 0.300633

(10) If (DENSITY $>$ 0.615891 and FOREARM $>$ 0.86331) mean = 0.668529

Boston Housing DATASET

(1) If (LSTAT $\leq$ 0.360651 and RM $\leq$ 0.617072 and B $\leq$ 0.451851) mean = 0.0877014

(2) If (LSTAT $\leq$ 0.360651 and RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $\leq$ 0.139663 and DIS $\leq$ 0.685111) mean = 0.493452

(3) If (LSTAT $\leq$ 0.360651 and RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $\leq$ 0.139663 and DIS $>$ 0.685111) mean = 0.346875

(4) If (RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $>$ 0.139663 and INDUS $\leq$ 0.743218 and LSTAT $\leq$ 0.274696 and RAD $\leq$ 0.195652) mean = 0.37843

(5) If (RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $>$ 0.139663 and INDUS $\leq$ 0.743218 and LSTAT $\leq$ 0.274696 and RAD $>$ 0.195652 and DIS $\leq$ 0.589439) mean = 0.449235

(6) If (RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $>$ 0.139663 and INDUS $\leq$ 0.743218 and LSTAT $\leq$ 0.274696 and RAD $>$ 0.195652 and DIS $>$ 0.589439) mean = 0.341977

(7) If (RM $\leq$ 0.617072 and INDUS $>$ 0.139663 and INDUS $\leq$ 0.743218 and LSTAT $>$ 0.274696 and LSTAT $\leq$ 0.360651 and B $>$ 0.451851 and B $\leq$ 0.819242) mean = 0.273943

(8) If (RM $\leq$ 0.617072 and LSTAT $>$ 0.274696 and LSTAT $\leq$ 0.360651 and B $>$ 0.819242 and INDUS $>$ 0.139663 and INDUS $\leq$ 0.348057) mean = 0.329922

(9) If (RM $\leq$ 0.617072 and LSTAT $>$ 0.274696 and LSTAT $\leq$ 0.360651 and B $>$ 0.819242 and INDUS $>$ 0.348057 and INDUS $\leq$ 0.743218) mean = 0.385475

(10) If (LSTAT $\leq$ 0.360651 and RM $\leq$ 0.617072 and B $>$ 0.451851 and INDUS $>$ 0.743218) mean = 0.218428

(11) If (LSTAT $\leq$ 0.360651 and PTRATIO $\leq$ 0.654255 and TAX $\leq$ 0.337786 and RM $>$ 0.617072 and RM $\leq$ 0.67053) mean = 0.565809

(12) If (LSTAT $\leq$ 0.360651 and PTRATIO $\leq$ 0.654255 and TAX $\leq$ 0.337786 and RM $>$ 0.67053) mean = 0.631538

(13) If (LSTAT $\leq$ 0.360651 and RM $>$ 0.617072 and PTRATIO $\leq$ 0.654255 and TAX $>$ 0.337786) mean = 0.533127

(14) If (LSTAT $\leq$ 0.360651 and RM $>$ 0.617072 and PTRATIO $>$ 0.654255) mean = 0.436267

(15) If (LSTAT $>$ 0.360651 and LSTAT $\leq$ 0.50414 and PTRATIO $\leq$ 0.904256 and B $\leq$ 0.547116) mean = 0.192451

(16) If (LSTAT $>$ 0.360651 and LSTAT $\leq$ 0.50414 and B $>$ 0.547116 and INDUS $\leq$ 0.961694 and PTRATIO $\leq$ 0.664894) mean = 0.335965

(17) If (LSTAT $>$ 0.360651 and LSTAT $\leq$ 0.50414 and B $>$ 0.547116 and INDUS $\leq$ 0.961694 and PTRATIO $>$ 0.664894 and PTRATIO $\leq$ 0.904256) mean = 0.270574

(18) If (LSTAT $>$ 0.360651 and LSTAT $\leq$ 0.50414 and PTRATIO $\leq$ 0.904256 and B $>$ 0.547116 and INDUS $>$ 0.961694) mean = 0.040872

(19) If (LSTAT $>$ 0.360651 and LSTAT $\leq$ 0.50414 and PTRATIO $>$ 0.904256) mean = 0.130591

(20) If (LSTAT $>$ 0.50414 and PTRATIO $\leq$ 0.75 and NOX $\leq$ 0.407408) mean = 0.352574

(21) If (LSTAT $>$ 0.50414 and PTRATIO $\leq$ 0.75 and NOX $>$ 0.407408) mean = 0.238734

(22) If (PTRATIO $>$ 0.75 and LSTAT $>$ 0.50414 and LSTAT $\leq$ 0.611203 and CRIM $\leq$ 0.293871) mean = 0.177909

(23) If (PTRATIO $>$ 0.75 and LSTAT $>$ 0.50414 and LSTAT $\leq$ 0.611203 and CRIM $>$ 0.293871) mean = -0.0125655

(24) If (PTRATIO > 0.75 and LSTAT > 0.611203 and CRIM $\leq$ 0.0088955) mean = -0.0855045

(25) If (PTRATIO > 0.75 and LSTAT > 0.611203 and CRIM > 0.0088955) mean = 0.0708505

Forest fires DATASET

(1) If (TEMPERATURE <= 0.766881) mean = 0.00659569

(2) If (TEMPERATURE > 0.766881 and WIND <= 0.277778) mean = 0.019837

(3) If (TEMPERATURE > 0.766881 and WIND > 0.277778) mean = 0.00767183

Pollution DATASET

(1) If (EDUC <= 0.621212) mean = 0.608813

(2) If (EDUC > 0.621212 and HUMID <= 0.842857) mean = 0.353643

(3) If (EDUC > 0.621212 and HUMID > 0.842857) mean = 0.0618133

Rules extracted using the proposed approach for regression problems over reduced features when Gaussian function is used in DEWNN are presented below:

Rule extracted using DEWNN+DENFIS:

Auto MPG DATASET

(1) If CYLINDERS is GaussianMF(0.40 0.96) HORSEPOWER is GaussianMF(0.47 0.60) DISPLACEMENT is GaussianMF(0.48 0.57) WEIGHT is GaussianMF(0.49 0.58) then Miles per Gallon = 2.03 - 0.23 * CYLINDERS - 0.37 * HORSEPOWER - 0.13 * DISPLACEMENT - 0.43 * WEIGHT

(2) If CYLINDERS is GaussianMF(0.47 0.23) HORSEPOWER is GaussianMF(0.45 0.21) DISPLACEMENT is GaussianMF(0.47 0.27) WEIGHT is GaussianMF(0.47 0.37) then Miles per Gallon = 2.11 - 0.18 * CYLINDERS - 0.57 * HORSEPOWER - 0.24 * DISPLACEMENT - 0.41 * WEIGHT

(3) If CYLINDERS is GaussianMF(0.46 0.96) HORSEPOWER is GaussianMF(0.37 0.96) DISPLACEMENT is GaussianMF(0.33 0.96) WEIGHT is GaussianMF(0.40 0.91) then Miles per Gallon = 1.10 + 0.42 * CYLINDERS - 0.26 * HORSEPOWER - 0.17 * DISPLACEMENT - 0.11 * WEIGHT

Bodyfat DATASET

(1) If Density is GaussianMF(0.34 0.34) and Height is GaussianMF(0.54 0.72) and Chest is GaussianMF(0.39 0.47) and Abdomen is GaussianMF(0.37 0.39) and Thigh is GaussianMF(0.38 0.30) and Ankle is GaussianMF(0.53 0.36) and Forearm is GaussianMF(0.49 0.67) then Bodyfat = 1.76 - 1.15 * Density + 0.37 * Height - 0.19 * Chest - 0.24 * Abdomen + 0.27 * Thigh - 0.24 * Ankle + 0.69 * Forearm

(2) If Density is GaussianMF(0.41 0.94) and Height is GaussianMF(0.58 0.77) and Chest is GaussianMF(0.52 0.06) and Abdomen is GaussianMF(0.45 0.05) and Thigh is GaussianMF(0.52 0.05) and Ankle is GaussianMF(0.64 0.13) and Forearm is GaussianMF(0.24 0.27) then Bodyfat = 2.19 - 1.38 * Density - 0.36 * Height - 0.23 * Chest + 0.32 * Abdomen + 0.44 * Thigh - 0.29 * Ankle + 0.96 * Forearm

(3) If Density is GaussianMF(0.16 0.18) and Height is GaussianMF(0.40 0.85) and Chest is GaussianMF(0.60 0.93) and Abdomen is GaussianMF(0.60 0.93) and Thigh is GaussianMF(0.71 0.93) and Ankle is GaussianMF(0.68 0.66) and Forearm is GaussianMF(0.49 0.60) then Bodyfat = 1.13 + 0.49 * Density + 0.55 * Height - 0.02 * Chest + 0.21 * Abdomen - 0.31 * Thigh + 0.15 * Ankle + 0.16 * Forearm

Boston Housing DATASET

(1) If ISTAT is GaussianMF(0.76 0.23) and RM is GaussianMF(0.33 0.59) and NOX is GaussianMF(0.51 0.59) and TAX is GaussianMF(0.17 0.89) and X5 is GaussianMF(0.53 0.58) then MEDV = 1.73 + 0.04 * ISTAT - 0.32 * RM - 0.12 * NOX + 0.02 * TAX - 0.26 * X5

(2) If ISTAT is GaussianMF(0.50 0.05) and RM is GaussianMF(0.57 0.49) and NOX is GaussianMF(0.36 0.61) and TAX is GaussianMF(0.29 0.15) and X5 is GaussianMF(0.27 0.38) then MEDV = 0.82 + 0.85 * ISTAT - 0.13 * RM + 1.15 * NOX + 0.41 * TAX - 0.26 * X5

(3) If ISTAT is GaussianMF(0.51 0.07) and RM is GaussianMF(0.31 0.95) and NOX is GaussianMF(0.67 0.34) and TAX is GaussianMF(0.51 0.43) and X5

is GaussianMF(0.63 0.70) then MEDV = 1.68 - 0.07 * ISTAT - 0.11 * RM - 0.32 * NOX + 0.02 * TAX - 0.23 * X5

(4) If ISTAT is GaussianMF(0.19 0.98) and RM is GaussianMF(0.49 0.58) and NOX is GaussianMF(0.42 0.65) and TAX is GaussianMF(0.50 0.88) and X5 is GaussianMF(0.44 0.43) then MEDV = 0.26 - 0.02 * ISTAT - 0.65 * RM - 0.35 * NOX + 1.37 * TAX - 0.29 * X5

(5) If ISTAT is GaussianMF(0.70 0.11) and RM is GaussianMF(0.28 0.67) and NOX is GaussianMF(0.66 0.06) and TAX is GaussianMF(0.24 0.89) and X5 is GaussianMF(0.51 0.18) then MEDV = 1.69 + 0.10 * ISTAT - 0.59 * RM + 0.08 * NOX + 0.11 * TAX - 0.21 * X5

Forest fires DATASET

(1) If X-axis is GaussianMF(0.37 0.16) and FFMC is GaussianMF(0.38 0.04) and DMC is GaussianMF(0.49 0.16) and Temperature is GaussianMF(0.49 0.22) and RH is GaussianMF(0.18 0.44) and Rain is GaussianMF(0.50 0.05) then Area = 1.05 + 0.01 * X-axis - 0.41 * FFMC - 0.03 * DMC + 0.09 * Temperature - 0.29 * RH + 16.56 * Rain

(2) If X-axis is GaussianMF(0.18 0.75) and FFMC is GaussianMF(0.70 0.77) and DMC is GaussianMF(0.51 0.21) and Temperature is GaussianMF(0.58 0.60) and RH is GaussianMF(0.67 0.62) and Rain is GaussianMF(0.41 0.04) then Area = 1.39 + 0.44 * X-axis - 0.56 * FFMC - 0.16 * DMC + 0.53 * Temperature - 0.30 * RH - 0.76 * Rain

(3) If X-axis is GaussianMF(0.21 0.06) and FFMC is GaussianMF(0.55 0.89) and DMC is GaussianMF(0.53 0.34) and Temperature is GaussianMF(0.56 0.62) and RH is GaussianMF(0.53 0.41) and Rain is GaussianMF(0.50 0.04) then Area = 1.07 + 0.31 * X-axis - 0.70 * FFMC - 0.05 * DMC + 0.31 * Temperature - 0.19 * RH + 10.71 * Rain

(4) If X-axis is GaussianMF(0.38 0.49) and FFMC is GaussianMF(0.50 0.88) and DMC is GaussianMF(0.03 0.99) and Temperature is GaussianMF(0.44 0.44) and RH is GaussianMF(0.44 0.52) and Rain is GaussianMF(0.46 0.04) then Area = 1.47 + 0.34 * X-axis - 0.64 * FFMC - 0.01 * DMC + 0.43 * Temperature - 0.23 * RH - 0.92 * Rain

(5) If X-axis is GaussianMF(0.50 0.73) and FFMC is GaussianMF(0.50 0.95) and DMC is GaussianMF(0.50 0.61) and Temperature is GaussianMF(0.50 0.78) and RH is GaussianMF(0.50 0.56) and Rain is GaussianMF(0.51 0.95) then Area = 1.82 + 0.54 * X-axis - 1.20 * FFMC - 0.09 * DMC + 0.64 * Temperature - 0.53 * RH - 0.17 * Rain

Pollution DATASET

(1) If PREC is GaussianMF(0.50 0.51) and JULT is GaussianMF(0.50 0.42) and OVER65 is GaussianMF(0.50 0.67) and NONW is GaussianMF(0.50 0.23) and WWDRK is GaussianMF(0.50 0.37) and NOX is GaussianMF(0.50 0.21) and HUMID is GaussianMF(0.50 0.51) then MORT = 1.86 + 0.25 * PREC - 0.15 * JULT - 0.16 * OVER65 + 0.06 * NONW - 0.61 * WWDRK - 0.46 * NOX + 0.45 * HUMID

(2) If PREC is GaussianMF(0.50 0.77) and JULT is GaussianMF(0.50 0.83) and OVER65 is GaussianMF(0.50 0.32) and NONW is GaussianMF(0.50 0.95) and WWDRK is GaussianMF(0.49 0.28) and NOX is GaussianMF(0.50 0.09) and HUMID is GaussianMF(0.50 0.59) then MORT = 1.83 + 0.17 * PREC - 0.14 * JULT - 0.14 * OVER65 + 0.04 * NONW - 0.61 * WWDRK - 0.34 * NOX + 0.56 * HUMID

(3) If PREC is GaussianMF(0.50 0.05) and JULT is GaussianMF(0.50 0.25) and OVER65 is GaussianMF(0.50 0.62) and NONW is GaussianMF(0.50 0.22) and WWDRK is GaussianMF(0.50 0.51) and NOX is GaussianMF(0.51 0.95) and HUMID is GaussianMF(0.50 0.28) then MORT = 1.85 + 0.24 * PREC - 0.08 * JULT - 0.18 * OVER65 + 0.06 * NONW - 0.67 * WWDRK - 0.51 * NOX + 0.50 * HUMID

(4) If PREC is GaussianMF(0.50 0.60) and JULT is GaussianMF(0.50 0.67) and OVER65 is GaussianMF(0.50 0.14) and NONW is GaussianMF(0.50 0.68) and WWDRK is GaussianMF(0.48 0.96) and NOX is GaussianMF(0.50 0.12) and HUMID is GaussianMF(0.50 0.41) then MORT = 2.03 + 0.20 * PREC - 0.21 * JULT - 0.16 * OVER65 + 0.03 * NONW - 0.74 * WWDRK - 0.49 * NOX + 0.40 * HUMID

Rule extracted using DEWNN+CART:

Auto MPG DATASET

(1) If (WEIGHT $\leq$ 0.0981 and DISPLACEMENT $\leq$ 0.0671835) mean = -0.116826

(2) If (WEIGHT $\leq$ 0.0981 and DISPLACEMENT $>$ 0.0671835 and DISPLACEMENT $\leq$ 0.13824) mean = -0.111568

(3) If (HORSEPOWER $\leq$ 0.222825 and WEIGHT $>$ 0.0981 and WEIGHT $\leq$ 0.180605 and DISPLACEMENT $\leq$ 0.05814) mean = -0.122025

(4) If (HORSEPOWER $\leq$ 0.222825 and WEIGHT $>$ 0.0981 and WEIGHT $\leq$ 0.146015 and DISPLACEMENT $>$ 0.05814 and DISPLACEMENT $\leq$ 0.086563) mean = -0.117681

(5) If (HORSEPOWER $\leq$ 0.222825 and WEIGHT $>$ 0.0981 and WEIGHT $\leq$ 0.146015 and DISPLACEMENT $>$ 0.086563 and DISPLACEMENT $\leq$ 0.107235) mean = -0.111484

(6) If (HORSEPOWER $\leq$ 0.222825 and DISPLACEMENT $>$ 0.05814 and DISPLACEMENT $\leq$ 0.107235 and WEIGHT $>$ 0.146015 and WEIGHT $\leq$ 0.180605) mean = -0.119571

(7) If (HORSEPOWER $\leq$ 0.222825 and WEIGHT $>$ 0.0981 and WEIGHT $\leq$ 0.180605 and DISPLACEMENT $>$ 0.107235 and DISPLACEMENT $\leq$ 0.13824) mean = -0.113909

(8) If (DISPLACEMENT $\leq$ 0.13824 and WEIGHT $>$ 0.180605 and WEIGHT $\leq$ 0.337965 and HORSEPOWER $\leq$ 0.15489) mean = -0.126168

(9) If (DISPLACEMENT $\leq$ 0.13824 and WEIGHT $>$ 0.180605 and WEIGHT $\leq$ 0.337965 and HORSEPOWER $>$ 0.15489 and HORSEPOWER $\leq$ 0.222825) mean = -0.122692

(10) If (DISPLACEMENT $\leq$ 0.13824 and HORSEPOWER $>$ 0.222825 and WEIGHT $>$ 0.0981 and WEIGHT $\leq$ 0.189965) mean = -0.112168

(11) If (DISPLACEMENT $\leq$ 0.13824 and HORSEPOWER $>$ 0.222825 and WEIGHT $>$ 0.189965 and WEIGHT $\leq$ 0.28055) mean = -0.116375

(12) If (DISPLACEMENT $\leq$ 0.13824 and HORSEPOWER $>$ 0.222825 and WEIGHT $>$ 0.28055 and WEIGHT $\leq$ 0.337965) mean = -0.121738

(13) If (HORSEPOWER $\leq$ 0.347825 and DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.22093 and WEIGHT $\leq$ 0.194075) mean = -0.107629

(14) If (DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.22093 and WEIGHT $>$ 0.194075 and WEIGHT $\leq$ 0.30536 and HORSEPOWER $\leq$ 0.22826) mean = -0.113944

(15) If (DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.22093 and WEIGHT $>$ 0.194075 and WEIGHT $\leq$ 0.30536 and HORSEPOWER $>$ 0.22826 and HORSEPOWER $\leq$ 0.347825) mean = -0.110512

(16) If (HORSEPOWER $\leq$ 0.347825 and WEIGHT $\leq$ 0.30536 and DISPLACEMENT $>$ 0.22093 and DISPLACEMENT $\leq$ 0.742895) mean = -0.104623

(17) If (DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.742895 and HORSEPOWER $\leq$ 0.347825 and WEIGHT $>$ 0.30536 and WEIGHT $\leq$ 0.337965) mean = -0.115415

(18) If (WEIGHT $\leq$ 0.337965 and DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.742895 and HORSEPOWER $>$ 0.347825) mean = -0.122758

(19) If (WEIGHT $>$ 0.337965 and DISPLACEMENT $\leq$ 0.41344 and HORSEPOWER $\leq$ 0.17391) mean = -0.145416

(20) If (DISPLACEMENT $\leq$ 0.41344 and HORSEPOWER $>$ 0.17391 and HORSEPOWER $\leq$ 0.391305 and WEIGHT $>$ 0.337965 and WEIGHT $\leq$ 0.342925) mean = -0.14203

(21) If (DISPLACEMENT $\leq$ 0.41344 and HORSEPOWER $>$ 0.17391 and HORSEPOWER $\leq$ 0.391305 and WEIGHT $>$ 0.342925 and WEIGHT $\leq$ 0.36348) mean = -0.122796

(22) If (HORSEPOWER $>$ 0.17391 and HORSEPOWER $\leq$ 0.391305 and WEIGHT $>$ 0.36348 and DISPLACEMENT $\leq$ 0.13566) mean = -0.135005

(23) If (DISPLACEMENT $>$ 0.13566 and DISPLACEMENT $\leq$ 0.41344 and WEIGHT $>$ 0.36348 and WEIGHT $\leq$ 0.521405 and HORSEPOWER $>$ 0.17391 and HORSEPOWER $\leq$ 0.233695) mean = -0.130169

(24) If (DISPLACEMENT $>$ 0.13566 and DISPLACEMENT $\leq$ 0.41344 and WEIGHT $>$ 0.36348 and WEIGHT $\leq$ 0.521405 and HORSEPOWER $>$ 0.233695 and HORSEPOWER $\leq$ 0.271735) mean = -0.123186

(25) If (DISPLACEMENT $>$ 0.13566 and DISPLACEMENT $\leq$ 0.41344 and WEIGHT $>$ 0.36348 and WEIGHT $\leq$ 0.521405 and HORSEPOWER $>$ 0.271735 and HORSEPOWER $\leq$ 0.27989) mean = -0.137116

(26) If (DISPLACEMENT $>$ 0.13566 and DISPLACEMENT $\leq$ 0.41344 and WEIGHT $>$ 0.36348 and WEIGHT $\leq$ 0.521405 and HORSEPOWER $>$ 0.27989 and HORSEPOWER $\leq$ 0.391305) mean = -0.127781

(27) If (HORSEPOWER $>$ 0.17391 and HORSEPOWER $\leq$ 0.391305 and DISPLACEMENT $>$ 0.13566 and DISPLACEMENT $\leq$ 0.41344 and WEIGHT $>$ 0.521405) mean = -0.133556

(28) If (WEIGHT $>$ 0.337965 and DISPLACEMENT $\leq$ 0.41344 and HORSEPOWER $>$ 0.391305) mean = -0.148828

(29) If (DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.742895 and WEIGHT $>$ 0.337965 and WEIGHT $\leq$ 0.40091) mean = -0.105806

(30) If (DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.742895 and WEIGHT $>$ 0.40091 and WEIGHT $\leq$ 0.452085) mean = -0.113076

(31) If (HORSEPOWER $\leq$ 0.494565 and CYLINDERS $\leq$ 0.8 and WEIGHT $>$ 0.452085 and WEIGHT $\leq$ 0.51021 and DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.447025) mean = -0.121367

(32) If (HORSEPOWER $\leq$ 0.494565 and CYLINDERS $\leq$ 0.8 and WEIGHT $>$ 0.452085 and WEIGHT $\leq$ 0.51021 and DISPLACEMENT $>$ 0.447025 and DISPLACEMENT $\leq$ 0.687335) mean = -0.114692

(33) If (DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.687335 and HORSEPOWER $\leq$ 0.494565 and CYLINDERS $\leq$ 0.8 and WEIGHT $>$ 0.51021 and WEIGHT $\leq$ 0.66501) mean = -0.123498

(34) If (WEIGHT $>$ 0.452085 and WEIGHT $\leq$ 0.66501 and DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.687335 and HORSEPOWER $\leq$ 0.494565 and CYLINDERS $>$ 0.8) mean = -0.131544

(35) If (DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.687335 and HORSEPOWER $>$ 0.494565 and WEIGHT $>$ 0.452085 and WEIGHT $\leq$ 0.51999) mean = -0.113201

(36) If (DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.687335 and HORSEPOWER $>$ 0.494565 and WEIGHT $>$ 0.51999 and WEIGHT $\leq$ 0.66501) mean = -0.121215

(37) If (WEIGHT $>$ 0.452085 and WEIGHT $\leq$ 0.66501 and DISPLACEMENT $>$ 0.687335 and DISPLACEMENT $\leq$ 0.742895) mean = -0.111229

(38) If (WEIGHT $>$ 0.66501 and DISPLACEMENT $>$ 0.41344 and DISPLACEMENT $\leq$ 0.550385) mean = -0.158739

(39) If (WEIGHT $>$ 0.66501 and DISPLACEMENT $>$ 0.550385 and DISPLACEMENT $\leq$ 0.63178) mean = -0.141378

(40) If (WEIGHT $>$ 0.66501 and DISPLACEMENT $>$ 0.63178 and DISPLACEMENT $\leq$ 0.687335) mean = -0.131248

(41) If (DISPLACEMENT $>$ 0.687335 and DISPLACEMENT $\leq$ 0.742895 and WEIGHT $>$ 0.66501 and WEIGHT $\leq$ 0.73689) mean = -0.115563

(42) If (DISPLACEMENT $>$ 0.687335 and DISPLACEMENT $\leq$ 0.742895 and WEIGHT $>$ 0.73689 and WEIGHT $\leq$ 0.793025) mean = -0.120091

(43) If (DISPLACEMENT $>$ 0.687335 and DISPLACEMENT $\leq$ 0.742895 and WEIGHT $>$ 0.793025) mean = -0.127829

(44) If (DISPLACEMENT $>$ 0.742895 and WEIGHT $\leq$ 0.947125 and HORSEPOWER $\leq$ 0.6875) mean = -0.10937

(45) If (DISPLACEMENT $>$ 0.742895 and WEIGHT $\leq$ 0.947125 and HORSEPOWER $>$ 0.6875 and HORSEPOWER $\leq$ 0.85326) mean = -0.0995773

(46) If (DISPLACEMENT $>$ 0.742895 and HORSEPOWER $\leq$ 0.85326 and WEIGHT $>$ 0.947125) mean = -0.122549

(47) If (DISPLACEMENT $>$ 0.742895 and HORSEPOWER $>$ 0.85326) mean = -0.08473

Bodyfat DATASET

(1) If (DENSITY $\leq$ 0.654965 and ANKLE $\leq$ 0.662165 and FOREARM $\leq$ 0.384895) mean = 0.289954

(2) If (DENSITY $\leq$ 0.654965 and ANKLE $\leq$ 0.662165 and FOREARM $>$ 0.384895 and FOREARM $\leq$ 0.48561) mean = 0.372198

(3) If (FOREARM $\leq$ 0.48561 and DENSITY $\leq$ 0.654965 and ANKLE $>$ 0.662165) mean = 0.046074

(4) If (DENSITY $>$ 0.654965 and FOREARM $\leq$ 0.36331) mean = 0.127847

(5) If (DENSITY $>$ 0.654965 and FOREARM $>$ 0.36331 and FOREARM $\leq$ 0.48561 and CHEST $\leq$ 0.36643 and THIGH $\leq$ 0.197005) mean = 0.225582

(6) If (DENSITY $>$ 0.654965 and FOREARM $>$ 0.36331 and FOREARM $\leq$ 0.48561 and CHEST $\leq$ 0.36643 and THIGH $>$ 0.197005) mean = 0.289161

(7) If (DENSITY $>$ 0.654965 and FOREARM $>$ 0.36331 and FOREARM $\leq$ 0.48561 and CHEST $>$ 0.36643) mean = 0.009781

(8) If (FOREARM $>$ 0.48561 and DENSITY $\leq$ 0.514485) mean = 0.432527

(9) If (FOREARM $>$ 0.48561 and DENSITY $>$ 0.514485 and DENSITY $\leq$ 0.66286) mean = 0.401337

(10) If (DENSITY $>$ 0.66286 and FOREARM $>$ 0.48561 and FOREARM $\leq$ 0.57194) mean = 0.312606

(11) If (DENSITY $>$ 0.66286 and FOREARM $>$ 0.57194) mean = 0.385311

Boston Housing DATASET

(1) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $\leq$ 0.21296 and ISTAT $\leq$ 0.128585) mean = -0.0767745

(2) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $\leq$ 0.21296 and ISTAT $>$ 0.128585 and ISTAT $\leq$ 0.28532 and CRIM $\leq$ 0.0003655) mean = -0.0730937

(3) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $\leq$ 0.21296 and CRIM $>$ 0.0003655 and ISTAT $>$ 0.128585 and ISTAT $\leq$ 0.2144) mean = -0.0668163

(4) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $\leq$ 0.21296 and CRIM $>$ 0.0003655 and ISTAT $>$ 0.2144 and ISTAT $\leq$ 0.28532) mean = -0.0597075

(5) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.21296 and NOX $\leq$ 0.297325 and ISTAT $\leq$ 0.23772) mean = -0.0813508

(6) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.21296 and NOX $\leq$ 0.297325 and ISTAT $>$ 0.23772 and ISTAT $\leq$ 0.28532) mean = -0.073779

(7) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and ISTAT $\leq$ 0.28532 and NOX $>$ 0.297325) mean = -0.0940233

(8) If (TAX $\leq$ 0.33779 and ISTAT $>$ 0.28532 and NOX $\leq$ 0.218105 and RM $\leq$ 0.42872) mean = -0.0456652

(9) If (TAX $\leq$ 0.33779 and ISTAT $>$ 0.28532 and NOX $\leq$ 0.218105 and RM $>$ 0.42872 and RM $\leq$ 0.51667) mean = -0.0539329

(10) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.218105 and ISTAT $>$ 0.28532 and ISTAT $\leq$ 0.33554) mean = -0.078812

(11) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.218105 and ISTAT $>$ 0.33554 and ISTAT $\leq$ 0.43322) mean = -0.071192

(12) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.218105 and ISTAT $>$ 0.43322 and ISTAT $\leq$ 0.484825) mean = -0.0624365

(13) If (TAX $\leq$ 0.33779 and RM $\leq$ 0.51667 and NOX $>$ 0.218105 and ISTAT $>$ 0.484825) mean = -0.0553199

(14) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and ISTAT $\leq$ 0.134105 and NOX $\leq$ 0.057613) mean = -0.075225

(15) If (RM $>$ 0.51667 and TAX $\leq$ 0.244275 and NOX $>$ 0.057613 and NOX $\leq$ 0.11214 and ISTAT $\leq$ 0.0193155) mean = -0.086118

(16) If (RM $>$ 0.51667 and TAX $\leq$ 0.244275 and NOX $>$ 0.057613 and NOX $\leq$ 0.11214 and ISTAT $>$ 0.0193155 and ISTAT $\leq$ 0.134105) mean = -0.0776657

(17) If (RM $>$ 0.51667 and TAX $\leq$ 0.244275 and NOX $>$ 0.11214 and NOX $\leq$ 0.21296 and ISTAT $\leq$ 0.085541) mean = -0.0854

(18) If (RM $>$ 0.51667 and TAX $\leq$ 0.244275 and NOX $>$ 0.11214 and NOX $\leq$ 0.21296 and ISTAT $>$ 0.085541 and ISTAT $\leq$ 0.134105) mean = -0.0804405

(19) If (RM $>$ 0.51667 and ISTAT $\leq$ 0.134105 and NOX $>$ 0.057613 and NOX $\leq$ 0.21296 and TAX $>$ 0.244275 and TAX $\leq$ 0.33779) mean = -0.0855967

(20) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and ISTAT $\leq$ 0.134105 and NOX $>$ 0.21296 and NOX $\leq$ 0.218105) mean = -0.091416

(21) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and ISTAT $>$ 0.134105 and ISTAT $\leq$ 0.18364 and NOX $\leq$ 0.11523) mean = -0.0720568

(22) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and ISTAT $>$ 0.134105 and ISTAT $\leq$ 0.18364 and NOX $>$ 0.11523 and NOX $\leq$ 0.21296) mean = -0.0762507

(23) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and NOX $\leq$ 0.21296 and ISTAT $>$ 0.18364 and ISTAT $\leq$ 0.212885) mean = -0.0696773

(24) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and ISTAT $>$ 0.134105 and ISTAT $\leq$ 0.212885 and NOX $>$ 0.21296 and NOX $\leq$ 0.218105) mean = -0.0808087

(25) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and NOX $\leq$ 0.218105 and ISTAT $>$ 0.212885) mean = -0.0636143

(26) If (TAX $\leq$ 0.33779 and NOX $>$ 0.218105 and RM $>$ 0.51667 and RM $\leq$ 0.61755 and ISTAT $\leq$ 0.104169) mean = -0.0970717

(27) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and RM $\leq$ 0.61755 and ISTAT $>$ 0.104169 and NOX $>$ 0.218105 and NOX $\leq$ 0.32099) mean = -0.0891799

(28) If (TAX $\leq$ 0.33779 and RM $>$ 0.51667 and RM $\leq$ 0.61755 and ISTAT $>$ 0.104169 and NOX $>$ 0.32099) mean = -0.0942965

(29) If (TAX $\leq$ 0.33779 and NOX $>$ 0.218105 and RM $>$ 0.61755 and RM $\leq$ 0.920965) mean = -0.0990969

(30) If (TAX $\leq$ 0.33779 and NOX $>$ 0.218105 and RM $>$ 0.920965) mean = -0.093315

(31) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and NOX $\leq$ 0.290125 and RM $\leq$ 0.438975) mean = -0.068136

(32) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $>$ 0.438975 and RM $\leq$ 0.526635 and NOX $\leq$ 0.13066) mean = -0.0790669

(33) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $>$ 0.438975 and RM $\leq$ 0.526635 and NOX $>$ 0.13066 and NOX $\leq$ 0.290125) mean = -0.0872906

(34) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $>$ 0.526635 and RM $\leq$ 0.9612 and NOX $\leq$ 0.080249) mean = -0.090078

(35) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $>$ 0.526635 and RM $\leq$ 0.9612 and NOX $>$ 0.080249 and NOX $\leq$ 0.290125) mean = -0.0954597

(36) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $\leq$ 0.26992 and ISTAT $\leq$ 0.18226 and NOX $>$ 0.290125 and NOX $\leq$ 0.738685) mean = -0.094883

(37) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $\leq$ 0.26992 and ISTAT $\leq$ 0.18226 and NOX $>$ 0.738685) mean = -0.073739

(38) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and NOX $>$ 0.290125 and CRIM $\leq$ 0.0031275 and ISTAT $>$ 0.18226 and ISTAT $\leq$ 0.32754) mean = -0.0990293

(39) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $\leq$ 0.0031275 and ISTAT $>$ 0.32754 and ISTAT $\leq$ 0.89583 and NOX $>$ 0.290125 and NOX $\leq$ 0.436215) mean = -0.0921769

(40) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $\leq$ 0.0031275 and ISTAT $>$ 0.32754 and ISTAT $\leq$ 0.89583 and NOX $>$ 0.436215) mean = -0.0971685

(41) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and NOX $>$ 0.290125 and ISTAT $>$ 0.18226 and ISTAT $\leq$ 0.89583 and CRIM $>$ 0.0031275 and CRIM $\leq$ 0.003377) mean = -0.081761

(42) If (TAX $>$ 0.33779 and ISTAT $>$ 0.18226 and ISTAT $\leq$ 0.89583 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.290125 and NOX $\leq$ 0.64095 and RM $\leq$ 0.345755) mean = -0.093269

(43) If (TAX $>$ 0.33779 and ISTAT $>$ 0.18226 and ISTAT $\leq$ 0.89583 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.290125 and NOX $\leq$ 0.64095 and RM $>$ 0.345755 and RM $\leq$ 0.9612) mean = -0.100468

(44) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.64095 and ISTAT $>$ 0.18226 and ISTAT $\leq$ 0.3572) mean = -0.0899927

(45) If (TAX $>$ 0.33779 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.64095 and ISTAT $>$ 0.3572 and ISTAT $\leq$ 0.50924 and RM $\leq$ 0.67254) mean = -0.0964781

(46) If (TAX $>$ 0.33779 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.64095 and ISTAT $>$ 0.3572 and ISTAT $\leq$ 0.50924 and RM $>$ 0.67254 and RM $\leq$ 0.9612) mean = -0.089334

(47) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.64095 and ISTAT $>$ 0.50924 and ISTAT $\leq$ 0.78215) mean = -0.101709

(48) If (TAX $>$ 0.33779 and RM $\leq$ 0.9612 and CRIM $>$ 0.003377 and CRIM $\leq$ 0.26992 and NOX $>$ 0.64095 and ISTAT $>$ 0.78215 and ISTAT $\leq$ 0.89583) mean = -0.0900067

(49) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $\leq$ 0.9612 and NOX $>$ 0.290125 and CRIM $>$ 0.26992) mean = -0.086012

(50) If (TAX $>$ 0.33779 and ISTAT $\leq$ 0.89583 and RM $>$ 0.9612) mean = -0.051975

(51) If (TAX $>$ 0.33779 and ISTAT $>$ 0.89583) mean = -0.0572142

Forest fires DATASET

(1) If (FFMC $\leq$ 0.88645) mean = 0.00321495

(2) If (FFMC $>$ 0.88645) mean = 0.00957039

Pollution DATASET

(1) If (WWDRK $\leq$ 0.6583) mean = 0.47349

(2) If (WWDRK $>$ 0.6583) mean = 0.318007

Rules extracted using the proposed approach for regression problems over reduced features when Morlet function is used in DEWNN are presented below:

Rule extracted using DEWNN+DENFIS:

Auto MPG DATASET

(1) If Displacement is GaussianMF(0.22 0.61) and Horsepower is GaussianMF(0.42 0.59) and Weight is GaussianMF(0.30 0.59) and Model year is GaussianMF(0.59 0.29) then Auto MPG = 2.57 - 0.85 * Displacement - 0.30 * Horsepower - 0.41 * Weight - 0.58 * Model year

(2) If Displacement is GaussianMF(0.49 0.21) and Horsepower is GaussianMF(0.50 0.28) and Weight is GaussianMF(0.48 0.37) and Model year is GaussianMF(0.22 0.97) then Auto MPG = 2.38 - 1.05 * Displacement - 0.22 * Horsepower - 0.11 * Weight - 0.43 * Model year

(3) If Displacement is GaussianMF(0.14 0.06) and Horsepower is GaussianMF(0.45 0.18) and Weight is GaussianMF(0.27 0.15) and Model year is GaussianMF(0.38 0.32) then Auto MPG = 2.09 - 0.61 * Displacement - 0.00 * Horsepower - 0.17 * Weight - 0.16 * Model year

Bodyfat DATASET

(1) If Density is GaussianMF(0.27 0.22) and Weight is GaussianMF(0.49 0.50) and Neck is GaussianMF(0.39 0.74) and Chest is GaussianMF(0.49 0.56) and Thigh is GaussianMF(0.50 0.45) and Knee is GaussianMF(0.50 0.46) and Biceps is GaussianMF(0.51 0.75) then Bodyfat = 2.19 - 0.80 * Density + 0.05 * Weight + 0.62 * Neck - 0.57 * Chest - 0.04 * Thigh + 0.32 * Knee - 0.47 * Biceps

(2) If Density is GaussianMF(0.24 0.76) and Weight is GaussianMF(0.48 0.20) and Neck is GaussianMF(0.47 0.31) and Chest is GaussianMF(0.45 0.29) and Thigh is GaussianMF(0.49 0.29) and Knee is GaussianMF(0.50 0.23) and Biceps is GaussianMF(0.59 0.35) then Bodyfat = 2.11 - 0.92 * Density + 0.08 * Weight + 0.63 * Neck - 0.19 * Chest + 0.17 * Thigh + 0.25 * Knee - 0.65 * Biceps

Boston Housing DATASET

(1) If CRIM is GaussianMF(0.45 0.04) and AGE is GaussianMF(-0.27 0.69) and DIS is GaussianMF(0.52 0.15) and TAX is GaussianMF(0.24 0.40) and ISTAT is GaussianMF(0.11 0.40) then MEDV = 2.00 - 0.25 * CRIM + 0.25 * AGE + 0.05 * DIS - 0.39 * TAX - 1.06 * ISTAT

(2) If CRIM is GaussianMF(0.33 0.04) and AGE is GaussianMF(0.81 0.12) and DIS is GaussianMF(0.56 0.37) and TAX is GaussianMF(0.52 0.26) and ISTAT is GaussianMF(-0.13 0.20) then MEDV = 1.80 - 3.19 * CRIM + 0.44 * AGE + 0.19 * DIS + 0.06 * TAX - 0.64 * ISTAT

(3) If CRIM is GaussianMF(0.34 0.46) and AGE is GaussianMF(0.47 0.96) and DIS is GaussianMF(0.55 0.08) and TAX is GaussianMF(0.43 0.90) and ISTAT is GaussianMF(0.85 0.76) then MEDV = 2.51 + 0.31 * CRIM - 0.29 * AGE - 0.72 * DIS - 0.57 * TAX - 0.84 * ISTAT

Forest fires DATASET

(1) If FFMC is GaussianMF(0.55 0.90) and DC is GaussianMF(0.52 0.80) and ISI is GaussianMF(0.50 0.39) and Temperature is GaussianMF(0.51 0.77) and WIND is GaussianMF(0.50 0.33) and RAIN is GaussianMF(0.33 0.04) then AREA = 1.55 + 0.38 * FFMC - 0.63 * DC + 0.12 * ISI + 0.58 * Temperature - 0.62 * WIND + 0.84 * RAIN

(2) If FFMC is GaussianMF(0.57 0.78) and DC is GaussianMF(0.47 0.15) and ISI is GaussianMF(0.51 0.24) and Temperature is GaussianMF(0.51 0.30) and WIND is GaussianMF(0.54 0.56) and RAIN is GaussianMF(0.50 0.05) then AREA = 1.60 + 0.34 * FFMC - 0.64 * DC + 0.14 * ISI + 0.57 * Temperature - 0.59 * WIND + 0.07 * RAIN

(3) If FFMC is GaussianMF(0.50 0.95) and DC is GaussianMF(0.50 0.76) and ISI is GaussianMF(0.50 0.65) and Temperature is GaussianMF(0.50 0.78) and WIND is GaussianMF(0.50 0.47) and RAIN is GaussianMF(0.54 0.95) then AREA = 1.58 + 0.40 * FFMC - 0.55 * DC + 0.03 * ISI + 0.48 * Temperature - 0.44 * WIND - 0.01 * RAIN

(4) If FFMC is GaussianMF(0.50 0.90) and DC is GaussianMF(0.50 0.67) and ISI is GaussianMF(0.53 0.91) and Temperature is GaussianMF(0.44 0.13) and WIND is GaussianMF(0.49 0.57) and RAIN is GaussianMF(0.41 0.04) then AREA = 1.55 + 0.42 * FFMC - 0.69 * DC + 0.12 * ISI + 0.61 * Temperature - 0.64 * WIND + 0.89 * RAIN

(5) If FFMC is GaussianMF(0.21 0.09) and DC is GaussianMF(0.46 0.29) and ISI is GaussianMF(0.49 0.04) and Temperature is GaussianMF(0.48 0.29) and WIND is GaussianMF(0.47 0.23) and RAIN is GaussianMF(0.50 0.05) then AREA = 0.78 + 0.30 * FFMC - 0.62 * DC + 0.25 * ISI + 0.57 * Temperature - 0.54 * WIND + 21.91 * RAIN

Pollution DATASET

(1) If PREC is GaussianMF(0.50 0.83) and OVER65 is GaussianMF(0.50 0.35) and EDUC is GaussianMF(0.49 0.27) and NONW is GaussianMF(0.49 0.96) and POOR is GaussianMF(0.50 0.91) and NOX is GaussianMF(0.50 0.21) and HUMID is GaussianMF(0.50 0.46) then MORT = 1.61 + 0.31 * PREC + 0.55 * OVER65 - 0.22 * EDUC + 0.45 * NONW - 0.25 * POOR + 0.03 * NOX - 0.47 * HUMID

(2) If PREC is GaussianMF(0.41 0.04) and OVER65 is GaussianMF(0.38 0.29) and EDUC is GaussianMF(0.40 0.89) and NONW is GaussianMF(0.50 0.17) and POOR is GaussianMF(0.50 0.28) and NOX is GaussianMF(0.48 0.39) and HUMID is GaussianMF(0.51 0.64) then MORT = 1.29 + 0.46 * PREC + 0.94 * OVER65 - 0.22 * EDUC + 0.34 * NONW + 0.15 * POOR + 0.03 * NOX - 0.43 * HUMID

(3) If PREC is GaussianMF(0.48 0.66) and OVER65 is GaussianMF(0.41 0.83) and EDUC is GaussianMF(0.41 0.89) and NONW is GaussianMF(0.50 0.11)

and POOR is GaussianMF(0.58 0.14) and NOX is GaussianMF(0.55 0.21) and HUMID is GaussianMF(0.48 0.52) then MORT = 1.30 + 0.21 * PREC + 0.75 * OVER65 - 0.20 * EDUC + 0.67 * NONW - 0.53 * POOR + 0.06 * NOX - 0.03 * HUMID

(4) If PREC is GaussianMF(0.42 0.40) and OVER65 is GaussianMF(0.51 0.44) and EDUC is GaussianMF(0.36 0.16) and NONW is GaussianMF(0.52 0.34) and POOR is GaussianMF(0.48 0.48) and NOX is GaussianMF(0.49 0.24) and HUMID is GaussianMF(0.50 0.54) then MORT = 1.09 + 0.41 * PREC + 0.65 * OVER65 - 0.00 * EDUC + 0.56 * NONW - 0.17 * POOR + 0.23 * NOX - 0.07 * HUMID

(5) If PREC is GaussianMF(0.53 0.61) and OVER65 is GaussianMF(0.53 0.14) and EDUC is GaussianMF(0.54 0.95) and NONW is GaussianMF(0.56 0.64) and POOR is GaussianMF(0.51 0.08) and NOX is GaussianMF(0.45 0.19) and HUMID is GaussianMF(0.51 0.41) then MORT = 1.55 + 0.47 * PREC + 0.79 * OVER65 - 0.33 * EDUC + 0.26 * NONW + 0.09 * POOR - 0.21 * NOX - 0.57 * HUMID

Rule extracted using DEWNN+CART:
Auto MPG DATASET

(1) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $\leq$ 0.25 and HORSEPOWER $\leq$ 0.0597825) mean = 0.481234

(2) If (MODEL YEAR $\leq$ 0.25 and HORSEPOWER $>$ 0.0597825 and HORSEPOWER $\leq$ 0.225545 and DISPLACEMENT $\leq$ 0.13824 and WEIGHT $\leq$ 0.159765) mean = 0.403338

(3) If (MODEL YEAR $\leq$ 0.25 and HORSEPOWER $>$ 0.0597825 and HORSEPOWER $\leq$ 0.225545 and DISPLACEMENT $\leq$ 0.13824 and WEIGHT $>$ 0.159765) mean = 0.432997

(4) If (MODEL YEAR $\leq$ 0.25 and HORSEPOWER $>$ 0.0597825 and HORSEPOWER $\leq$ 0.225545 and DISPLACEMENT $>$ 0.13824 and DISPLACEMENT $\leq$ 0.37339) mean = 0.376497

(5) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER $>$ 0.225545 and MODEL YEAR $\leq$ 0.125001) mean = 0.336636

(6) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER $>$ 0.225545 and MODEL YEAR $>$ 0.125001 and MODEL YEAR $\leq$ 0.25) mean = 0.370679

(7) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.25 and MODEL YEAR $\leq$ 0.458335 and HORSEPOWER $\leq$ 0.14674) mean = 0.477045

(8) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.25 and MODEL YEAR $\leq$ 0.458335 and HORSEPOWER $>$ 0.14674 and HORSEPOWER $\leq$ 0.24185) mean = 0.457267

(9) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.25 and MODEL YEAR $\leq$ 0.458335 and HORSEPOWER $>$ 0.24185) mean = 0.422775

(10) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.458335 and MODEL YEAR $\leq$ 0.625 and HORSEPOWER $\leq$ 0.125005) mean = 0.508065

(11) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.458335 and MODEL YEAR $\leq$ 0.625 and HORSEPOWER $>$ 0.125005 and HORSEPOWER $\leq$ 0.171195) mean = 0.483831

(12) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.625 and MODEL YEAR $\leq$ 0.791665 and HORSEPOWER $\leq$ 0.10598) mean = 0.531739

(13) If (DISPLACEMENT $\leq$ 0.37339 and MODEL YEAR $>$ 0.625 and MODEL YEAR $\leq$ 0.791665 and HORSEPOWER $>$ 0.10598 and HORSEPOWER $\leq$ 0.171195) mean = 0.513165

(14) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER $>$ 0.171195 and HORSEPOWER $\leq$ 0.244565 and MODEL YEAR $>$ 0.458335 and MODEL YEAR $\leq$ 0.708335 and WEIGHT $\leq$ 0.44145) mean = 0.474834

(15) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER $>$ 0.171195 and HORSEPOWER $\leq$ 0.244565 and MODEL YEAR $>$ 0.458335 and MODEL YEAR $\leq$ 0.708335 and WEIGHT $>$ 0.44145) mean = 0.49956

(16) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER > 0.171195 and HORSEPOWER $\leq$ 0.244565 and MODEL YEAR > 0.708335 and MODEL YEAR $\leq$ 0.791665) mean = 0.495696

(17) If (HORSEPOWER $\leq$ 0.244565 and MODEL YEAR > 0.791665 and DISPLACEMENT $\leq$ 0.20672) mean = 0.53518

(18) If (HORSEPOWER $\leq$ 0.244565 and MODEL YEAR > 0.791665 and DISPLACEMENT > 0.20672 and DISPLACEMENT $\leq$ 0.37339) mean = 0.515537

(19) If (HORSEPOWER > 0.244565 and MODEL YEAR > 0.458335 and MODEL YEAR $\leq$ 0.833335 and DISPLACEMENT $\leq$ 0.21447) mean = 0.458113

(20) If (HORSEPOWER > 0.244565 and MODEL YEAR > 0.458335 and MODEL YEAR $\leq$ 0.833335 and DISPLACEMENT > 0.21447 and DISPLACEMENT $\leq$ 0.37339) mean = 0.422533

(21) If (DISPLACEMENT $\leq$ 0.37339 and HORSEPOWER > 0.244565 and MODEL YEAR > 0.833335) mean = 0.493041

(22) If (DISPLACEMENT > 0.37339 and DISPLACEMENT $\leq$ 0.55297 and MODEL YEAR $\leq$ 0.250001) mean = 0.248189

(23) If (DISPLACEMENT > 0.37339 and DISPLACEMENT $\leq$ 0.55297 and MODEL YEAR > 0.250001 and MODEL YEAR $\leq$ 0.541665 and HORSEPOWER $\leq$ 0.307065) mean = 0.389201

(24) If (DISPLACEMENT > 0.37339 and DISPLACEMENT $\leq$ 0.55297 and MODEL YEAR > 0.250001 and MODEL YEAR $\leq$ 0.541665 and HORSEPOWER > 0.307065 and HORSEPOWER $\leq$ 0.33424) mean = 0.345481

(25) If (DISPLACEMENT > 0.37339 and DISPLACEMENT $\leq$ 0.55297 and HORSEPOWER $\leq$ 0.33424 and MODEL YEAR > 0.541665 and MODEL YEAR $\leq$ 0.708335) mean = 0.411903

(26) If (DISPLACEMENT > 0.37339 and DISPLACEMENT $\leq$ 0.55297 and MODEL YEAR > 0.250001 and MODEL YEAR $\leq$ 0.708335 and HORSEPOWER > 0.33424) mean = 0.324175

(27) If (DISPLACEMENT > 0.37339 and DISPLACEMENT ≤ 0.55297 and MODEL YEAR > 0.708335) mean $= 0.439781$

(28) If (DISPLACEMENT > 0.55297 and HORSEPOWER ≤ 0.614135 and MODEL YEAR ≤ 0.125001) mean $= 0.0520118$

(29) If (DISPLACEMENT > 0.55297 and MODEL YEAR > 0.125001 and MODEL YEAR ≤ 0.291665 and HORSEPOWER ≤ 0.524455) mean $= 0.171588$

(30) If (MODEL YEAR > 0.125001 and MODEL YEAR ≤ 0.291665 and HORSEPOWER > 0.524455 and HORSEPOWER ≤ 0.614135 and DISPLACEMENT > 0.55297 and DISPLACEMENT ≤ 0.729975) mean $= 0.0989105$

(31) If (MODEL YEAR > 0.125001 and MODEL YEAR ≤ 0.291665 and HORSEPOWER > 0.524455 and HORSEPOWER ≤ 0.614135 and DISPLACEMENT > 0.729975) mean $= 0.0686527$

(32) If (DISPLACEMENT > 0.55297 and MODEL YEAR > 0.291665 and HORSEPOWER ≤ 0.36141) mean $= 0.356819$

(33) If (DISPLACEMENT > 0.55297 and MODEL YEAR > 0.291665 and HORSEPOWER > 0.36141 and HORSEPOWER ≤ 0.52989 and WEIGHT ≤ 0.50312) mean $= 0.197255$

(34) If (DISPLACEMENT > 0.55297 and MODEL YEAR > 0.291665 and WEIGHT > 0.50312 and HORSEPOWER > 0.36141 and HORSEPOWER ≤ 0.49185) mean $= 0.282426$

(35) If (DISPLACEMENT > 0.55297 and MODEL YEAR > 0.291665 and WEIGHT > 0.50312 and HORSEPOWER > 0.49185 and HORSEPOWER ≤ 0.52989) mean $= 0.242095$

(36) If (DISPLACEMENT > 0.55297 and HORSEPOWER > 0.52989 and HORSEPOWER ≤ 0.614135 and MODEL YEAR > 0.291665 and MODEL YEAR ≤ 0.541665) mean $= 0.174652$

(37) If (DISPLACEMENT > 0.55297 and HORSEPOWER > 0.52989 and HORSEPOWER ≤ 0.614135 and MODEL YEAR > 0.541665) mean $= 0.211448$

(38) If (HORSEPOWER > 0.614135 and MODEL YEAR $\leq$ 0.375 and DISPLACEMENT > 0.55297 and DISPLACEMENT $\leq$ 0.835915) mean = 0.0125348

(39) If (HORSEPOWER > 0.614135 and MODEL YEAR $\leq$ 0.375 and DISPLACEMENT > 0.835915 and DISPLACEMENT $\leq$ 0.89535) mean = -0.0316863

(40) If (HORSEPOWER > 0.614135 and DISPLACEMENT > 0.55297 and DISPLACEMENT $\leq$ 0.89535 and MODEL YEAR > 0.375) mean = 0.0628513

(41) If (HORSEPOWER > 0.614135 and DISPLACEMENT > 0.89535) mean = -0.108177

Bodyfat DATASET

(1) If (DENSITY $\leq$ 0.16725) mean = 0.241918

(2) If (DENSITY > 0.16725 and DENSITY $\leq$ 0.406495) mean = 0.432444

(3) If (DENSITY > 0.406495 and DENSITY $\leq$ 0.613695) mean = 0.394895

(4) If (DENSITY > 0.613695) mean = 0.352495

Boston Housing DATASET

(1) If (DIS $\leq$ 0.48489 and TAX $\leq$ 0.26813 and ISTAT $\leq$ 0.0593265) mean = 0.565917

(2) If (DIS $\leq$ 0.48489 and TAX $\leq$ 0.26813 and ISTAT > 0.0593265 and ISTAT $\leq$ 0.084023) mean = 0.54494

(3) If (DIS $\leq$ 0.48489 and ISTAT $\leq$ 0.084023 and TAX > 0.26813) mean = 0.519812

(4) If (DIS $\leq$ 0.48489 and ISTAT > 0.084023 and ISTAT $\leq$ 0.14597 and AGE $\leq$ 0.792995) mean = 0.526752

(5) If (DIS $\leq$ 0.48489 and ISTAT > 0.084023 and ISTAT $\leq$ 0.14597 and AGE > 0.792995) mean = 0.497316

(6) If (ISTAT $\leq$ 0.14597 and DIS > 0.48489 and DIS $\leq$ 0.58157) mean = 0.469039

(7) If (DIS $\leq$ 0.470445 and AGE $\leq$ 0.74871 and ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.234825 and TAX $\leq$ 0.20134) mean = 0.505973

(8) If (DIS $\leq$ 0.470445 and AGE $\leq$ 0.74871 and ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.234825 and TAX $>$ 0.20134 and TAX $\leq$ 0.40744) mean = 0.485266

(9) If (TAX $\leq$ 0.40744 and DIS $\leq$ 0.470445 and AGE $\leq$ 0.74871 and ISTAT $>$ 0.234825 and ISTAT $\leq$ 0.268075) mean = 0.472529

(10) If (ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.268075 and TAX $\leq$ 0.40744 and DIS $\leq$ 0.470445 and AGE $>$ 0.74871) mean = 0.468777

(11) If (ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.268075 and TAX $\leq$ 0.40744 and DIS $>$ 0.470445 and DIS $\leq$ 0.58157) mean = 0.439222

(12) If (DIS $\leq$ 0.58157 and ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.268075 and TAX $>$ 0.40744 and TAX $\leq$ 0.47233 and CRIM $\leq$ 0.000679) mean = 0.420202

(13) If (DIS $\leq$ 0.58157 and ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.268075 and TAX $>$ 0.40744 and TAX $\leq$ 0.47233 and CRIM $>$ 0.000679) mean = 0.451171

(14) If (DIS $\leq$ 0.58157 and ISTAT $>$ 0.14597 and ISTAT $\leq$ 0.268075 and TAX $>$ 0.47233) mean = 0.417957

(15) If (ISTAT $\leq$ 0.268075 and TAX $\leq$ 0.41889 and DIS $>$ 0.58157 and DIS $\leq$ 0.72147 and CRIM $\leq$ 0.0001135) mean = 0.464799

(16) If (ISTAT $\leq$ 0.268075 and TAX $\leq$ 0.41889 and DIS $>$ 0.58157 and DIS $\leq$ 0.72147 and CRIM $>$ 0.0001135) mean = 0.410094

(17) If (ISTAT $\leq$ 0.268075 and TAX $\leq$ 0.41889 and DIS $>$ 0.72147) mean = 0.374015

(18) If (ISTAT $\leq$ 0.268075 and DIS $>$ 0.58157 and TAX $>$ 0.41889) mean = 0.279276

(19) If (ISTAT $>$ 0.268075 and ISTAT $\leq$ 0.34575 and DIS $\leq$ 0.560905 and AGE $\leq$ 0.816685 and CRIM $\leq$ 0.0248575) mean = 0.433518

(20) If (ISTAT $>$ 0.268075 and ISTAT $\leq$ 0.34575 and DIS $\leq$ 0.560905 and AGE $\leq$ 0.816685 and CRIM $>$ 0.0248575 and CRIM $\leq$ 0.032519) mean = 0.478952

(21) If (CRIM ≤ 0.032519 and DIS ≤ 0.560905 and AGE > 0.816685 and ISTAT > 0.268075 and ISTAT ≤ 0.29622) mean $= 0.416133$

(22) If (CRIM ≤ 0.032519 and DIS ≤ 0.560905 and AGE > 0.816685 and ISTAT > 0.29622 and ISTAT ≤ 0.34575) mean $= 0.388988$

(23) If (CRIM ≤ 0.032519 and ISTAT > 0.268075 and ISTAT ≤ 0.34575 and DIS > 0.560905) mean $= 0.359066$

(24) If (CRIM ≤ 0.032519 and ISTAT > 0.34575 and ISTAT ≤ 0.399695 and AGE ≤ 0.78424) mean $= 0.387589$

(25) If (CRIM ≤ 0.032519 and ISTAT > 0.34575 and ISTAT ≤ 0.399695 and AGE > 0.78424) mean $= 0.361293$

(26) If (CRIM > 0.032519 and ISTAT > 0.268075 and ISTAT ≤ 0.34175 and DIS ≤ 0.1454) mean $= 0.342352$

(27) If (CRIM > 0.032519 and ISTAT > 0.268075 and ISTAT ≤ 0.34175 and DIS > 0.1454) mean $= 0.312174$

(28) If (CRIM > 0.032519 and ISTAT > 0.34175 and ISTAT ≤ 0.399695) mean $= 0.3024$

(29) If (TAX ≤ 0.69561 and ISTAT > 0.399695 and ISTAT ≤ 0.442325 and AGE ≤ 0.559735) mean $= 0.382642$

(30) If (TAX ≤ 0.69561 and ISTAT > 0.399695 and ISTAT ≤ 0.442325 and AGE > 0.559735 and CRIM ≤ 0.001228) mean $= 0.281789$

(31) If (TAX ≤ 0.69561 and ISTAT > 0.399695 and ISTAT ≤ 0.442325 and AGE > 0.559735 and CRIM > 0.001228) mean $= 0.328371$

(32) If (TAX ≤ 0.69561 and ISTAT > 0.442325 and ISTAT ≤ 0.47172) mean $= 0.297154$

(33) If (TAX > 0.69561 and ISTAT > 0.399695 and ISTAT ≤ 0.425085) mean $= 0.27226$

(34) If (TAX > 0.69561 and ISTAT > 0.425085 and ISTAT ≤ 0.47172) mean $= 0.247492$

(35) If (ISTAT $>$ 0.47172 and ISTAT $\leq$ 0.5367 and CRIM $\leq$ 0.035231) mean = 0.260931

(36) If (ISTAT $>$ 0.47172 and ISTAT $\leq$ 0.5367 and CRIM $>$ 0.035231) mean = 0.219767

(37) If (ISTAT $>$ 0.5367 and ISTAT $\leq$ 0.59147) mean = 0.193774

(38) If (ISTAT $>$ 0.59147 and ISTAT $\leq$ 0.76918 and CRIM $\leq$ 0.190695 and AGE $\leq$ 0.93512) mean = 0.166575

(39) If (ISTAT $>$ 0.59147 and ISTAT $\leq$ 0.76918 and AGE $>$ 0.93512 and CRIM $\leq$ 0.127715) mean = 0.139309

(40) If (ISTAT $>$ 0.59147 and ISTAT $\leq$ 0.76918 and AGE $>$ 0.93512 and CRIM $>$ 0.127715 and CRIM $\leq$ 0.190695) mean = 0.114915

(41) If (ISTAT $>$ 0.59147 and ISTAT $\leq$ 0.76918 and CRIM $>$ 0.190695) mean = 0.0719827

(42) If (ISTAT $>$ 0.76918 and ISTAT $\leq$ 0.81871) mean = 0.0378593

(43) If (ISTAT $>$ 0.81871) mean = -0.0133432

Forest fires DATASET

(1) If (FFMC $\leq$ 0.204515) mean = -0.089132

(2) If (FFMC $>$ 0.204515 and FFMC $\leq$ 0.75613) mean = -0.000116

(3) If (FFMC $>$ 0.75613 and DC $\leq$ 0.0089715) mean = -7E-006

(4) If (FFMC $>$ 0.75613 and DC $>$ 0.0089715 and ISI $\leq$ 0.341355) mean = 0

(5) If (FFMC $>$ 0.75613 and DC $>$ 0.0089715 and ISI $>$ 0.341355) mean = -6.66667E-007

Pollution DATASET

(1) If (PREC <= 0.13) mean = -0.0407647

(2) If (PREC $>$ 0.13) mean = 0.460172

Table 6.7: Average rules of 10FCV for classification problems

	Gaussian function		Morlet function	
Dataset	DEWNN + DT	DEWNN + RIPPER	DEWNN + DT	DEWNN + RIPPER
Full features				
IRIS	4.1	3.4	4.9	3.3
WINE	8.1	4.5	7.1	4.1
SPANISH	4.5	2.5	3.4	2.5
TURKISH	2	2	2.4	2.1
US	4.9	3.1	5.1	2.9
UK	4.5	2.8	5.1	2.4
Reduced features				
WINE	6.3	4.4	5.1	3.8
SPANISH	2.4	2.3	5.1	2.5
TURKISH	2	2	2.4	2.4
UK	3.6	2.1	2.6	2.4

Tables 6.7 and 6.8 present the rule base size of the hybrids with full feature datasets, reduced feature datasets of the classification and regression problems respectively. From Table 6.7, in the case of full features datasets, it is clearly shown that DEWNN+Ripper achieved less number of rules when Morlet activation function is used in all datasets. In the case of reduced features datasets, from Table 6.7, the average rule base size of DEWNN+Ripper yielded less number of rules when Gaussian activation function is used in all datasets. From Table 6.8, in the case of regression problems, it is observed that DEWNN + DENFIS obtained less number of rules when compared to that of DEWNN + CART no matter which wavelet activation function is used over all datasets. Further, the average rule base size of reduced features datasets from Table 6.8, it is shown that DEWNN + DENFIS yielded less number of rules, when Morlet activation function is used over Body fat dataset, Boston housing dataset, Forest fires dataset. However, for the Auto MPG and pollution datasets, DEWNN + DENFIS achieved less number of rules when Gaussian activation function is used in DEWNN and DEWNN + CART achieved less number of rule base size, when Morlet activation function is used in DEWNN. From the above discussion, it is concluded that we can use both the hybrids for rule extraction purpose.

Further, t-test is performed between the DEWNN and DEWNN + DT and

Table 6.8: Average rules of 10FCV for regression problems

	Gaussian function		Morlet function	
Dataset	DEWNN+CART	DEWNN+DENFIS	DEWNN+CART	DEWNN+DENFIS
Full features				
AUTOMPG	49.4	4.8	43.7	4.8
BODYFAT	19.1	2.9	20.5	2.9
BOSTONHOUSING	56	5	50	5
FORESTFIRES	19.8	9.8	22.1	9.8
POLLUTION	5.1	5.8	4.9	5.8
Reduced features				
AUTOMPG	54.4	2.9	47.8	3.9
BODYFAT	16	3.4	21.3	2.9
BOSTONHOUSING	66	5.6	66	3.3
FORESTFIRES	64.6	6.1	60.8	5.3
POLLUTION	6.6	4.8	4.3	5.8

DEWNN + Ripper with respect to the accuracy for 3 class problems and sensitivity for 2 class problems for classification problems for full features and reduced features of the Gaussian and Morlet function and tabulated in Table 6.9 and 6.10 respectively. From table 6.9 with respect to Gaussian function, in the case of Iris, Wine, spanish banks of the hybrid DEWNN + Ripper, US banks and UK banks datasets both the DEWNN and DEWNN + Rule extraction methods are found to be statistically insignificant at 1% level of significance. In the case of Morlet function Iris, Wine of the hybrid DEWNN + Ripper, US banks and UK banks datasets are statistically insignificant. However, in the case of Gaussian Spanish banks DEWNN is better when t-test is performed between the DEWNN and DEWNN + DT. It is found that in the case of Turkish bank dataset, DEWNN is better when t-test is performed between plain DEWNN and the proposed hybrids. Further, in the case of Morlet function with respect to Wine dataset, DEWNN + DT is better than the DEWNN. Also, in the case of Spanish banks and Turkish banks DEWNN is found to be better than the hybrids when t-test is performed.

In the case of classification problems for reduced features from Table 6.10, Gaussian function wine dataset when t-test performed between DEWNN and DEWNN + DT Spanish banks, UK banks, and in the case of Morlet function Wine, Spanish banks, UK banks the proposed hybrids and plain DEWNN are sta-

Table 6.9: t-test values between DEWNN and Hybrids of classification problems for full features

Dataset	DEWNN Vs DEWNN + DT	DEWNN Vs DEWNN+ Ripper	model DEWNN Vs DEWNN + DT	model DEWNN Vs DEWNN + Ripper
Gaussian function				
iris	0.93	0.80	insignificant	insignificant
wine	0.00	0.37	insignificant	insignificant
spanish	3.24	2.43	DEWNN	insignificant
turkish	7.79	7.79	DEWNN	DEWNN
US	1.99	0.31	insignificant	insignificant
UK	0.92	1.35	insignificant	insignificant
Morlet function				
iris	1.40	0.87	insignificant	insignificant
wine	3.43	2.10	DEWNN + DT	insignificant
spanish	3.58	4.09	DEWNN	DEWNN
turkish	8.57	5.69	DEWNN	DEWNN
US	2.72	2.28	insignificant	insignificant
UK	1.20	1.44	insignificant	insignificant

tistically insignificant in nature at 1% level of significance. However, in the case of Gaussian function, in the case of Turkish banks dataset plain DEWNN and the hybrids, and wine dataset DEWNN and DEWNN + Ripper at 1% level of significance they are statistically significant and DEWNN is better than the hybrids in these cases. Also, Turkish banks dataset 1% level of significance DEWNN is better than the proposed hybrids.

Also, t-test is performed between the DEWNN and DEWNN + CART / DEWNN + DENFIS for regression problems for both Gaussian and Morlet function in the case of the full features and reduced features t-test values are tabulated in Table 6.11 and 6.12. From Table 6.11 and 6.12 in the case of all the datasets except forest fires dataset in reduced features in the case of Gaussian function, in both the Gaussian and Morlet function when full features and reduced features are considered, at 1% level of significance plain DEWNN and hybrids are statistically significant and DEWNN is better than the the proposed hybrids. In the case of forest fires dataset in the case of Gaussian function when reduced features are considered both the plain DEWNN and proposed hybrids DEWNN + CART / DEWNN + DENFIS are statistically insignificant at 1% level of significance.

Table 6.10: t-test values between DEWNN and Hybrids of classification problems for reduced features

Dataset	DEWNN Vs DEWNN + DT	DEWNN Vs DEWNN+ Ripper	model DEWNN Vs DEWNN + DT	model DEWNN Vs DEWNN + Ripper
Gaussian function				
Wine	2.11	2.32	insignificant	insignificant
Spanish	2.71	2.71	insignificant	insignificant
Turkish	9.00	9.00	DEWNN	DEWNN
UK	0.58	0.48	insignificant	insignificant
Morlet function				
Wine	0.11	0.48	insignificant	insignificant
Spanish	2.41	1.86	insignificant	insignificant
Turkish	3.59	5.71	DEWNN	DEWNN
UK	0.29	0.00	insignificant	insignificant

Table 6.11: t-test values between DEWNN and Hybrids of regression problems for full features

Dataset	DEWNN Vs DEWNN+CART	DEWNN Vs DEWNN+DENFIS	Better model DEWNN Vs DEWNN+CART	Better model DEWNN Vs DEWNN+DENFIS
Gaussian function				
AUTOMPG	25.23	25.24	DEWNN	DEWNN
BODYFAT	8.61	8.73	DEWNN	DEWNN
BOSTON HOUSING	13.52	17.69	DEWNN	DEWNN
FORESTFIRES	8.12	7.89	DEWNN	DEWNN
POLLUTION	12.88	13.00	DEWNN	DEWNN
Morlet function				
AUTOMPG	9.92	9.92	DEWNN	DEWNN
BODYFAT	8.71	8.91	DEWNN	DEWNN
BOSTON HOUSING	7.35	8.40	DEWNN	DEWNN
FORESTFIRES	7.41	7.07	DEWNN	DEWNN
POLLUTION	5.83	5.80	DEWNN	DEWNN

Further, we also performed t-test between our proposed hybrids DERBF + GATree and DEWNN + DT / Ripper with respect to classification problems for Gaussian and Morlet functions in DEWNN. The t-test values with respect to Gaussian function and Morlet function in DEWNN are tabulated in Table 6.13.

Table 6.12: t-test values between DEWNN and Hybrids of regression problems for reduced features

Dataset	DEWNN Vs DEWNN+CART	DEWNN Vs DEWNN+DENFIS	Better model DEWNN Vs DEWNN+CART	Better model DEWNN Vs DEWNN+DENFIS
Gaussian function				
AUTOMPG	27.95	27.95	DEWNN	DEWNN
BODYFAT	9.72	9.73	DEWNN	DEWNN
BOSTON HOUSING	14.53	14.55	DEWNN	DEWNN
FORESTFIRES	2.71	2.71	insignificant	insignificant
POLLUTION	11.68	11.35	DEWNN	DEWNN
Morlet function				
AUTOMPG	39.24	39.24	DEWNN	DEWNN
BODYFAT	20.44	20.74	DEWNN	DEWNN
BOSTON HOUSING	22.78	22.88	DEWNN	DEWNN
FORESTFIRES	3.35	3.35	DEWNN	DEWNN
POLLUTION	8.50	8.52	DEWNN	DEWNN

From the Table 6.13 it is found that at 1% level of significance DERBF+GATree hybrid is better than that of DEWNN + DT and DEWNN + Ripper, in the case of Gaussian function in DEWNN in Spanish banks, Turkish banks, UK banks and in the case of Morlet function in DEWNN Spanish banks and Turkish banks. In remaining datasets when Gaussian function is used in DEWNN, DEWNN + DT and DEWNN + Ripper are better in the case of Iris, Wine datasets. However, when Morlet function is used in DEWNN, hybrids of DEWNN are better for Iris, Wine datasets. In reset of the datasets like in Gaussian function used in DEWNN in DEWNN, the hybrids proposed hybrids DERBF+GATree and DEWNN+DT / DEWNN+Ripper are statistically insignificant at 1% level of significance. Also, when Morlet function used in DEWNN, in the case of UK banks and US banks datasets at 1% level of significance the proposed hybrids DERBF+GATree and DEWNN+DT / DEWNN+Ripper are statistically insignificant. From these results, it can be concluded that one can use any of the proposed hybrids for classification purpose by extracting rules from proposed novel architectures.

Table 6.13: t-test values between DERBF + GATree and DEWNN + DT / Ripper of classification problems for full features

Dataset	DERBF+GATree Vs DEWNN + DT	DERBF+GATree Vs DEWNN+ Ripper	Better model DERBF+GATree Vs DEWNN + DT	Better model DERBF+GATree Vs DEWNN + Ripper
Gaussian function				
Iris	9.48	9.85	DEWNN +DT	DEWNN+Ripper
Wine	10.42	9.81	DEWNN +DT	DEWNN+Ripper
Spanish	4.87	3.69	DERBF+GATree	DERBF+GATree
Turkish	13.50	13.50	DERBF+GATree	DERBF+GATree
UK	2.90	3.01	DERBF+GATree	DERBF+GATree
US	0.00	1.01	insignificant	insignificant
Morlet function				
Iris	4.87	5.59	DEWNN +DT	DEWNN+Ripper
Wine	11.81	11.00	DEWNN +DT	DEWNN+Ripper
Spanish	4.95	5.98	DERBF+GATree	DERBF+GATree
Turkish	13.50	8.49	DERBF+GATree	DERBF+GATree
UK	2.72	2.51	insignificant	insignificant
US	0.00	0.27	insignificant	insignificant

6.5 Conclusions

We presented a hybrid method to extract rules from a trained DEWNN using DT, Ripper to solve classification problems viz., Spanish banks, Turkish banks, UK banks and US banks and Iris, Wine dataset. Further, we also presented the hybrid method to extract rules in solving regression problems using CART, DENFIS namely, Auto MPG, Body Fat, Boston Housing, Forest Fires and Pollution. The average rule base size obtained by the proposed hybrids turned out to be smaller with feature selection. Further, based on t-test, it is inferred that the hybrids not only make the DEWNN transparent but also are not very far from standalone DEWNN in terms of sensitivity in classification problems. In the case of regression problems, the standalone DEWNN outscored the hybrids in terms of MSE values.

Part IV - Optimization Technique

Chapter 7

Rule Extraction from Firefly Optimization Algorithm for solving Classification problems

This Chapter presents the proposed rule mining algorithm based on firefly optimization and named it as firefly miner which is one of the main contributions of this thesis. The main objective of firefly miner is to extract classification rules from a given dataset. First section presents introduction followed by the firefly miner algorithm in second section. Dataset analyzed by firelfy miner are presented in section three. Section four presents the results and discussions. Final section concludes the chapter

7.1 Introduction

Data mining is the extraction of non-obvious, hidden and actionable knowledge from massive datasets. The knowledge extracted is useful in providing decision support in many disciplines including service industries such as banking, insurance, telecom etc, primarily, such knowledge is represented in the form of if-then rules and decision trees Quinlan [Quinlan, 1986] are classical examples of such constructs. Essentially, these rules solve classification problems arising in these areas. The rule antecedent part consists of features with conditions which are connected with logical operator and. The consequent part signifies the class label.

In general the rule is represented as

$$\text{If feature}_1 \ <\text{value}> \text{ and feature}_2 \ <\text{value}> \text{ and feature}_n<\text{value}> \text{ Then Class} \tag{7.1}$$

7.2 Firefly Miner

We proposed firefly based rule extraction algorithm for classification purpose and named it as firefly miner. We followed a sequential covering approach followed by [DeSousa et al., 2003], which generates a single rule in each run. The miner runs the firefly algorithm at least as many times as the number of classes available in the dataset. Initially, the rule list is empty. Generate population size number of rules consisting of all features, where each feature value lies between its lower and upper bounds in the training dataset. The logical operator ¿ or ¡ for each feature in the rules was chosen randomly. Compute the fitness function for each of the firefly. After one run of firefly, it outputs a rule with the highest quality. Then, we remove the samples, which are correctly classified by that rule, from the training set and run the firefly miner again for the same class until the number of uncovered samples for that class is less than a pre-specified threshold. This process is repeated for the remaining classes also. Finally, we end up with a set of rules, which can be used to classify the new dataset or patterns. Here, we followed the Michigan approach, where, each individual encodes a single rule. The pseudo code 1 of the firefly rule miner is presented below.

RS = /* initially, Rule Set is empty */;

for *Each Class C* **do**

TS = All training examples;

while *(Number of uncovered training examples belonging to class C > MaxUncovExampPerClass)* **do**

Run the Firefly algorithm to discover the predicting class C and return the best discovered rule BestRule;

RS = RS U BestRule;

TS = TS training examples correctly covered by the discovered rule;

end

end

Algorithm 1: Pseudo code of firefly miner

In rule classifier systems, there are two distinct approaches to individual or particle representation (i) the Michigan and (ii) the Pittsburgh [Lopes et al., 1997]. In the Michigan approach, each individual encodes a single rule, whereas in the Pittsburgh approach each individual encodes a set of rules. In our work, we followed the Michigan approach. The rule representation is as follows:

$$\text{If feature}_1 < or > \text{value and feature}_2\text{<or> value and feature}_n\text{<or> value Then Class} \tag{7.2}$$

7.2.1 Fitness function

The fitness function for the firefly miner is classification accuracy given by the firefly. For the firefly miner the fitness function is evaluated as follows:

Let, A = No. of records in the dataset where both antecedent and consequentmatch;

B = No. of records in the dataset where, antecedent doesnt match and consequentmatches;

C = No. of records in the dataset where, antecedent matches and consequentdoesnt;

D = No. of records in the dataset where, both antecedent and consequent doesntmatch;

$$Fitness = \frac{A}{A+B} * \frac{D}{C+D} \tag{7.3}$$

This formula penalizes each firefly, which moves out of boundary values, assigning it with negative value (-1.0), and forcing it to be in the search space [DeSousa et al., 2003].

7.3 Datasets Analyzed

We chose publicly available benchmark datasets such as IRIS, WINE and WBC datasets.The effectiveness of the proposed model is tested over classification datasets of bank bankruptcy datasets such as Spanish banks [Olmeda and Fernndez, 1997], Turkish banks [Canbas et al., 2005], UK banks [Beynon and Peel, 2001] and US banks [Rahimian et al., 1996]. Information about the classification datasets analyzed is presented in Appendix C, out of the listed datasets in Table 1.1 for demonstration purpose.

Rules are generated for each of the 10 folds in the 10-fold cross validation method using 80% training data. Generated rules are then tested against the validation set and the results are presented in Results and discussions section. Prediction accuracy of the rules is determined in terms of accuracy for classification problems.

7.4 Results and Discussions

We developed the Firefly-miner in Java JDK 1.6 in Windows 7 environment on a desktop with 3 GB RAM. In our work, we used 8 datasets for testing the effectiveness of the algorithm. The datasets Iris, Wine and WBC, datasets are taken from the UCI machine learning repository (`http://archive.ics.uci.edu/ml/datasets.html`).

In our experimental methodology, we first divided every dataset into two parts: 80% for training and 20% for validation by following stratified random sampling. Then, 10-Fold cross validation (10-FCV) is performed on the training set and the validation dataset is kept aside for evaluating the efficiency of the rules generated in the 10-FCV. Further, given the evolutionary nature of the algorithm, for each fold, we repeated the experiment 10 times by using different initial random seeds. This process resulted in 100 experiments for 10 folds. Accordingly, we presented the average results in terms of sensitivity, specificity and accuracy and Area under ROC Curve (AUC) in for firefly miner and DT Tables 1 and 2 respectively. Further, the population size and attractiveness parameter β_0 in the firefly miner were chosen as 100 and 1 respectively.

From Tables 7.1 and 7.2, we observe that in the case of WBC and all banks datasets except US banks, firefly miner yielded a sensitivity of 91.34%, 99.58%, 68.16% and 63.89% respectively thereby outperforming DT. In the case of US banks, DT outperformed firefly miner by producing a sensitivity of 93.07%. In the case of Iris, firefly miner outperformed DT by producing an accuracy of 95.87%. However, in the case of Wine, DT outperformed firefly miner with an accuracy of 97.42% The average number of rules produced by firefly miner and DT is also presented in Tables 7.1 and 7.2 respectively. When the rule base size is considered, the firefly miner turned out to be better than the DT. Rules produced by the firefly miner for all the datasets are presented below.

Table 7.1: Average results of firefly miner

Dataset	Sens*	Spec*	Acc*	No. rules	AUC
Iris	NA	NA	95.87	3.05	NA
Wine	NA	NA	86.21	8.87	NA
WBC	91.34	94.47	93.67	5.9	9290
Spanish	99.58	86.74	91.36	2.2	9316
Turkish	68.16	77.48	77.75	4.96	7282
US	90.76	99.88	94.70	2.01	9532
UK	63.89	80.52	73.43	3.44	7221

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

The rules generated by firefly miner are presented below:

IRIS DATASET

(1) If Sepal length>6.8 and Sepal width>4.4 and Petal length>5.1 and Petal width>0.8 Then Class=IRIS-SETOSA

(2) If Sepal length<4.8 and Sepal width>3.5 and Petal length<2.6 and Petal

Table 7.2: Average results of Decision Tree

Dataset	Sens*	Spec*	Acc*	No. rules	AUC
Iris	NA	NA	92.66	4.8	NA
Wine	NA	NA	97.42	5.8	NA
WBC	31.11	91.7	71.19	2.5	6140
Spanish	71.66	100	86.91	3.7	8583
Turkish	55	100	77.50	2	7750
US	93.07	85.37	89.22	5.7	8922
UK	48.36	73.33	60.83	4.8	6084

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

width>1.7 Then Class=VERSICOLOR

(3) If Sepal length>7.9 and Sepal width>3.9 and Petal length<4.7and Petal width<1.0 Then Class=VIRGINICA

WINE DATSET

(1) If Alcohol>13.8 and Malic acid<1.3 and Ash<1.8 and Alcalinity of ash<14.1 and Magnesium>159.6 and Total phenols<1.2 and Flavanoids>3.6 and Non-flavanoid phenols<0.1and Proanthocyanins>1.3 and Color intensity<2.2and Hue>0.7 and OD280/OD315 of diluted wines>3.9and Proline>1014.0 Then Class=C

(2) If Alcohol<11.8 and Malic acid<1.5 and Ash>3.1 and Alcalinity of ash>25.9 and Magnesium>119.3 and Total phenols>3.5 and Flavanoids<0.6 and on-flavanoid phenols>0.6 and Proanthocyanins<0.6 and Color intensity>8.8 and Hue<0.9and OD280/OD315 of diluted wines<1.4 and Proline<307.6 Then Class=A

(3) IfAlcohol>14.5 and Malic acid<0.9 and Ash>2.4 and Alcalinity of ash<11.7 and Magnesium>104.3 and Total phenols>2.8 and Flavanoids>1.5 and Non-flavanoid phenols<0.3 and Proanthocyanins>2.2 and Color intensity>11.8 and Hue<0.and OD280/OD315 of diluted wines<1.2 and Proline<538.1 Then Class=C

(4) If Alcohol<12.3 and Malic acid<2.1 and Ash<2.1 and Alcalinity of ash>26.7

and Magnesium<75.4 and Total phenols>2.9 and Flavanoids>3.7 and Non-flavanoid phenols>0.6 and Proanthocyanins>1.6 and Color intensity<3.6 and Hue<0.5and OD280/OD315 of diluted wines>3.7 and Proline>1630.2 Then Class=C

(5) If Alcohol<12.1 and Malic acid>5.1 and Ash>2.7 and Alcalinity of ash>24.6 and Magnesium>133.7 and Total phenols>3.5 and Flavanoids<2.0 and Non-flavanoid phenols<0.1 and Proanthocyanins>2.3 and Color intensity>6.2 and Hue>1.3 and OD280/OD315 of diluted wines>3.5 and Proline>1211.4 Then Class=A

(6) If Alcohol>13.5 and Malic acid>3.8 and Ash<1.4 and Alcalinity of ash<12.4 and Magnesium<73.1 and Total phenols<1.4 and Flavanoids<0.6 andNonflavanoid phenols < 0.1 and Proanthocyanins>2.2 and Color intensity>7.6 and Hue<0.9 and OD280/OD315 of diluted wines>3.1 and Proline>1650.8 Then Class=B

(7) If Alcohol<11.9 and Malic acid<1.0 and Ash>2.6 and Alcalinity of ash>25.3 and Magnesium>156.1 and Total phenols>3.1 and Flavanoids>2.6 and Non-flavanoid phenols>0.4 and Proanthocyanins<0.8 and Color intensity>7.7 and Hue>1.2and OD280/OD315 of diluted wines<2.3 and Proline>1089.5 Then Class=B

(8) If Alcohol<11.3 and Malic acid<1.0and Ash>3.2 and Alcalinity of ash>26.1 and Magnesium<72.9 and Total phenols>3.5 and Flavanoids>3.3 and Non-flavanoid phenols<0.3 and Proanthocyanins>1.8 and Color intensity>6.1 and Hue<0.5and OD280/OD315 of diluted wines<2.5 and Proline>794.4 Then Class=B

WBC DATASET

(1) If Clump Thickness>6.6 and Uniformity of Cell Size>5.0 and Uniformity of Shape>9.5 and Marginal Adhesion>4.0 and Single Epithelial Cell Size>7.7and Bare Nuclei>7.5 and Bland Chromatin>5.7 and Normal Nucleoli>3.7 and Mitoses>8. Then Benign

(2) If Clump Thickness<3.1 band Uniformity of Cell Size>10.0 and Uniformity of Shape<1.1 and Marginal Adhesion>7.1 and Single Epithelial Cell Size<1.8

and Bare Nuclei<4.9 and Bland Chromatin<2.6 and Normal Nucleoli>9.1 and Mitoses>5.6 Then Malignant

(3) If Clump Thickness<4.7 and Uniformity of Cell Size<4.7 and Uniformity of Shape<3.3 and Marginal Adhesion<2.0 and Single Epithelial Cell Size<2.1 and Bare Nuclei<1.0 and Bland Chromatin<4.4 and Normal Nucleoli<1.9 and Mitoses>8.9 Then Malignant

(4) If Clump Thickness<1.1 and Uniformity of Cell Size<2.7 and Uniformity of Shape>9.4 and Marginal Adhesion<1.0 and Single Epithelial Cell Size<2.4 and Bare Nuclei<1.8 and Bland Chromatin>10.0 and Normal Nucleoli<5.7 and Mitoses>3.4 Then Malignant

(5) If Clump Thickness>6.2 and Uniformity of Cell Size>7.4 and Uniformity of Shape>7.8 and Marginal Adhesion>7.0 and Single Epithelial Cell Size<1.0 and Bare Nuclei<1.07 and Bland Chromatin<1.0 and Normal Nucleoli>5.8 and Mitoses<0.19 Then Malignant

SPANISH DATASET

(1) If CA/TA >0.7 and CAC/TA s<0.08 and CA/L <0.04 and R/L >0.08 and NI/TA <-0.06 and NI/TEC <-1.74 and NI/L <-0.019 and CS/S >1.1 and CF/L < 0.009Then NonBankrupt

(2) If CA/TA >0.66 and CAC/TA >0.49 and CA/L >0.6 and R/L <0.001 and NI/TA >0.022 and NI/TEC <-1.35 and NI/L >0.02 and CS/S <0.69 and CF/L >0.008 Then Bankrupt

TURKISH DATASET

(1) If IE/APA >21.8 and IE/ANA >27.4 and (SE+TI)/(D+NF) >40.3 and II+IE <144.7 and (SE+TI)/(TA+CC) <-12.0 and (SE+TI)/(TA+CC) <-102.8 and NC/TA <-11.2 and (SEB+RR)/P <7.7 and LA/(D+NF) >74.9 and IE/TE <39.2 and LA/TA <19.2 and SCR >85.2 Then NonBankrupt

(2) If IE/APA >41.4 and IE/ANA <11.5 and (SE+TI)/(D+NF) <3.3 and II+IE >208.5 and (SE+TI)/(TA+CC) <-3.9 and (SE+TI)/(TA+CC) <-54.1 and NC/TA >25.0 and (SEB+RR)/P >32.5 andLA/(D+NF) >91.7 and IE/TE >81.5 and LA/TA >86.4 and SCR <-10.4 Then Bankrupt

(3) If IE/APA >69.4 and IE/ANA >25.4 and (SE+TI)/(D+NF) <-47.8 and II+IE >316.5 and (SE+TI)/(TA+CC) >14.0 and (SE+TI)/(TA+CC) >15.1 and NC/TA >-1.9 and (SEB+RR)/P <7.4 and LA/(D+NF) <34.5 and IE/TE <61.3 and LA/TA <31.0 and SCR >89.6 Then Bankrupt

US DATASET

(1) If WC/TA <-0.53 and RE/TA >0.22 and EIT/TA >0.14 and ME/TA >33.0 and S/TA >3.7 Then Bankrupt

(2) If WC/TA <-0.1 and RE/TA <0.16 and EIT/TA <0.04 and ME/TA >19.2 and S/TA >5.79 Then NonBankrupt

UK DATASET

(1) If Sales>100557.2 and PBT/C>23.9 and FF/TL<-0.2 and (CL + LTD)/TA>3.0 and CL/TA<0.4 and CA/CL>4.1 and (CA + S)/CL>1.3and (CA + CL)/TA>0.4 and LAG <199.2 and AGE>51.5 Then NonBankrupt

(2) If Sales <2860.8 and PBT/C>17.4 and FF/TL>0.3 and (CL + LTD)/TA>2.4 and CL/TA>1.4 and CA/CL>2.8 and (CA + S)/CL>2.7and (CA + CL)/TA>0.4 and LAG <243.5 and AGE<12.6 Then Bankrupt

(3) If Sales >102976.8 and PBT/C<-33.4 and FF/TL<-0.1 and (CL + LTD)/TA>1.2 and CL/TA>1.3 and CA/CL>4.04 and (CA + S)/CL>2.9 and (CA + CL)/TA>0.2 and LAG >264.5and AGE>53.4 Then Bankrupt

7.5 Conclusions

We developed firefly miner for extracting rules from a classification dataset. The effectiveness of the proposed rule miner is demonstrated on various datasets taken from the financial domain viz., bankruptcy prediction, credit scoring and benchmark problems taken from UCI machine learning repository. The proposed miner outperformed the baseline DT in terms of the sensitivity and rule base size across most of the datasets. From the results, it is observed that the results are encouraging and it can be used as a viable alternative to decision trees for classification purpose. The future directions include making necessary modification to the design of the algorithm so that it can solve classification problems with a mixture of numerical and categorical features.

As regards future directions, the length of a rule can be reduced by introducing feature selection concept for the proposed firefly miner. Also, the logical operators $\leq$, $==$ and $\geq$ can be added to the existing operators. Further, in the initial generation the rules pertaining to all classes can be taken and the highest coverage will be considered in the initial and succeeding generations.

Chapter 8

Overall Conclusions

Data mining applications are becoming very much common in every domain like finance, biology, manufacturing, etc. Artificial Neural Networks (ANN) yield outstanding performance in prediction, by learning from the past experience (data) and generalizing on the new data. In general, ANN achieves very high accuracies. An important drawback, however, of the neural networks is *black box* stigma. It means that the knowledge learned by the neural network cannot be comprehended by human beings because the neural network (NN) simply does not output it. To overcome this disadvantage many researchers proposed alternative methods to extract knowledge from trained NN, the form *If then* rules. Researchers also started using optimization problems in solving the classification and regression problems. *eclectic* and using *optimization algorithms* in solving classification and regression problems.

The present study found the efficiency of the proposed rule extraction approaches in solving the bankruptcy prediction in banks. Bankruptcy prediction in banks and financial firms are the most researched areas in banking and finance domain. Bank management is very much interested in algorithms used for predictions. We extracted crisp and fuzzy rules from trained neural networks using GATree, Decision Tree and Ripper, and crisp rules from firefly optimization algorithm.

- For the last few decades, business, finance and medical sectors are more concerned about the privacy of the data. This very concern for the privacy of the data threw up new challenges for business decision making based on data mining. We proposed a new 3-layered auto-associative neural network architecture for privacy preservation. Also, we proposed a new feature selection method for the proposed architecture. Finally, rules are extracted from the proposed architecture in solving the classification and regression problems by using DT, Ripper and CART, DENFIS respectively.

- Differential evolution algorithm is used to train the radial basis function network and extracted rules from the new architecture using GATree. In this, we addressed classification problems relates to bankruptcy prediction.

- Churn prediction problem in banks decision makers, are more conscious about their quality of service they provide to the customer because of increase in the attrition or churn of customers day-by-day. Loyal customers defecting from one bank to another has become common. This trend, called churn which occurs due to lack of latest technology, unease of utility of the technology, customer friendly staff, propinquity of geographical locations, etc. Thus, there is a vital need in developing a model which can predict the probability that the today's loyal customer is going to churn out in near future. Churn prediction is one of the most vital activities in customer relationship management. Once potential churners are identified, management employs anti-churn strategies that are less expensive that acquiring new customers. This thesis addressed the churn prediction in banks problem. GMDH neural network is used as a black box and decision tree is used to extract rules to remove the black box stigma of the neural network.

- Rules are extracted for classification and regression problems from differential evolution trained wavelet neural network (DEWNN) using DT, Ripper and CART, DENFIS respectively. In DEWNN we thoroughly exploited both the Gaussian and Morlet wavelet functions.

- Further, we also proposed rule extraction method using firefly optimization algorithm. In recent years, many researchers are using the optimization algorithms in extracting *If then* rules direclty from the dataset and from ANN. This thesis addressed the classification problems using the firefly miner.

Overall, in this thesis various approaches to extract the knowledge of ANN in the form of *If then* rules are presented in all categories of rule extraction methods namely *decompositional*, *pedagogical* and using optimization algorithm. We have analyzed the efficiency of various rule generating algorithms in solving the classification and regression problems for extracting *If then* rules namely Decision Tree, Ripper, GATree, CART, DENFIS and Firefly miner. As the problems solved in banking and finance domain, management would be very much interested to have transparent model to understand the behavior of the customer and the financial health of a bank. The management of the bank can use the rules extracted by the models presented in this thesis as an early warning expert system to avoid heavy losses to the organization.

Future enhancement would be, (i) extraction of fuzzy rules by using the parameters of neural networks, (ii) extraction of association rules from the PSOAANN architecture, (iii) improving the firefly miner by desiging new objective functions and also extracting rules of different classes from the population at a time. Also, working in different applications of banking, finance, bio-informatics, etc.

Appendix A

Overview of Optimzation Algorithms

This section provides the detailed overview of optimization techniques used to update the weights of the neural networks. The techniques are Particle Swarm Optimization, Differential Evolution and Firefly Optimization algorithm.

A.1 Particle Swarm Optimization (PSO)

The PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995. Development of PSO was based on observations of the social behavior of animals such as bird flocking, fish schooling, and swarm theory. Like the GA, the PSO is initialized with a population of random solutions. It also requires only the information about the fitness values of the individuals in the population. This differs from many optimization methods requiring the derivation information or the complete knowledge of the problem structure and parameter. Compared with the GA, the PSO has memory so that the information of good solutions is retained by all individuals. Furthermore, it has constructive cooperation between individuals, individuals in the population share information between them.

The PSO algorithm [Kennedy and Eberhart, 1995] is a population based optimization technique. This algorithm mimics the behavior of bird flocking, fish schooling or the social behavior of group of people. Each individual is considered to be a dimensionless particle i.e., a point in the N dimensional space. The procedure of PSO consists of initialization and velocity update. In initialization phase, randomly generate a population of solutions called particles with each particle assigned with random velocity V_{id}^{old}. The neighborhood best p_{id} is the best path traveled by each of the particle. The global best p_{gd} is the best path from the entire population. In velocity updation, each particle's velocity is dynamically adjusted with respect to its position x_{id}^{old} using neighborhood best or global best particle. The velocity V_{id}^{new} and position x_{id}^{new} of each particle are updated by following the equations:

$$V_{id}^{new} = w * V_{id}^{old} + c_1 * rand * (p_{id} - x_{id}^{old}) + c_2 * rand * (p_{gd} - x_{id}^{old}) \tag{A.1}$$

$$x_{id}^{new} = x_{id}^{old} + V_{id}^{new} \tag{A.2}$$

Where c_1 and c_2 are two predefined positive constants (usually $c_1 = c_2 = 2$) and represent the weighting of the stochastic acceleration terms that pull each particle toward p_{id} and p_{gd} positions. w is the inertia weight value, which is continuously decreased as the iterations pass, $rand$ is a random number generated from uniform distribution $U(0, 1)$. Flowchart of the PSO algorithm is shown in Figure A.1. The inertia weight w is considered critical for the convergence behavior of particles. It controls the impact of the previous velocities and the current velocities. Further it regulated the trade-off between the local and global search abilities of the particles. Large values of inertia facilitates the global search space and small values facilitates the local search space. A inertia weight values must be reduced with the increase of iterations in order to locate the optimum solution. Initially it is take as 1. If the inertia weight values is set large initially then it search the solutions globally and gradually decrease to get more refined solutions in the search space.

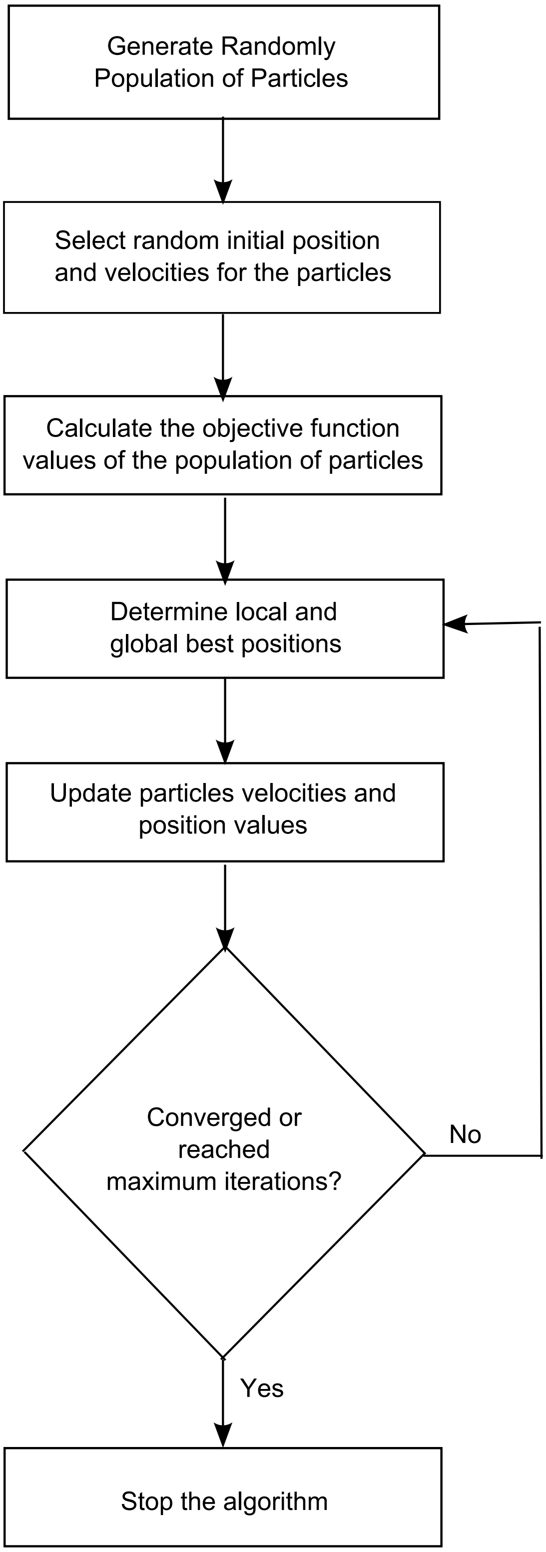

Figure A.1: Flow chart of Particle Swarm Optimization

A.2 Differential Evolution

Differential evolution is a novel approach in evolutionary algorithms. It was proposed by Storn and Price [Storn and Price, 1997]. It is a stochastic, population-based optimization method. Differential evolution algorithm consists mainly of four steps: *initialization*, *mutation*, *recombination* and *selection*. DE differs from other population-based techniques in that it employs differential mutation.

In a population of solutions within an n-dimensional search space, a fixed number of vectors are randomly initialized, then evolved over time to explore the search space and to locate the minima of the objective function. Inside a generation, new vectors are generated by the combination of vectors randomly chosen from the current population (mutation). The vectors so generated are then mixed with a predetermined target vector. This operation is called recombination and produces the trial vector. Finally, the trial vector is accepted for the next generation if and only if it yields a reduction in the value of the objective function. This last operator is referred to as selection.

In each generation, a population P consists of N such R vectors, where N is the size of the population. Thus, $P = (R_1, ..., R_N)$.

It must be noted that the initial population is randomly chosen by using the user specified lower and upper bounds of weights. Accordingly,

$$R_i = R_{imin} + rand(0,1) * (R_{imax} - R_{imin}) \tag{A.3}$$

where $i = 1...N$.

Mutation is a search technique, which together with recombination and selection determines optimal solution. In this step, three distinct target vectors R_a, R_b, R_c are randomly chosen from N population vectors on the basis of three random numbers a, b and c chosen between 1 and N. One of the vectors of the three vectors, for example R_a, is taken as the base of the mutated vector. To this vector, the weighted difference of the remaining two vectors R_b, R_c is added to generate the noisy random vector n_i.

$$n_i = R_a + F * (R_b - R_c) \tag{A.4}$$

where $i = 1...N$. F is the scaling factor which is supplied by the user. This mutation process is repeated to create mate for each of the solution in the parent population. Mutation helps in search of the solution space in each of the dimension.

In the recombination (crossover) operation, target vectors of the parent population are allowed to mate with mutated vector n_i. The vector R_i is recombined with the noisy vector to generate a trail vector t_i. Each element in the trail vector is determined by a binomial experiment, whose success or failure is determined by the user supplied crossover factor CR. CR is used to control the crossover rate at which it takes place.

$$t_i = \begin{cases} n_i & \text{If } rand(0,1) < CR \\ R_i & \text{otherwise} \end{cases} \tag{A.5}$$

where $i = 1...N$. Hence, t_i is the child of two parent vectors n_i and R_i.

If the trail vector violates the upper bound, the error is calculated by subtracting the upper bound from the trail vector. Then the difference between the upper bound and error is taken as new trail vector. If the new trail vector violates the lower bound, then trail vector is regenerated. If the trail vector violates the lower and upper bound, it is brought back within bounds by using the equation [Bhat et al., 2006]

$$t_i = R_{imin} + 2.0 * \frac{p}{q} * (R_{imax} R_{imin}) \tag{A.6}$$

if $t_i > R_{imax}$, with $p = t_i - R_{imax}$, $q = t_i - R_{imin}$ and if $t_i < R_{imin}$, with $p = R_{imin} - t_i$, $q = R_{imax} - t_i$ respectively. The trail vector and target vector proceed to next generation. The vector having the minimum objective function value will get to the next generation. After N competitions in each generation, one will have a new population. This process is repeated until the termination condition reached i.e., maximum number of generations and/or the difference of Error over successive generations is less than or equal to a pre-specified small value. Finally, we will get the final population of weights, which will be used in predicting the output variable. Flowchart of the differential evolution algorithm is shown in Figure A.2.

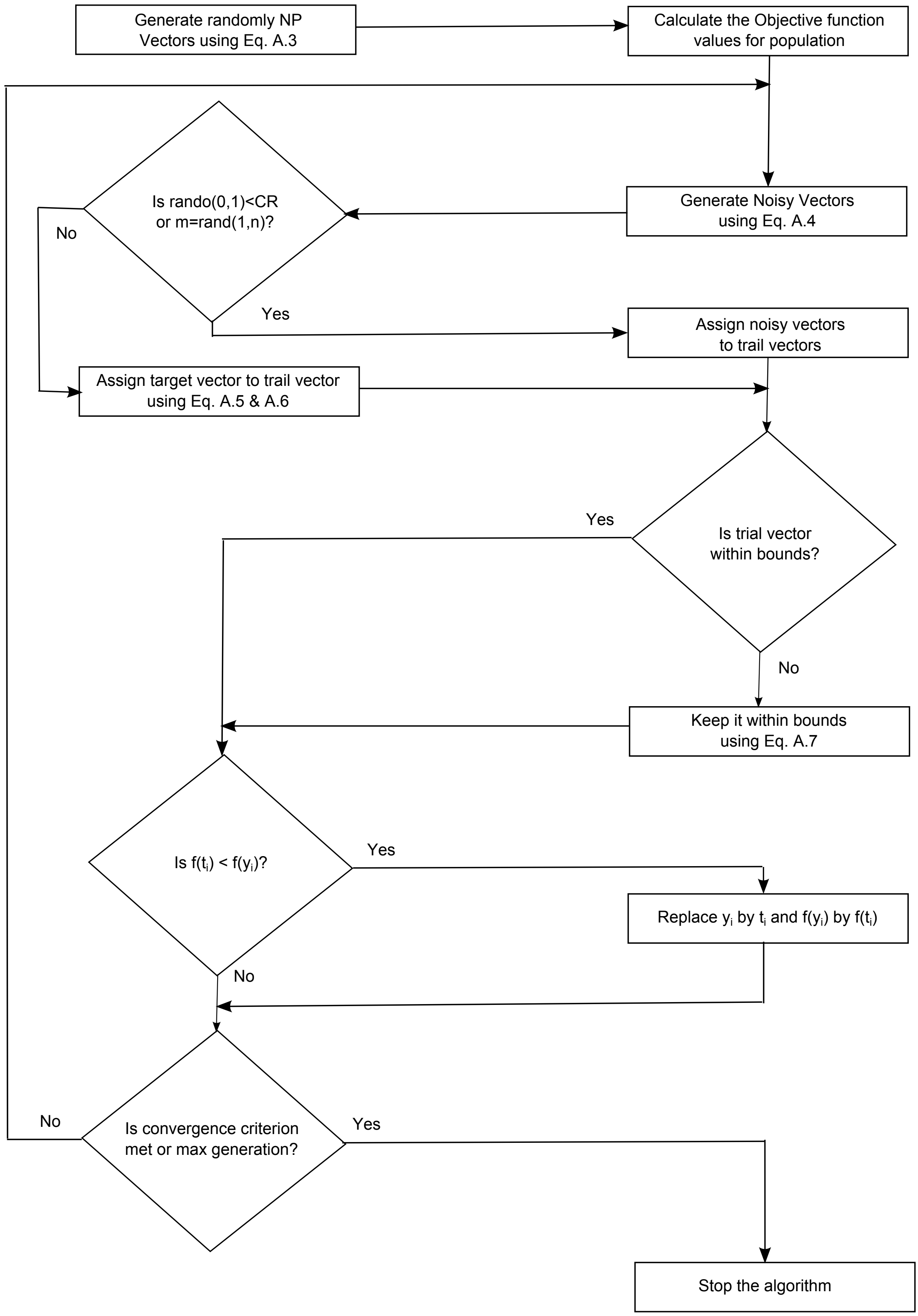

Figure A.2: Flow chart of Differential Evolution

A.3 Firefly Optimization

Firefly algorithm was proposed by Yang [Yang, 2009] [Yang, 2010] based on the nature inspired Fireflies are interesting and night luminous insects that reside mainly in tropical and temperate areas. The flashing light of fireflies is an amazing sight in the summer sky in the tropical and temperate regions. There are about two thousand firefly species, and most fireflies produce short and rhythmic flashes. The pattern of flashes is often unique for a particular species. The light is produced in them by a process of bio-luminescence, and the true functions of such signaling systems are still debating. However, The two fundamental functions of producing flashes in the night time are to attract mating partners and attract the prey. In addition, flashing may also serve as a shielding mechanism from the predators. The rhythmic flash, the rate of flashing and the amount of time form part of the signal system that brings both sexes together. Females respond to a male's unique pattern of flashing in the same species, while in some species such as photuris, female fireflies can mimic the mating flashing pattern of other species so as to lure and eat the male fireflies who may mistake the flashes as a potential suitable mate.

The flashing light is formulated with an objection function to be optimized which in turn formulates new optimization algorithm. Mechanisms of firefly communication via luminescent flashes and their synchronization have been imitated effectively in various techniques of wireless network design [Leidenfrost and Elmenreich, 2008] mobile robots [Krishnanand and Ghose, 2006] and dynamic market pricing [Jumadinova and Dasgupta, 2008]. The light intensity at a particular distance r from the light source obeys the inverse square law. That is to say, the light intensity I decreases as the distance r increases in terms of $I \propto \frac{1}{r^2}$. Furthermore, the air absorbs light which becomes weaker and weaker as the distance increases. These two combined factors make most fireflies visible only to a limited distance, usually several hundred meters at night, which is usually good enough for fireflies to communicate. The flashing light can be formulated in such a way that it is associated with the objective function to be optimized, which makes it possible to formulate new optimization algorithms. In the rest of this paper, we will first outline the basic formulation of the Firefly Algorithm and then discuss the implementation as well as its analysis in detail.

The Firefly algorithm is presented in below.

A.3.1 Algorithm

(1) Generate initial population of fireflies and placed at random positions within the n-dimensional search space x_i. Define the light absorption coefficient ζ.

(2) Define the light intensity of each firefly as the value of cost function for x_i.

(3) For each firefly x_i compare its light intensity L_i with the light intensity L_i if every other firefly x_i.

(4) If L_i is less than L_j then move firefly x_i towards x_j in n dimensions. The value of attractiveness β_0 between flies varies with the distance between them r.

$$x_i = x_i + \beta_0 e^{-\zeta r_{ij}^2}(x_j - x_i) \tag{A.7}$$

(5) Calculate the new values for the cost function of each fly x_i and update the light intensity L_i.

(6) Rank the fireflies and determine the current best.

(7) Repeat step 2 to 6 until definite termination conditions are met, such as predefined number of iterations or a failure to make progress for a fixed number of iterations.

Flowchart of the firefly algorithm is shown in Figure A.3.

A.3.2 Attractiveness

There are two major problems with attractiveness, namely (i) the variation of light intensity and (ii) formulation [Yang, 2009] [Yang, 2010]. For ease of understanding, the attractiveness of a firefly is determined by using its brightness which in turn is associated with the encoded objective function. The brightness I of a firefly at a particular location x can be chosen as $I(x) \propto f(x)$. However, the attractiveness β is relative, it should be seen in the eyes other fireflies. Thus, it will vary

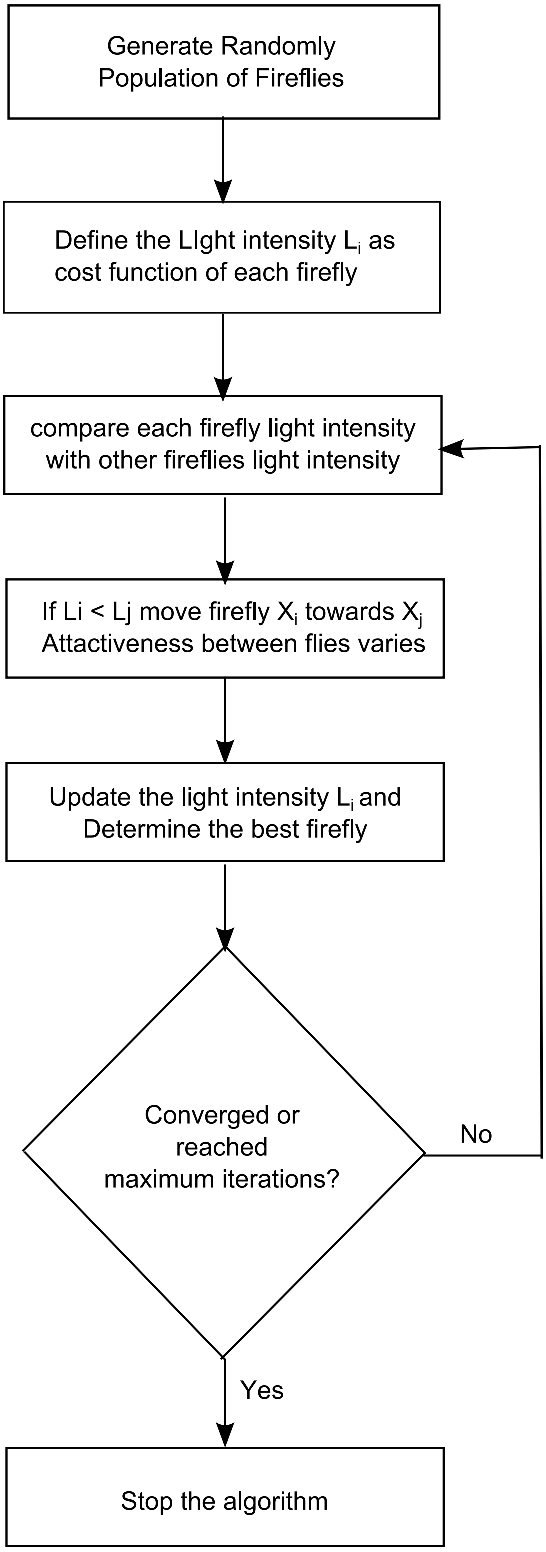

Figure A.3: Flow chart of Firefly Algorithm

with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases when the distance increase from the source i.e., light intensity is inversely proportional to the attractiveness, light is absorbed in the media, so we should allow the attractiveness to vary with the degree of absorption. In other words, the light intensity $I(r)$ varies according to the inverse square law $I(r) = I_s/r^2$ where I_s is the intensity at the source. For a given medium with a fixed light absorption coefficient , the light intensity I varies with the distance r. That is $I = I_0 e^{\zeta r}$, where I_0 is the original light intensity. In order to avoid the singularity at $r = 0$ in the expression I_s/r^2, the combined effect of both the inverse square law and absorption can be approximated using the following Gaussian form

$$I(r) = I_0 e^{-\zeta r^2} \tag{A.8}$$

Sometimes, we may require a function which decreases monotonically at a slower rate. In this case, we can use the following approximation.

$$I(r) = \frac{I_0}{1 + \zeta r^2} \tag{A.9}$$

At a shorter distance, the above two equations are fundamentally same. This is because the series of expansions about $r = 0$.

$$e^{-\zeta r^2} \approx 1 - \zeta r^2 + \frac{1}{2}\zeta^2 r^4 + \ldots \frac{1}{1 + \zeta r^2} \approx 1 - \zeta r^2 + \zeta^2 r^4 + \ldots \tag{A.10}$$

are equivalent to each other up to the order of $O(r^3)$. As a firefly's attractiveness is directly proportional to the light intensity seen by adjacent fireflies, the attractiveness β of a firefly is given by by

$$\beta(r) = \beta_0 e^{-\zeta r^2} \tag{A.11}$$

whereβ_0 is the attractiveness at $r = 0$. As it is often faster to calculate $\frac{1}{(1+r2)}$ than an exponential function, the above function can conveniently be replaced by $\beta = \frac{\beta_0}{1+\zeta r^2}$. The above equation defines a characteristic distance $\Gamma = \frac{1}{\sqrt{\zeta}}$ over which the attractiveness changes significantly from β_0 to $\beta_0 e^{-1}$. In the implementation, the attractiveness function $\beta(r)$ can be any monotonically decreasing functions such as the following generalized equation form

$$\beta(r) = \beta_0 e^{-\zeta r^m}, (m \geq 1). \tag{A.12}$$

For a fixed ζ, the characteristic length becomes $\Gamma = \zeta^{-\frac{1}{m}} \rightarrow 1$ as $m \rightarrow \infty$. Conversely, for a given length scale Γ in an optimization problem, the parameter ζ can be used as a typical initial value. That is $\zeta = \frac{1}{\Gamma^m}$.

A.3.3 Distance and Movement

The distance between any two fireflies i and j at X_i and X_j , respectively, is the Cartesian distance [Yang, 2009] [Yang, 2010]

$$r_{ij} = \| X_i - X_j \| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2}, \tag{A.13}$$

where $x_{i,k}$ is the kth component of the spatial coordinate X_i of ith firefly. In 2-D case, we have $r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. The movement of a firefly i is attracted to another firefly with more attractiveness (brighter) firefly j which is determined by

$$X_i = X_i + \beta_0 e^{-\zeta r_{ij}^2}(X_j - X_i) + \alpha(rand - \frac{1}{2}), \tag{A.14}$$

The second term in the above equation is due to the attraction, while the third term is randomization with α being the randomization parameter. $rand$ is a random number generator between [0, 1]. For most cases in our implementation, we can take $\beta_0 = 1$ and $\alpha \in [0, 1]$. Furthermore, the randomization term can easily be extended to a normal distribution $N(0, 1)$ or other distributions. The parameter ζ now characterizes the variation of the attractiveness, and its value is crucially important in determining the speed of the convergence and how the Firefly algorithm behaves. In theory, $\zeta \in [0,\infty)$, but in practice, $\zeta = O(1)$ is determined by the characteristic length Γ of the system to be optimized. Thus, in most applications, it typically varies from 0.01 to 100.

Appendix B

Overview of Rule Extraction Methods

This section provides the detailed overview of rule extraction techniques used for rule generation purpose. The techniques explained are GATree, Decision Tree (DT), Ripper, Classification and Regression Tree (CART) and Dynamic Evolving Fuzzy Inference Systems (DENFIS).

B.1 GATree

Papagelis and Kalles [Papagelis and Kalles, 2000] [Papagelis and Kalles, 2001] proposed the breeding decision tree using evolutionary technique called genetic algorithms (GA). When applying GA to any kind of problem, one should know internal the appropriate representation of the search space, Generally GA uses binary string values to represent the points in search space. But this kind of representation doesnt appear in GATree approach.

In GATree, GALIBs (`http://lancet.mit.edu/ga/`) tree representation is used to build the population of minimal binary decision trees. Every decision node is a randomly chosen value. Building the population is accomplished in two steps. First, select an attribute randomly. If this attribute is nominal then randomly select one of its possible value. If it is continuous then select an integer randomly between the min and max of the attribute ranges. This kind of approach will reduce the size of the search space and is also straightforward.

The basic form of the GATree has introduced the minimum changes to the mutation-crossover operators in GA. Mutation chooses a random node from a tree and replaces the node value with randomly chosen new value. If the random node is a leaf node, then it is replaced with randomly chosen new class. The mutation example diagram is depicted in Figure B.1.

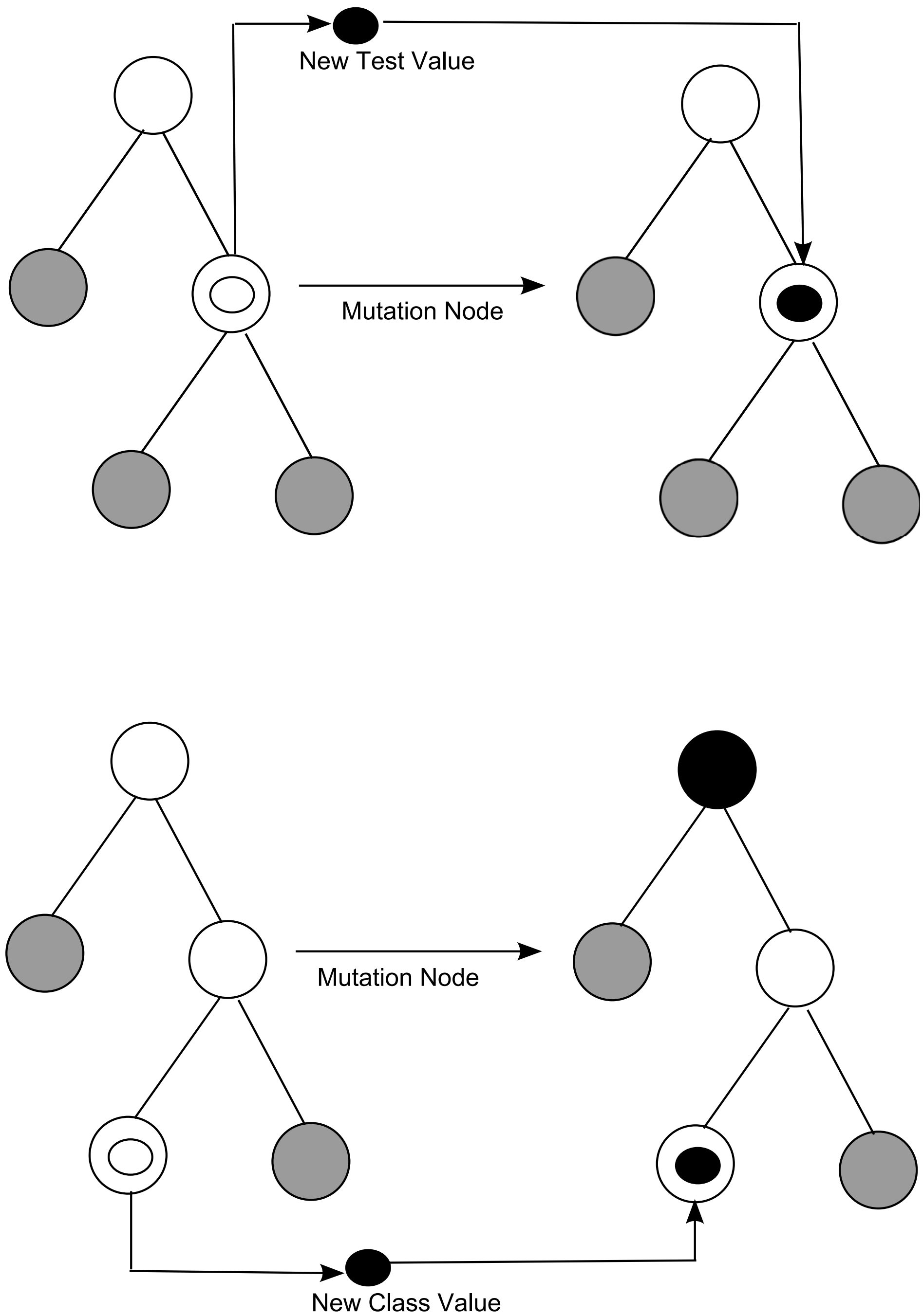

Figure B.1: GATree Mutation examples

The crossover operator chooses two random nodes which are either decision nodes or leaf nodes and swaps those nodes of sub-tree. The crossover operator does not affect the tree's rationality because the predicted value rests only on the leaves. The crossover example is shown in Figure B.2.

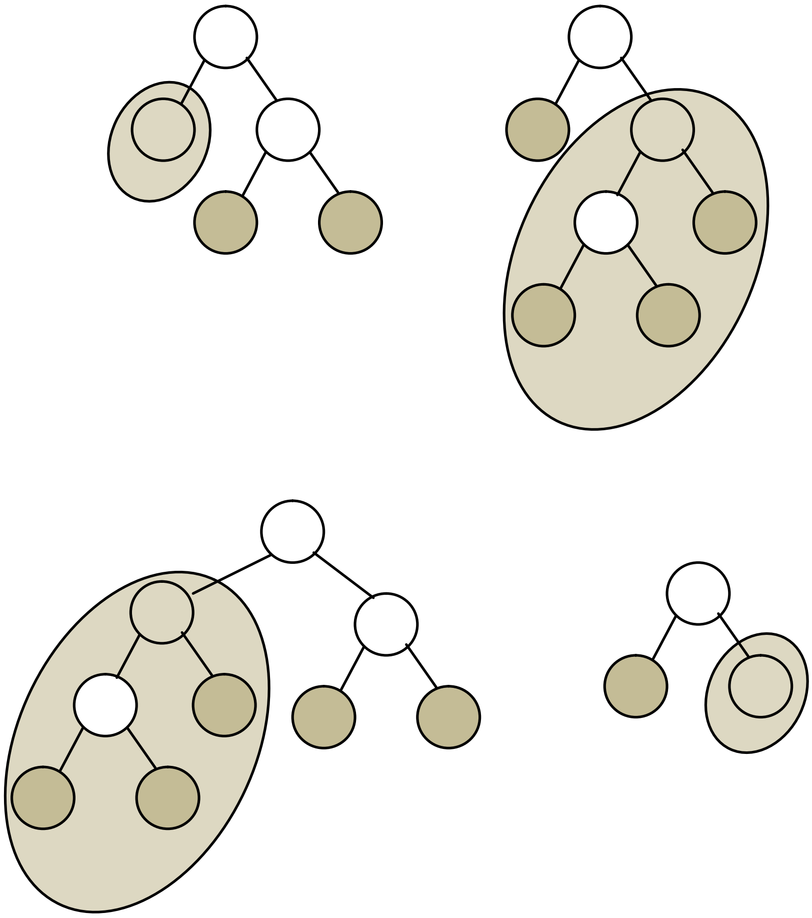

Figure B.2: GATree Crossover examples

In each generation, each tree is fixed to a scale payoff according to its performance. The higher payoff value is granted for smaller trees assuming that the smaller trees perform as good as the bigger trees. The payoff function is calculated to balance the accuracy and also the size of the tree.

$$payoff(tree_i) = correctclassified_i^2 * \frac{x}{size_i^2 + x} \tag{B.1}$$

The size factor (second part of the product) includes the factor x which is set to be a big number. When the size of the tree is small the size factor is nearly one and when the tree grows bigger the size factor decreases. Thus, the payoff will be greater for smaller trees. For example, if the x values is set to 1,000,000 then GA will search inside a bigger search space i.e., more number of trees. If the search space is bigger, then there will be less number of optimized trees for a fixed number of generations. Different size factor values will prefer different trees within some range. This may lead to more competent search.

If many decision trees generated in a generation have similar characteristics, then the similarity is reduced using the payoff function. Since GA evolves the complete solution, the algorithm can be stopped whenever necessary or convergence criterion is satisfied. The advantage of GA is that it is a highly parallel procedure. And the output is not only a single decision tree, but it is also a group of decision trees which can be used together.

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 2.0 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, 1978] [Quinlan, 1983] [Quinlan, 1993] [Quinlan, 1986]. 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 [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.

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

Decision tree induction is the learning of decision trees from class-labeled training objects. A decision tree is a flow chart like tree structure, where each internal node (nonleaf 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 rootnode. An example tree is shown in Figure B.3.

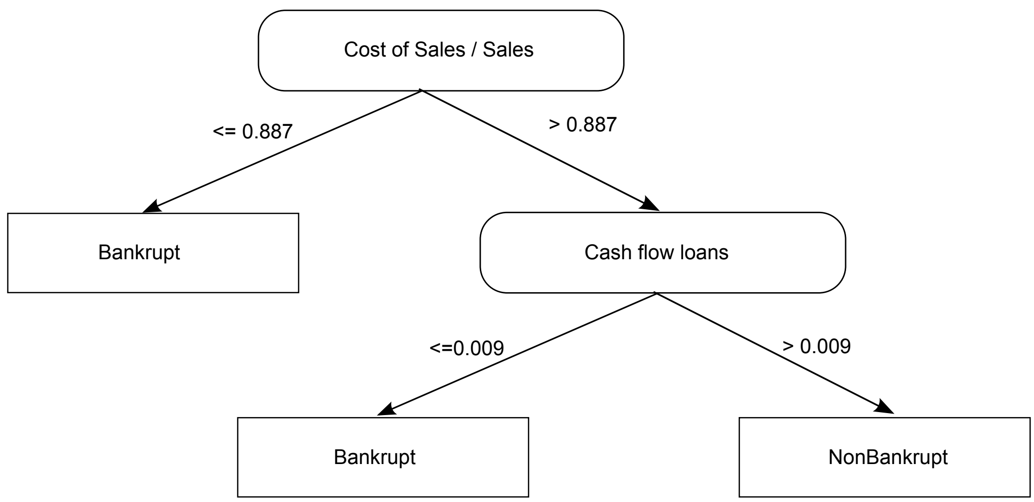

Figure B.3: Example of Decision Tree for Bankruptcy prediction in Banks

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 Dichotomizer). 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 a decision tree from a training object of data partition D.
- Data Partition, D - Attribute List - Attribute Selection 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 labeled with the class C;
4. If attribute list is empty then
5. Return N as a leaf node labeled 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 splitting 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 labeled 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 labeled 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 labeled 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, $a1, a2, ..., av$, 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 labeled with that value (Figure B.4a). Partition D_j is

the subset of class-labeled 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).

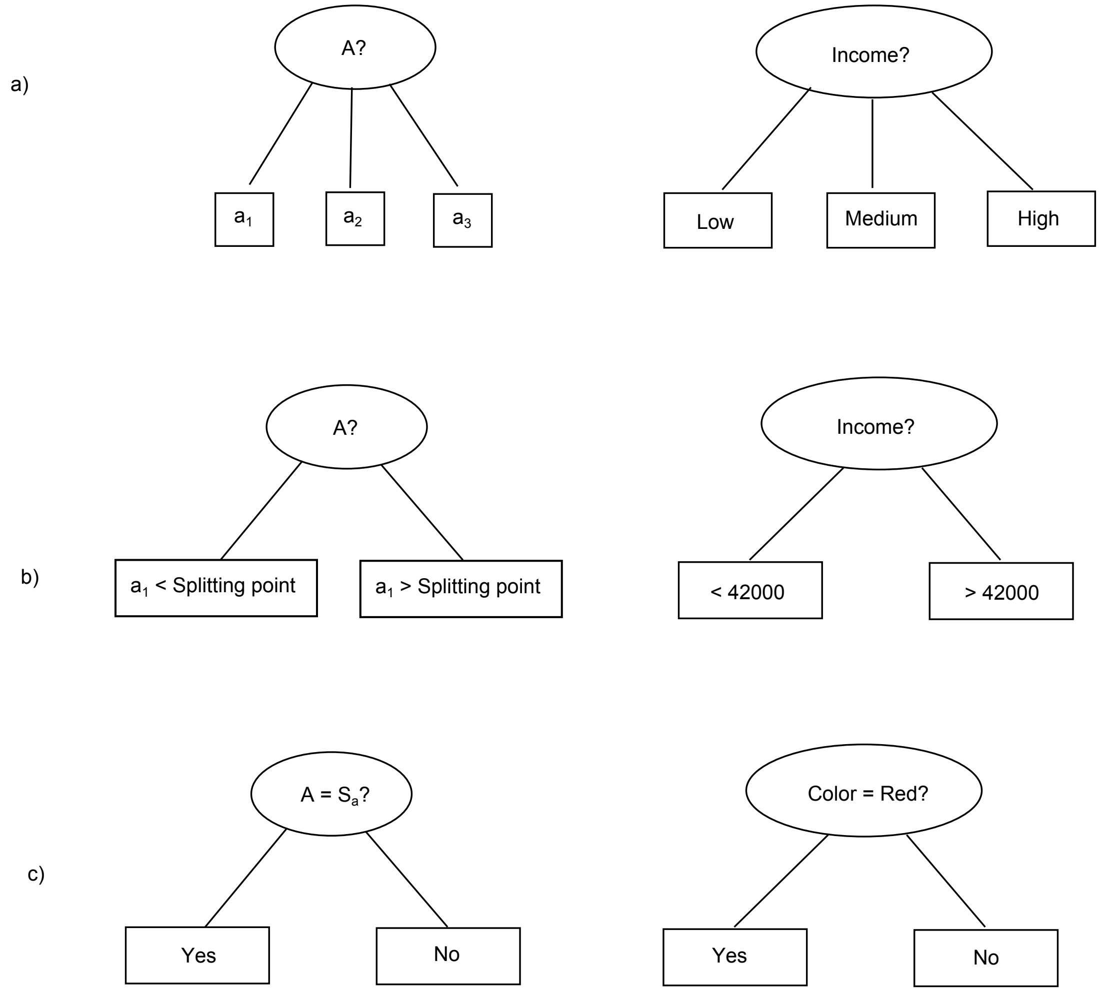

Figure B.4: Three possibilities for partitioning objects based on the splitting criterion

2. A is continuous-valued: in this case, the test at node N has two possible outcomes corresponding to the conditions $A \leq A \leq splitpoint$ and $A > splitpoint$, respectively. Two branches are grown from N and labeled according to the above outcomes (Figure B.4 b). The objects are partitioned such that D_1 holds the subset of class-labeled objects in D for which $A \leq splitpoint$, 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 Feature 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 (FigureB.4 c). By convention, the left branch out of N is labeled yes so that D_1 corresponds to the subset of class labeled objects in D that satisfy the test. The right branch out of N is labeled no so that D_2 corresponds to the subset of class-labeled 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).

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.

$$i(t) = \sum_{k!=l} p(\frac{k}{t})p(\frac{l}{t}) \tag{B.2}$$

where $k, l = 1, ..., K$ index of the class; $p(\frac{k}{t})$ conditional probability of class k provided in node t.

$$\triangle i(t) = -\sum_{k=1}^{K} p^2(\frac{k}{t_p}) + P_l \sum k = 1^K p^2(\frac{k}{t_l}) + P_r \sum_{k=1}^{K} p^2(\frac{k}{t_r}) \tag{B.3}$$

Therefore, Gini algorithm solves the following problem:

$$argmax_{x_j \leq x_j^R, j=1...M}[-\sum_{k=1}^{K} p^2(\frac{k}{t_p}) + P_l \sum k = 1^K p^2(\frac{k}{t_l}) + P_r \sum_{k=1}^{K} p^2(\frac{k}{t_r})] \tag{B.4}$$

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

B.3 Repeated Incremental Pruning to Produce Error Reduction (Ripper)

The Ripper [Cohen, 1995] is a classification algorithm designed to generate rules set directly from the training dataset. The name is drawn from the fact that the rules are learned incrementally. A new rule associated with a class value will cover various attributes of that class .The algorithm was designed to be fast and effective when dealing with large and noisy datasets compared to decision trees [Cohen, 1995].

The Ripper algorithm is illustrated by Algorithm 1 (adapted from [Cohen, 1995]):

1. $S = X, C$ represents the training set, where $X = x_1, x_2, , x_k$ represents the instances and $C = c_1, c_2, , c_k$ represents the class-label associated with each instance.

2. The classes $c_1, , c_k$ are sorted in the order of least prevalent class to the most

```
begin sort classes in the order of least prevalent class to the most prevalent
class.
create a new rule set
while while iterating from the prevalent class to the most prevalent class do
    split S into S_pos and S_neg
    while S_pos is not empty do
        split SPos and Sneg into Gpos and Gneg subsets and Ppos and Pneg
        subsets.
        create and prune a new rule
        if the error rate of the new rule is very large then

        end
        else add new rule to rule set
        the total description length l is computed
        if l > d then
        end
    end
end
```

Algorithm 2: Pseudocode for Ripper Algorithm

frequent class. This is done by counting the number instances associated with each class. The instances associated with the least prevalent class are separated into S_{Pos} subset whilst the remaining instances are grouped into S_{neg} subset.

3. IREP is invoked (with S_{Pos} and S_{neg} subsets passed as parameters) to find the rule set that splits least prevalent class from the other classes.

4. Initialise an empty rule set R.

5. S_{Pos} and S_{neg} are split into growing positive G_{pos} and growing negative G_{neg} subsets as well as pruning positive P_{pos} and negative P_{neg} subsets. Growing positive subsets contains instances that are associated with the least prevalent class. Growing negative subset contains instances associated with the remaining classes. This is similar to the P_{pos} and P_{neg} subsets.

6. A new rule is created by growing G_{pos} and G_{neg}. This is done by iteratively adding conditions that maximize the information gain criterion until the rule cannot cover any negative instances from the growing dataset.

7. The new rule is pruned for optimization of the function $v = \frac{p-n}{p+n}$. using P_{pos} and P_{neg} subsets. p is number of rules to prune and n is the total number

of rules.

8. Check the error rate of the new rule very large, and then return the rule set. Otherwise, the new rule is added to the rule set and the total *descriptionlength* is computed. If the lengths exceeds a certain number d, then the algorithm stops, otherwise repeat from step 5.

9. Iterate to the next least prevalent class and then repeat from step 3.

During the growing phase of the algorithm, a greedy approach of learning is applied, i.e., each rule is learned one at a time.

B.4 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 et al., [Breiman et al., 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.5.

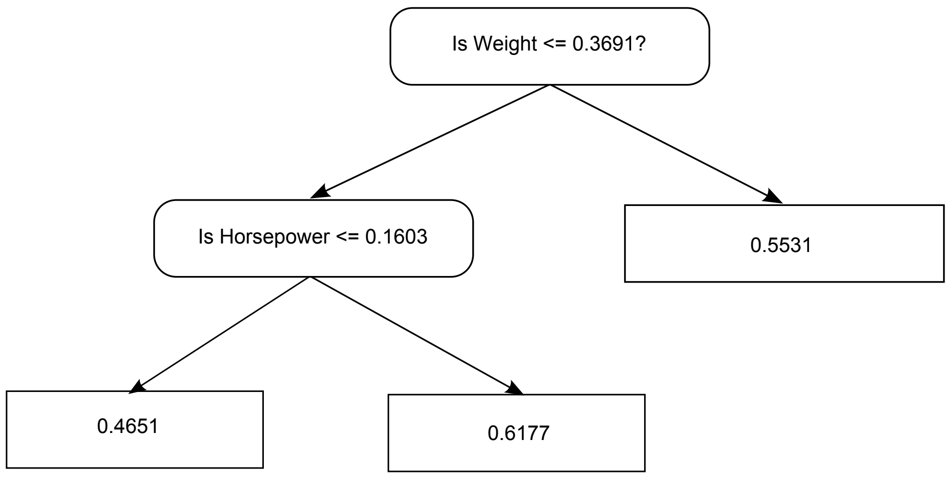

Figure B.5: CART AutoMPG example

CART methodology

– Construction of maximum tree

– Choice of the right tree size

– Classification of new data using constructed tree

B.4.1 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.6.

where t_p, t_l, t_r parent, left and right nodes; x_j variable j; R_j x best splitting value of variable x_j .

Maximum homogeneity of child nodes is defined by so called impurity function

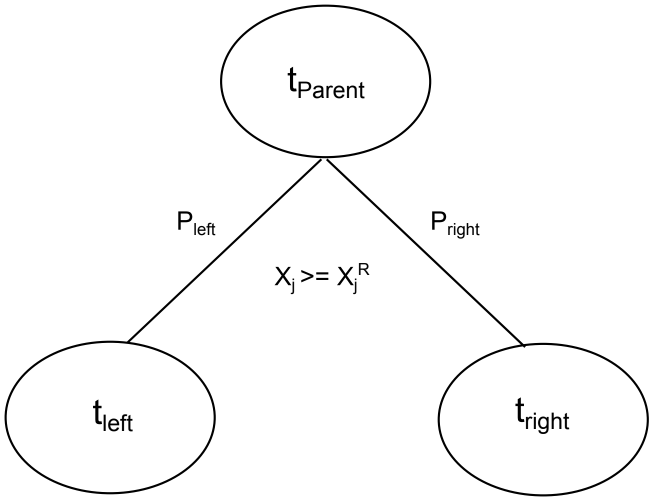

Figure B.6: Splitting Algorithm of CART

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

$$\triangle i(t) = i(t_p) - E[i(t_c)] \tag{B.5}$$

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

$$\triangle i(t) = i(t_p) - P_l i(t_l) - P_r i(t_r) \tag{B.6}$$

Therefore, at each node CART solves the following maximization problem:

$$argmax_{x_j \leq x_j^R, j=1...M} i(t_p) - E[i(t_c)] \tag{B.7}$$

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 preassigned 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.

$$argmax_{x_j \leq x_j^R, j=1...M}[P_l Var(Y_l) + P_r Var(Y_r)] \quad (B.8)$$

where $Var(Y_l)$, $Var(Y_r)$ response vectors for corresponding left and right child node; $x_j < x_j^R$, $j = 1, ..., M$ optimal splitting question satisfying above condition. Squared residuals minimization algorithm is identical to Gini splitting rule.

B.4.2 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.

$$R_\alpha(T) = R(T) + \alpha(\hat{T}) \rightarrow min_T \quad (B.9)$$

where $R(T)$ is misclassification error of the tree T; $\alpha(\hat{T})$ (T - complexity measure which depends on $\hat{T}$(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.4.3 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.4.4 Advantages of CART

1. It can handle both numerical and categorical variables.
2. It is robust to outliers.
3. It is the only tree capable of generating regression trees.
4. 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.
5. CART is a nonparametric approach.
6. It does not require variables to be selected in advance.
7. CART results are invariant to monotone transformations of its independent variables.

8. CART has no assumptions and it is computationally fast.

9. CART is flexible and has an ability to adjust in time.

B.4.5 Disadvantages of CART

1. CART may have unstable decision trees.

2. CART splits only by one variable.

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, 1998], [Kasabov, 1999a], [Kasabov, 1999b], [Kasabov and Song, 2002]) . Fast learning, online incremental adaptive learning, open structure organization, memorizing information, active interaction, knowledge acquisition and self-improvement and spatial and temporal learning are some of the major requirements of the intelligent systems [Kasabov, 1998], [Kasabov, 1999a], [Kasabov, 2001], [Kasabov and Woodford, 1999]. Neucom 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. DENFISs 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, $Dthr$.

B.5.3 ECM Algorithm

The ECM algorithm is described as follows.

1. 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.

2. 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.

3. 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 $D_{ij} \leq Ru_j$, $j = 1, 2, ..., n$.

   In this case, neither a new cluster is created nor any existing cluster is updated (in cases of x_4 and x_6) the algorithm returns to step 2 else go to next step.

4. 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 $S_{ia} : S_{ia} = D_{ia} + Ru_a = min(S_{ij})$, $j = 1, 2, ..., n$.

5. 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 1 (the cases of x_3 and x_8) and the algorithm returns to step 2.

6. 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 new a 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). The algorithm returns to Step 2.

B.5.4 DENFIS

Online and offline models of DENFIS [Kasabov and Song, 2002] 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 : $\mu_{R_{ij}} : X_j \to [0, 1]$, $i = 1, 2, ..., m$; $j = 1, 2, ..., q$. 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:

$$\mu(x) = mf(x, a, b, c) = 0, x \leq a\frac{x-a}{b-a}, a \leq x \leq b\frac{c-x}{c-b}, b \leq x \leq c0.c \leq xfa \tag{B.10}$$

Where: b is the value of the cluster center on the x dimension, $a = b - d * Dthr$ and $c = b + d \times Dthr$, $d = 1.2 - 2$ 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 $x_0 = [x_1 0, x_2 0, ..., x_q 0]$, the result of inference, y_0 (the output of the system) is the weighted average of each rule's output indicated as follows:

$$y_0 = frac\sum_{i=1}^{m} \omega_i f_i(x_{10}, x_{20}, ..., x_{q0}\sum_{i=1}^{m} \omega_i \tag{B.11}$$

where $\omega_i = \pi_{j=1}^{q} \mu_{R_{ij}}(x_{j0}); i = 1, 2, .., m; j = 1, 2, ..., q;$

Appendix C

Datasets Description

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

S.No	Predictor Variables	Accronym
1	Current Assets/Total Assets	CA/TA
2	Current Assets-Cash/Total Assets	CAC/TA
3	Current Assets/Loans	CA/L
4	Reserves/Loans	R/L
5	Net Income/Total Assets	NI/TA
6	Net Income/Total Equity Capital	NI/TEC
7	Net Income/Loans	NI/L
8	Cost Of Sales/Sales	CS/S
9	Cash Flow/Loans	CF/L

Table C.2: Financial Ratios of Turkish Banks Dataset

S.No	Predictor Variables	Accronym
1	Interest Expenses/Average Profitable Assets	IE/APA
2	Interest Expenses/Average Non-Profitable Assets	IE/ANA
3	(Share Holders Equity + Total Income)/ (Deposits + Non-Deposit Funds)	(SE+TI)/(D+NF)
4	Interest Income/Interest Expenses	II+IE
5	(Share Holders Equity + Total Income)/ Total Assets	(SE+TI)/TA
6	(Share Holders Equity + Total Income)/ (Total Assets +Contingencies and Commitments)	(SE+TI)/(TA+CC)
7	Networking Capital/Total Assets	NC/TA
8	(Salary And Employees Benefits + Reserve For Retirement)/No. Of Personnel	(SEB+RR)/P
9	Liquid Assets/(Deposits + Non-Deposit Funds)	LA/(D+NF)
10	Interest Expenses/Total Expenses	IE/TE
11	Liquid Assets/Total Assets	LA/TA
12	Standard Capital Ratio	SCR

Table C.3: Financial Ratios of US Banks Dataset

S.No	Predictor Variables	Accronym
1	Working Capital/Total Assets	WC/TA
2	Retained Earnings/ Total Assets	RE/TA
3	Earnings Before Interest And Taxes/ Total Assets	EIT/TA
4	Market Value Of Equity/ Total Assets	ME/TA
5	Sales/ Total Assets	S/TA

Table C.4: Financial Ratios of UK Banks Dataset

S.No	Predictor Variables	Accronym
1	Sales	Sales
2	Profit Before Tax/Capital Employed (%)	PBT/CE
3	Funds Flow/Total Liabilities	FF/TL
4	(Current Liabilities + Long Term Debits)/ Total Assets	(CL+LTD)/TA
5	Current Liabilities/Total Assets	CL/TA
6	Current Assets/Current Liabilities	CA/CL
7	Current Assets-Stock/Current Liabilities	CA-S/CL
8	Current Assets-Current Liabilities/ Total Assets	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 Churn Prediction Dataset

S.No	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	Customers 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	Customers margin for the company in months T	Real Number
16	MAR_T-1	Customers margin for the company in months T-1	Real Number
17	MAR_T-2	Customers margin for the company in months T-2	Real Number
18	MAR_T-3	Customers margin for the company in months T-3	Real Number
19	MAR_T-4	Customers margin for the company in months T-4	Real Number
20	MAR_T-5	Customers margin for the company in months T-5	Real Number
21	MAR_T-6	Customers margin for the company in months T-6	Real Number

Table C.6: Feature description of Auto MPG Dataset

S.No	Features	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.7: Feature description of Body Fat Dataset

S.No	Features	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.8: Feature description of Boston Housing Dataset

S.No	Features	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 USD10,000	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 USD1000's**	**TARGET**

Table C.9: Feature description of Forest Fires Dataset

S.No	Features	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.10: Feature description of Pollution Dataset

S.No	Features	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 ¡ USD 3000	Continuous
12	HC Relative hydrocarbon pollution potential	Continuous
13	NOX nitric oxides	Continuous
14	SO sulphur dioxide	Continuous
15	HUMID Annual average % relative humidity at 1pm	Continuous
16	**MORT Total age-adjusted mortality rate per 100,000**	**TARGET**

References

[Agrawal and Srikant, 2000] Agrawal, R. and Srikant, R. (2000). ”privacy preserving data mining”. *ACM SIGMOD Record*, 29(2):439–450. 29

[Aliev et al., 2008] Aliev, R., Aliev, R., Guirimov, B., and Uyar, K. (2008). Dynamic data mining technique for rules extraction in a process of battery charging. *Applied Soft Computing*, 8(3):1252 – 1258. ¡ce:title¿Forging the Frontiers - - Soft Computing¡/ce:title¿. 6

[Amari and Kasabov, 1998] Amari, S.-I. and Kasabov, N., editors (1998). *Brain-like Computing and Intelligent Information Systems*. Springer-Verlag Singapore Pte. Limited, 1st edition. 215

[Amora et al., 2009] Amora, M., Almeida, O., Braga, A., Barbosa, F., Lima, S., and Lisboa, L. (2009). Decompositional rule extraction from artificial neural networks and application in analysis of transformers. In *Intelligent System Applications to Power Systems, 2009. ISAP '09. 15th International Conference on*, pages 1 –6. 18

[Andrews et al., 1995] Andrews, R., Diederich, J., and Tickle, A. B. (1995). Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, 8(6):373 – 389. ¡ce:title¿Knowledge-based neural networks¡/ce:title¿. 2, 3, 4, 7

[Andrews and Geva, 2002] Andrews, R. and Geva, S. (2002). Rule extraction from local cluster neural nets. *Neurocomputing*, 47(14):1 – 20. 16

[Arbatli and Akin, 1997] Arbatli, A. D. and Akin, H. L. (1997). Rule extraction from trained neural networks using genetic algorithms. *Nonlinear Anal.*, 30:1639–1648. 4, 16

[Asuncion and Newman, 2007] Asuncion, A. and Newman, D. (2007). Uci machine learning repository, irvine. CA:University of California, School of Information and Computer Science (2007). 10, 11, 39, 106

[Baldi and Hornik, 1989] Baldi, P. and Hornik, K. (1989). Neural networks and principal component analysis: learning from examples without local minima. *Neural Networks*, 2(1):53–58. 34

[Bansal et al., 2011] Bansal, A., Chen, T., and Zhong, S. (2011). Privacy preserving back-propagation neural network learning over arbitrarily partitioned data. *Neural Comput. Appl.*, 20(1):143–150. 32

[Becerra et al., 2005] Becerra, V. M., Galvão, R. K., and Abou-Seada, M. (2005). Neural and wavelet network models for financial distress classification. *Data Min. Knowl. Discov.*, 11(1):35–55. 100

[Behlen and Johnson, 1999] Behlen, F. and Johnson, S. (1999). Multicenter patient records research: security policies and tools. *Journal of the American Medical Informatics Association*, 6(6):435 – 443. 30

[Benoudjit and Verleysen, 2003] Benoudjit, N. and Verleysen, M. (2003). On the kernel widths in radial-basis function networks. *Neural Process. Lett.*, 18:139–154. 68, 72

[Berman, 2002] Berman, J. J. (2002). Confidentiality issues for medical data miners. *Artif. Intell. Med.*, 26(1-2):25–36. 30

[Bertino et al., 2005] Bertino, E., Fovino, I. N., and Provenza, L. P. (2005). A framework for evaluating privacy preserving data mining algorithms*. *Data Min. Knowl. Discov.*, 11(2):121–154. 31

[Beynon and Peel, 2001] Beynon, M. J. and Peel, M. J. (2001). Variable precision rough set theory and data discretisation: an application to corporate failure prediction. *Omega*, 29(6):561 – 576. 10, 39, 73, 106, 178

[Bhalla et al., 2012] Bhalla, D., Bansal, R. K., and Gupta, H. O. (2012). Function analysis based rule extraction from artificial neural networks for transformer incipient fault diagnosis. *Electrical Power and Energy Systems*, 43:11961203. 21

[Bhat et al., 2006] Bhat, T., Venkataramani, D., Ravi, V., and Murty, C. (2006). An improved differential evolution method for efficient parameter estimation in biofilter modeling. *Biochemical Engineering Journal*, 28(2):167 – 176. 192

[Bolton et al., 2000] Bolton, R., Kannan, P., and Bramlett, M. (2000). Implications of loyalty program membership and service experiences for customer retention and value. *Journal of the Academy of Marketing Science*, 28:95–108. 83

[Bolton, 1998] Bolton, R. N. (Winter 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. 83

[Boyens et al., 2004] Boyens, C., Krishnan, R., and Padman, R. (2004). On privacy-preserving access to distributed heterogeneous healthcare information. *Hawaii International Conference on System Sciences*, 6:60135a. 30

[Breiman et al., 1984] Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and regression trees*. Chapman & Hall/CRC. 210

[Campos and Ludermir, 2005] Campos, P. G. and Ludermir, T. B. (2005). Literal and prorulext: Algorithms for rule extraction of anns. In *Proceedings of the Fifth International Conference on Hybrid Intelligent Systems*, HIS '05, pages 143–148, Washington, DC, USA. IEEE Computer Society. 6

[Canbas et al., 2005] Canbas, S., Cabuk, A., 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(2):528 – 546. 9, 39, 73, 106, 178

[Chauhan et al., 2009] Chauhan, N., Ravi, V., and Chandra, D. K. (2009). Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks. *Expert Systems with Applications*, 36(4):7659 – 7665. 70, 100, 104

[Chawla et al., 2002] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16:321–357. 88, 89

[Chen, 1995] Chen, H. (1995). Machine learning for information retrieval: neural networks, symbolic learning, and genetic algorithms. *J. Am. Soc. Inf. Sci.*, 46(3):194–216. 87

[Chile, 2004] Chile (2004). Business intelligence cup-2004: Organized by the university of chile. `http://www.tis.cl/bicup_04/textbicup/BICUP/202004/20public/20data.zip`. Accessed: 2004. 10, 90, 91

[Clark and Niblett, 1989] Clark, P. and Niblett, T. (1989). The cn2 induction algorithm. *Machine Learning*, 3:261–283. 10.1023/A:1022641700528. 5

[Clifton and Marks, 1996] Clifton, C. and Marks, D. (1996). Security and privacy implications of data mining. (96-08):15–19. 31

[Cohen et al., 2006] Cohen, G., Hilario, M., Sax, H., Hugonnet, S., and Geissbuhler, A. (2006). Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7 – 18. ¡ce:title¿Intelligent Data Analysis in Medicine¡/ce:title¿. 89

[Cohen, 1995] Cohen, W. W. (1995). Fast effective rule induction. In *ICML'95*, pages 115–123. 208

[Craven, 1996] Craven, M. W. (1996). Extracting comprehensible models from trained neural networks. Technical report. Ph.D Thesis. 3, 4, 18

[Craven and Shavlik, 1996] Craven, M. W. and Shavlik, J. W. (1996). Extracting Tree-Structured representations of trained networks. In Touretzky, D. S., Mozer, M. C., and Hasselmo, M. E., editors, *Advances in Neural Information Processing Systems*, volume 8, pages 24–30. The MIT Press. 5

[Crises, 2004] Crises, G. (2004). Non-perturbative methods for microdata privacy in statistical databases. 33

[Dalenius and Reiss, 1982] Dalenius, T. and Reiss, S. P. (1982). Data-swapping: A technique for disclosure control. *Journal of Statistical Planning and Inference*, 6(1):73 – 85. 33

[Davel and Barnard, 2004] Davel, M. H. and Barnard, E. (2004). The efficient generation of pronunciation dictionaries: machine learning factors during bootstrapping. In *INTERSPEECH*. ISCA. 87

[d'Avila Garcez et al., 2001] d'Avila Garcez, A., Broda, K., and Gabbay, D. (2001). Symbolic knowledge extraction from trained neural networks: A sound approach. *Artificial Intelligence*, 125(12):155 – 207. 19

[Davis et al., 1977] Davis, R., Buchanan, B., and Shortliffe, E. (1977). Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence*, 8(1):15 – 45. 3

[DeSousa et al., 2003] DeSousa, T. F., Silva, A., and Neves, A. (2003). A particle swarm data miner. In *EPIA'03*, pages 43–53. 24, 176, 178

[Domingos, 1999] Domingos, P. (1999). Metacost: a general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 155–164, New York, NY, USA. ACM. 88

[Dumitrescu and Simon, 2003] Dumitrescu, D. and Simon, K. (2003). A new dynamic evolutionary clustering techniques application in designing rbf neural network topologies clustering algorithm. *Informatica, Studia Universal Babes-Bolyai*, 48(2). 70

[Edwards, 2007] Edwards, J. (2007). *Get It Together with Collaborative CRM*. Inside CRM. Tippit. 82

[Elalfi et al., 2004] Elalfi, A., Haque, R., and Elalami, M. (2004). Extracting rules from trained neural network using ga for managing e-business. *Applied Soft Computing*, 4(1):65 – 77. 16

[Estabrooks et al., 2004] Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36. 88

[Etchells and Lisboa, 2006] Etchells, T. A. and Lisboa, P. J. (2006). Orthogonal search-based rule extraction (osre) for trained neural networks: a practical and efficient approach. *Trans. Neur. Netw.*, 17(2):374–384. 20

[Evfimievski, 2002] Evfimievski, A. (2002). Randomization in privacy preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):43–48. 32

[Faifer et al., 1999] Faifer, M., Janikow, C., and Krawiec, K. (1999). Extracting fuzzy symbolic representation from artificial neural networks. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pages 600 –604. 8

[Fan and Li, 2002] Fan, Y. and Li, C. (2002). Diagnostic rule extraction from trained feedforward neural networks. *Mechanical Systems and Signal Processing*, 16(6):1073 – 1081. 19

[Fawcett, 2006] Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recogn. Lett.*, 27:861–874. 7

[Fawcett and Provost, 1997] Fawcett, T. and Provost, F. (1997). Adaptive fraud detection. *Data Min. Knowl. Discov.*, 1(3):291–316. 87

[Fayyad et al., 1996] Fayyad, U., Piatetsky-shapiro, G., and Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54. 2, 9

[Fienberg, 2005] Fienberg, S. (2005). homeland insecurity: datamining, terrorism detection, and confidentiality. 30

[Frawley et al., 1992] Frawley, W. J., Piatetsky-shapiro, G., and Matheus, C. J. (1992). Knowledge discovery in databases: an overview. 1

[Fu, 1994] Fu, L. (1994). Rule generation from neural networks. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(8):1114 –1124. 16

[Fujimoto and Nakabayashi, 2003] Fujimoto, K. and Nakabayashi, S. (2003). Applying gmdh algorithm to extract rules from examples. *Syst. Anal. Model. Simul.*, 43:1311–1319. 19, 93

[Fung et al., 1996] Fung, C. F., Billings, S. A., and Luo, W. (1996). On-line supervised adaptive training using radial basis function networks. *Neural Networks*, 9:1597–1617. 69

[Gail A. and Ah-H Tee, 1995] Gail A., C. and Ah-H Tee, T. (1995). Rule extraction: From neural representation architecture to symbolic. *Connection Science*, 7(1). 18

[Galindo and Tamayo, 2000] Galindo, J. and Tamayo, P. (2000). Credit risk assessment using statistical and machinelearning: Basic methodology and risk modeling applications. *Comput. Econ.*, 15(1-2):107–143. 87

[Gallant, 1988] Gallant, S. I. (1988). Connectionist expert systems. *Commun. ACM*, 31:152–169. 3, 6

[Garson, 1991] Garson, G. D. (1991). Interpreting neural-network connection weights. *AI Expert*, 6(4):46–51. 105

[Gilbert, 1989] Gilbert, N. (1989). Explanation and dialogue. *The Knowledge Engineering Review*, 4(03):235–247. 3

[Giles and Omlin, 1993] Giles, C. L. and Omlin, C. W. (1993). Extraction, insertion and refinement of symbolic rules in dynamically driven recurrent neural networks. *Connection Science*, 5(3):307–337. 8

[Granmo and Oleshchuk, 2005] Granmo, O. and Oleshchuk, V. (2005). Privacy preserving data mining in telecommunication services. *International Journal of Computing*, 3(4):8590. 32

[Grossmann and Morlet, 1984] Grossmann, A. and Morlet, J. (1984). Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, 15(4):723–736. 99

[Guangbin et al., 2007] Guangbin, Y., Guixian, L., Yanwei, B., and Xiangyang, J. (2007). Tuning of the structure and parameters of wavelet neural network using improved chaotic pso. In *Control Conference, 2007. CCC 2007. Chinese*, pages 228 –232. 100

[Guo et al., 2007] Guo, P., Chen, J., and Sun, S.-J. (2007). An method to extract comprehensible rules. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 2, pages 808 –812. 20

[Guo et al., 2001] Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2001). The effect of class distribution on classifier learning. In *Technical Report Machine Learning*, number 43, pages 192 –201. 88

[Guo et al., 2008] Guo, X., Yin, Y., Dong, C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In *Natural Computation, 2008. ICNC '08. Fourth International Conference on*, volume 4, pages 192 –201. 88

[Guyon and Elisseeff, 2003] Guyon, I. and Elisseeff, A. (2003). An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182. 37, 104

[Han et al., 2005] Han, H., Wang, W.-Y., and Mao, B.-H. (2005). Borderline-SMOTE: A new Over-Sampling method in imbalanced data sets learning. pages 878–887. 89

[Han and Xi, 2004] Han, M. and Xi, J. (2004). Efficient clustering of radial basis perceptron neural network for pattern recognition. *Pattern Recognition*, 37(10):2059 – 2067. 70

[Hart, 1968] Hart, P. (1968). The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515–516. 88

[Hayashi and Imura, 1990] Hayashi, Y. and Imura, A. (1990). Fuzzy neural expert system with automated extraction of fuzzy if-then rules from a trained neural network. In *Uncertainty Modeling and Analysis, 1990. Proceedings., First International Symposium on*, pages 489 –494. 16

[Holden and Freitas, 2007] Holden, N. P. and Freitas, A. A. (2007). A hybrid pso/aco algorithm for classification. In *Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, GECCO '07, pages 2745–2750, New York, NY, USA. ACM. 23

[Hruschka and Ebecken, 2006] Hruschka, E. R. and Ebecken, N. F. (2006). Extracting rules from multilayer perceptrons in classification problems: A clustering-based approach. *Neurocomputing*, 70(13):384 – 397. 21

[Hruschka and Natter, 1999] Hruschka, H. and Natter, M. (1999). Comparing performance of feedforward neural nets and k-means for cluster-based market segmentation. *European Journal of Operational Research*, 114(2):346–353. 34, 35

[Ilonen et al., 2003] Ilonen, J., Kamarainen, J.-K., and Lampinen, J. (2003). Differential evolution training algorithm for feed-forward neural networks. *Neural Process. Lett.*, 17:93–105. 70

[Ivakhnenko, 1996] Ivakhnenko, A. G. (1996). The group method of data handling- a rival of the method of stochastic approximation. *Soviet Automatic Control*, 13(3):43–55. 83, 85

[Jackson and Craven, 1996] Jackson, J. C. and Craven, M. W. (1996). Learning sparse perceptrons. In *In Advances in Neural Information Processing Systems 8*, pages 654–660. MIT Press. 3

[Japkowicz and Stephen, 2002] Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449. 87

[Ji et al., 2006] Ji, J., Zhang, N., Liu, C., and Zhong, N. (2006). An ant colony optimization algorithm for learning classification rules. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '06, pages 1034–1037, Washington, DC, USA. IEEE Computer Society. 23

[Jian-guo et al., 2008] Jian-guo, W., Jian-hong, Y., Wen-xing, Z., and Jin-wu, X. (2008). Rule extraction from artificial neural network with optimized activation functions. In *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, volume 1, pages 873 –879. 20

[Jumadinova and Dasgupta, 2008] Jumadinova, J. and Dasgupta, P. (2008). Firefly-inspired synchronization for improved dynamic pricing in online markets. In *Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, SASO '08, pages 403–412, Washington, DC, USA. IEEE Computer Society. 194

[Kahramanli and Allahverdi, 2009] Kahramanli, H. and Allahverdi, N. (2009). Rule extraction from trained adaptive neural networks using artificial immune systems. *Expert Systems with Applications*, 36(2):1513–1522. 21

[Karunanithi et al., 1992] Karunanithi, N., Whitley, D., and Malaiya, Y. K. (1992). Prediction of software reliability using connectionist models. *IEEE Trans. Softw. Eng.*, 18(7):563–574. 85

[Kasabov and Song, 2002] Kasabov, N. and Song, Q. (2002). Denfis: Dynamic, evolving neural-fuzzy inference systems and its application for time-series prediction. *IEEE Transactions on Fuzzy Systems*, 10:144 –154. 215, 217

[Kasabov, 1998] Kasabov, N. K. (1998). Ecos: Evolving connectionist systems and the eco learning paradigm. pages 1222–1235. 215

[Kasabov, 1999a] Kasabov, N. K., editor (1999a). *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*. 215

[Kasabov, 1999b] Kasabov, N. K., editor (1999b). *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*. The MIT Press (October 11, 1996). 215

[Kasabov, 2001] Kasabov, N. K. (2001). Evolving fuzzy neural networks for on-line, adaptive, knowledge based learning. *IEEE Transactions Systems, Man, Cybernetics, PART B, 31*, pages 902–918. 215

[Kasabov and Woodford, 1999] Kasabov, N. K. and Woodford, B. (1999). Rule insertion and rule extraction from evolving fuzzy neural networks: Algorithms and applications for building adaptive, intelligent expert systems. *IEEE International Fuzzy Systems Conference Proceedings*. 215

[Kennedy and Eberhart, 1995] Kennedy, J. and Eberhart, R. (1995). Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942 –1948. 188

[Ketel and Homaifar, 2005] Ketel, M. and Homaifar, A. (2005). Privacy preserving mining by rotational data transformation. In *Proceedings of the 43rd annual Southeast regional conference*, volume 1 of *ACM-SE 43*, pages 233–236, New York, NY, USA. ACM. 32

[Kim and Han, 2003] Kim, M.-J. and Han, I. (2003). The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms. *Expert Systems with Applications*, pages 637–646. 22

[Kononenko, 2001] Kononenko, I. (2001). Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89 – 109. 87

[Kotler, 2000] Kotler, P. (2000). Marketing management: The millennium edition. *Englewood Cliffs, NJ: Prentice Hall International, USA*. 82

[Kotsiantis and Pintelas, 2006] Kotsiantis, S., K. D. and Pintelas, P. (2006). Handling imbalanced datasets: a review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36. 88

[Kramer, 1991] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243. 35

[Krishnan et al., 1999] Krishnan, R., Sivakumar, G., and Bhattacharya, P. (1999). A search technique for rule extraction from trained neural networks. *Pattern Recognition Letters*, 20(3):273 – 280. 6, 18

[Krishnanand and Ghose, 2006] Krishnanand, K. N. and Ghose, D. (2006). Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. *Multiagent and Grid Systems*, pages 209–222. 194

[Kubat et al., 1998] Kubat, M., Holte, R. C., and Matwin, S. (1998). Machine learning for the detection of oil spills in satellite radar images. *Mach. Learn.*, 30(2-3):195–215. 87, 88

[Kubat and Matwin, 1997] Kubat, M. and Matwin, S. (1997). Addressing the curse of imbalanced training sets: One-sided selection. In *In Proceedings of* the *Fourteenth International Conference on Machine Learning*, pages 179–186. Morgan Kaufmann. 88

[Kulluk et al., 2012] Kulluk, S., zbakr, L., and Baykasolu, A. (2012). Fuzzy difaconn-miner: A novel approach for fuzzy rule extraction from neural networks. *Expert Systems with Applications*, (0):–. 22

[Kumar and Ravi, 2008] Kumar, D. A. and Ravi, V. (2008). Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, 1(1):4 – 28. 88

[Langley and Simon, 1995] Langley, P. and Simon, H. A. (1995). Applications of machine learning and rule induction. *Commun. ACM*, 38(11):54–64. 1

[Lariviere and den Poel, 2004] Lariviere, B. and den Poel, D. V. (2004). Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services. *Expert Systems with Applications*, 27(2):277 – 285. 83

[Laurikkala, 2001] Laurikkala, J. (2001). Improving identification of difficult small classes by balancing class distribution. In *Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine*, AIME '01, pages 63–66, London, UK, UK. Springer-Verlag. 88

[Lee et al., 1999] Lee, W., Stolfo, S. J., and Mok, K. W. (1999). Mining in a dataflow environment: experience in network intrusion detection. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 114–124, New York, NY, USA. ACM. 87

[Leidenfrost and Elmenreich, 2008] Leidenfrost, R. and Elmenreich, W. (2008). Establishing wireless time-triggered communication using a firefly clock synchronization approach. In *Intelligent Solutions in Embedded Systems, 2008 International Workshop on*, pages 1 –18. 194

[Lindell and Pinkas, 2000] Lindell, Y. and Pinkas, B. (2000). Privacy preserving data mining. In *Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '00, pages 36–54, London, UK, UK. Springer-Verlag. 29

[Ling and Li, 1998] Ling, C. X. and Li, C. (1998). Data mining for direct marketing: Problems and solutions. In *Knowledge Discovery and Data Mining*, pages 73–79. 88

[Liu et al., 2006a] Liu, K., Kargupta, H., and Ryan, J. (2006a). Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *Knowledge and Data Engineering, IEEE Transactions on*, 18(1):92 – 106. 32

[Liu et al., 2006b] Liu, Y., Chawla, N. V., Harper, M. P., Shriberg, E., and Stolcke, A. (2006b). A study in machine learning from imbalanced data for sentence boundary detection in speech. *Computer Speech and Language*, pages 468–494. 89

[Liu et al., 2004] Liu, Y., Qin, Z., Shi, Z., and Chen, J. (2004). Rule discovery with particle swarm optimization. In Chi, C.-H. and Lam, K.-Y., editors, *Content Computing*, volume 3309 of *Lecture Notes in Computer Science*, pages 291–296. Springer Berlin / Heidelberg. 10.1007/978-3-540-30483-835. 24

[Lopes et al., 1997] Lopes, H. S., Coutinho, M. S., and Lima, W. C. (1997). An evolutionary approach to simulate cognitive feedback learning in medical domain. In Sanchez, E., Shibata, T., and Zadeh, L. A., editors, *Genetic Algorithms and Fuzzy Logic Systems*, volume 7 of *Advances in Fuzzy Systems - Applications and Theory*, pages 193–207. World Scientific Publishing. 177

[Mahfoud and Mani, 1996] Mahfoud, S. and Mani, G. (1996). Financial forecasting using genetic algorithms. *Applied Artificial Intelligence*, 10:543–565. 22

[Maloof et al., 2003] Maloof, M. A., Langley, P., Binford, T. O., Nevatia, R., and Sage, S. (2003). Improved rooftop detection in aerial images with machine learning. *Mach. Learn.*, 53(1-2):157–191. 88

[Markowska-Kaczmar and Trelak, 2005] Markowska-Kaczmar, U. and Trelak, W. (2005). Fuzzy logic and evolutionary algorithmtwo techniques in rule extraction from neural networks. *Neurocomputing*, 63(0):359 – 379. ¡ce:title¿New Aspects in Neurocomputing: 11th European Symposium on Artificial Neural Networks¡/ce:title¿. 19

[Mashor, 2000] Mashor, M. (2000). Hybrid training algorithm for rbf network. *International Journal of the Computer, the Internet and Management*, 8:50–65. 69

[Massey et al., 2001] Massey, A. P., Montoya-Weiss, M. M., and Holcom, K. (2001). Re-engineering the customer relationship: leveraging knowledge assets at ibm. *Decision Support Systems*, 32(2):155 – 170. 82

[McDonald and Schwing, 1973] McDonald, G. C. and Schwing, R. C. (1973). Instabilities of regression estimates relating air pollution to mortality. *Technometrics*, 15(3):463–481. 11, 39, 106

[McGarry et al., 1999] McGarry, K., Tait, J., Wermter, S., and MacIntyre, J. (1999). Rule-extraction from radial basis function networks. In *Artificial Neural Networks, 1999. ICANN 99. Ninth International Conference on (Conf. Publ. No. 470)*, volume 2. 16

[McMillan et al., 1991] McMillan, C., Mozer, M. C., and Smolensky, P. (1991). The connectionist scientist game: Rule extraction and refinement in a neural network. In *In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pages 424–430. 16

[Mitra et al., 1997] Mitra, S., De, R. K., and Pal, S. K. (1997). Knowledge-based fuzzy mlp for classification and rule generation. *Trans. Neur. Netw.*, 8(6):1338–1350. 18

[Mitra. and Hayashi, 2000] Mitra., S. and Hayashi, Y. (2000). Neuro-fuzzy rule generation: Survey in soft computing framework. *IEEE Transactions on Neural Networks*, 11(3):748 – 768. 19

[Mohamed, 2011] Mohamed, M. H. (2011). Rules extraction from constructively trained neural networks based on genetic algorithms. *Neurocomputing*, 74(17):3180 – 3192. 20

[Mols, 1998] Mols, N. P. (1998). The behavioral consequences of pc banking. *International Journal of Bank Marketing*, 16(5):195–201. 83

[Monard and Batista, 2002] Monard, M. C. and Batista, G. E. (2002). Learning with skewed class distribution. page 173–180, São Paulo, SP. IOS Press. 88

[Moody and Darken, 1989] Moody, J. and Darken, C. J. (1989). Fast learning in networks of locally tuned processing units. *Neural Comput.*, 1:281–294. 69

[Mozer et al., 2000] 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. *Trans. Neur. Netw.*, 11(3):690–696. 83

[Muralidhar and Sarathy, 2006] Muralidhar, K. and Sarathy, R. (2006). Data shufflinga new masking approach for numerical data. *Manage. Sci.*, 52(5):658–670. 33

[Muthu Anbananthen et al., 2007] Muthu Anbananthen, K. S., Sainarayanan, G., Chekima, A., and Teo, J. (2007). Artificial neural network tree approach in data mining. *Malaysian Journal of Computer Science*, 20(1):51 – 62. 17

[Nabney, 2004] Nabney, I. T. (2004). Efficient training of rbf networks for classification. *International journal of neural systems*, 4(3):201–208. 70

[Nath et al., 1997] Nath, R., Rajagopalan, B., and Ryker, R. (1997). Determining the saliency of input variables in neural network classifiers. *Comput. Oper. Res.*, 24(8):767–773. 105

[Nayak et al., 2007] Nayak, H., , Hayward, R., Nayak, R., and Diederich, J. (2007). Using predicates to explain networks. *In: ECAI-2000 Workshop: Foundations of Connectionist-Symbolic Integration: Representation, Paradigms and Algorithms, Berlin, Germany.* 3

[O'Keefe, 1983] O'Keefe, R. A. (1983). Concept formation from very large training sets. In *Proceedings of the Eighth international joint conference on Artificial intelligence - Volume 1*, IJCAI'83, pages 479–481, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 202

[Oliveira and Zaane, 2007] Oliveira, S. R. and Zaane, O. R. (2007). A privacy-preserving clustering approach toward secure and effective data analysis for business collaboration. *Computers and amp; Security*, 26(1):81 – 93. 30

[Olmeda and Fernndez, 1997] Olmeda, I. and Fernndez, E. (1997). Hybrid classifiers for financial multicriteria decision making: The case of bankruptcy prediction. *Computational Economics*, 10:317–335. 9, 39, 73, 106, 178

[Ozbakir et al., 2008] Ozbakir, L., Baykaso lu, A., and Kulluk, S. (2008). Learning and intelligent optimization. chapter Rule Extraction from Neural Networks Via Ant Colony Algorithm for Data Mining Applications, pages 177–191. Springer-Verlag, Berlin, Heidelberg. 17

[Ozbakir et al., 2009] Ozbakir, L., Baykaso lu, A., Kulluk, S., and Yapi, H. (2009). Taco-miner: An ant colony based algorithm for rule extraction from trained neural networks. *Expert System with Applications*, 36(10):12295–12305. 23

[Pan and Chen, 2008] Pan, C. and Chen, W.and Yun, Y. (2008). Fault diagnostic method of power transformers based on hybrid genetic algorithm evolving wavelet neural network. *Electric Power Applications, IET*, 2(1):71–76. 100, 104

[Papagelis and Kalles, 2000] Papagelis, A. and Kalles, D. (2000). Ga tree: Genetically evolved decision trees. In *Tools with Artificial Intelligence, 2000. ICTAI 2000. Proceedings. 12th IEEE International Conference on*, pages 203–206. 199

[Papagelis and Kalles, 2001] Papagelis, A. and Kalles, D. (2001). Breeding decision trees using evolutionary techniques. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 393–400, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc. 199

[Parpinelli et al., 2002] Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. (2002). Data mining with an ant colony optimization algorithm. *Trans. Evol. Comp*, 6(4):321–332. 23

[Pearl, 1978] Pearl, J. (1978). Entropy, information and rational decisions. *Cognitive Systems Laboratory, University of California, Los Angeles.* Technical report. 202

[Penrose et al., 1985] Penrose, K., A.G.Nelson, and Fisher, A. (1985). Generalized body composition prediction equation for men using simple measurement techniques. 10, 39, 106

[Peppers and Rogers, 1995] Peppers, D. and Rogers, M. (1995). A new marketing paradigm. *Planning Review*, 23(2):14–18. 81

[Pinkas, 2002] Pinkas, B. (2002). Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):12–19. 33

[Pramodh and Ravi, 2007] 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. 35

[Provost, 2000] Provost, F. (2000). Machine learning from imbalanced data sets 101. *Proceedings of the AAAI-2000 Workshop on Imbalanced Data Sets*. 88

[Quinlan, 1983] Quinlan, J. R. (1983). *Learning efficient classification procedures and their application to chess endgames. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell, (Eds.), Machine learning: An artificial intelligence approach. Palo Alto: Tioga Publishing Company.* 202

[Quinlan, 1986] Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106. 175, 202

[Quinlan, 1993] Quinlan, J. R. (1993). *C4.5: Programs for machine learning.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 202

[Rahimian et al., 1996] Rahimian, E., Singh, S., Thammachote, T., and Virmani, R. (1996). Bankruptcy prediction by neural network, in trippi-turban (eds.) neural networks in finance and investing, irwin professional publishing, burr ridge, usa. 9, 39, 73, 106, 178

[Ramu and Ravi, 2009] Ramu, K. and Ravi, V. (2009). Privacy preservation in data mining using hybrid perturbation methods: an application to bankruptcy prediction in banks. *Int. J. Data Anal. Tech. Strateg.*, 1(4):313–331. 32

[Ravi et al., 2008] Ravi, V., Kumar, P. R., Srinivas, E. R., and Kasabov, N. K. (2008). A semi-online training algorithm for the radial basis function neural networks : applications to bankruptcy prediction in banks. pages 243–260. 70

[Ravi and Pramodh, 2010] Ravi, V. and Pramodh, C. (2010). Non-linear principal component analysis-based hybrid classifiers: an application to bankruptcy prediction in banks. *International Journal of Information and Decision Sciences*, 2(1):50–67. 35

[Saad and II, 2007] Saad, E. W. and II, D. C. W. (2007). Neural network explanation using inversion. *Neural Networks*, 20(1):78 – 93. 20

[Saito and Nakano, 1988] Saito, K. and Nakano, R. (1988). Medical diagnostic expert system based on pdp model. In *Neural Networks, 1988., IEEE International Conference on*, pages 255 –262 vol.1. 3, 6

[Saito and Nakano, 2002] Saito, K. and Nakano, R. (2002). Extracting regression rules from neural networks. *Neural Networks*, 15(10):1279 – 1288. 16

[Sarimveis et al., 2003] Sarimveis, H., Alexandridis, A., and Bafas, G. (2003). A fast training algorithm for rbf networks based on subtractive clustering. *Neurocomputing*, 51(0):501 – 505. 70

[Sarimveis et al., 2002] Sarimveis, H., Alexandridis, A., Tsekouras, G., and Bafas, G. (2002). A fast and efficient algorithm for training radial basis function neural networks based on a fuzzy partition of the input space. *Industrial and Engineering Chemistry Research*, 41(4):751–759. 69

[Sato and Tsukimoto, 2001] Sato, M. and Tsukimoto, H. (2001). Rule extraction from neural networks via decision tree induction. In *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, volume 3, pages 1870 –1875 vol.3. 6

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47. 87

[Setiono, 1997] Setiono, R. (1997). Extracting rules from neural networks by pruning and hidden-unit splitting. *Neural Computation*, 9(1):205–225. 18

[Setiono, 2000a] Setiono, R. (2000a). Extracting m-of-n rules from trained neural networks. *Trans. Neur. Netw.*, 11(2):512–519. 19

[Setiono, 2000b] Setiono, R. (2000b). Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine*, 18(3):205 – 219. 19

[Setiono and Leow, 2000] Setiono, R. and Leow, W. (2000). Fernn: An algorithm for fast extraction of rules from neural networks. *Applied Intelligence*, 12:15–25. 21

[Setiono and Thong, 2004] Setiono, R. and Thong, J. Y. (2004). An approach to generate rules from neural networks for regression problems. *European Journal of Operational Research*, 155(1):239 – 250. 16

[Shaw et al., 2001] Shaw, M. J., Subramaniam, C., Tan, G. W., and Welge, M. E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems*, 31(1):127 – 137. 81

[Shin and Lee, 2002] Shin, K. and Lee, Y. (2002). A genetic algorithm application in bankruptcy prediction modeling. *Expert System with Applications*, 23(3):321–328. 22

[Snchez et al., 2009] Snchez, D., Vila, M., Cerda, L., and Serrano, J. (2009). Association rules applied to credit card fraud detection. *Expert Systems with Applications*, 36(2, Part 2):3630 – 3640. 87

[Storn and Price, 1997] Storn, R. and Price, K. (1997). Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *J. of Global Optimization*, 11(4):341–359. 190

[Su et al., 2010] Su, H., Yang, Y., and Zhao, L. (2010). Classification rule discovery with de/qde algorithm. *Expert Syst. Appl.*, 37(2):1216–1222. 24

[Sweeney, 2002] Sweeney, L. (2002). k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570. 33

[Takagi and Sugeno, 1985] Takagi, T. and Sugeno, M. (1985). Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132. 217

[Tan et al., 2007] Tan, S. C., Lim, C. P., and Rao, M. V. C. (2007). A hybrid neural network model for rule generation and its application to process fault detection and diagnosis. *Eng. Appl. Artif. Intell.*, 20(2):203–213. 17

[Thrun, 1995] Thrun, S. (1995). Extracting rules from artificial neural networks with distributed representations. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS 7*, pages 505–512. MIT Press. 3, 6

[Thuraisingham, 2003] Thuraisingham, B. (2003). Web data mining and its applications in business intelligence and counter-terrorism. CRC Press. 30

[Thuraisingham, 2004] Thuraisingham, B. (2004). Data mining for counter-terrorism. in H. Kargupta et al. (Eds.), AAAI Press. 30

[Tickle et al., 1998] 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 networks. *Neural Networks, IEEE Transactions on*, 9(6):1057 –1068. 3, 4, 7

[Tokinaga et al., 2005] Tokinaga, S., Lu, J., and Ikeda, Y. (2005). Neural network rule extraction by using the genetic programming and its applications to explanatory classifications. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, E88-A(10):2627–2635. 17

[Towell and Shavlik, 1993] Towell, G. G. and Shavlik, J. W. (1993). Extracting refined rules from knowledge-based neural networks. *Mach. Learn.*, 13:71–101. 4

[Vaidya et al., 2006] Vaidya, J., Clifton, C. W., and Zhu, Y. M. (2006). Privacy preserving data mining. *Advances in Information Security*, 19. 32

[Verhoef and Donkers, 2001] Verhoef, P. C. and Donkers, B. (2001). Predicting customer potential value an application in the insurance industry. *Decision Support Systems*, 32(2):189 – 199. ¡ce:title¿Decision Support Issues in Customer Relationship Management and Interactive Marketing for E-Commerce¡/ce:title¿. 81

[Verykios et al., 2004] Verykios, V. S., Bertino, E., Fovino, I. N., Provenza, L. P., Saygin, Y., and Theodoridis, Y. (2004). State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1):50–57. 31

[Vinay Kumar et al., 2008] Vinay Kumar, K., Ravi, V., Carr, M., and Raj Kiran, N. (2008). Software development cost estimation using wavelet neural networks. *J. Syst. Softw.*, 81(11):1853–1867. 100

[Visa and Ralescu, 2005] Visa, S. and Ralescu, A. (2005). Issues in Mining Imbalanced Data Sets - A Review Paper. pages 67–73. 87

[Wang and Fu, 2005] Wang, L. and Fu, X. (2005). A simple rule extraction method using a compact rbf neural network. In Wang, J., Liao, X., and Yi, Z., editors, *Advances in Neural Networks ISNN 2005*, volume 3496 of *Lecture Notes in Computer Science*, pages 91–118. Springer Berlin / Heidelberg. 17

[Weiss, 1995] Weiss, G. M. (1995). Learning with rare cases and small disjuncts. pages 558–565. 87

[Weiss, 2004] Weiss, G. M. (2004). Mining with rarity: a unifying framework. *SIGKDD Explor. Newsl.*, 6(1):7–19. 88

[Weiss and Provost, 2003] Weiss, G. M. and Provost, F. (2003). Learning when training data are costly: the effect of class distribution on tree induction. *J. Artif. Int. Res.*, 19(1):315–354. 89

[Widrow et al., 1994] Widrow, B., Rumelhart, D. E., and Lehr, M. A. (1994). Neural networks: applications in industry, business and science. *Commun. ACM*, 37(3):93–105. 1

[Wilson, 1972] Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, 2(3):408 –421. 88

[Xiao-dan et al., 2005] Xiao-dan, W., Dian-min, Y., Feng-li, L., Yun-feng, W., and Chao-Hsien, C. (2005). Privacy preserving bio-terrorism surveillance. 30

[Xiao-dan et al., 2006] Xiao-dan, W., Dian-min, Y., Feng-li, L., Yun-feng, W., and Chao-Hsien, C. (2006). Privacy preserving data mining algorithms by data distortion. pages 223 – 228. 29

[Yang, 2010] Yang, X. (2010). Firefly algorithm, stochastic test functions and design optimisation. *Int. J. Bio-Inspired Comput.*, 2(2):78–84. 194, 195, 198

[Yang, 2009] Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. In *Proceedings of the 5th international conference on Stochastic algorithms: foundations and applications*, SAGA'09, pages 169–178, Berlin, Heidelberg. Springer-Verlag. 194, 195, 198

[Yao, 1982] Yao, A. C. (1982). Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, SFCS '82, pages 160–164, Washington, DC, USA. IEEE Computer Society. 34

[Yao, 1986] Yao, A. C.-C. (1986). How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, SFCS '86, pages 162–167, Washington, DC, USA. IEEE Computer Society. 34

[Yuan and Chang, 2001] 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. 81

[zbakr et al., 2010] zbakr, L., Baykasolu, A., and Kulluk, S. (2010). A soft computing-based approach for integrated training and rule extraction from artificial neural networks: Difaconn-miner. *Applied Soft Computing*, 10(1):304 – 317. 21

[zbakr et al., 2011] zbakr, L., Baykasolu, A., and Kulluk, S. (2011). Rule extraction from artificial neural networks to discover causes of quality defects in fabric production. *Neural Computing and Applications*, 20:1117–1128. 10.1007/s00521-010-0434-0. 23

[Zhang et al., 2005] Zhang, N., Zhao, W., and Chen, J. (2005). Performance measurements for privacy preserving data mining. In Ho, T., Cheung, D., and Liu, H., editors, *Advances in Knowledge Discovery and Data Mining*, volume 3518 of *Lecture Notes in Computer Science*, pages 327–349. Springer Berlin / Heidelberg. 31

[Zhang, 1997] Zhang, Q. (1997). Using wavelet network in nonparametric estimation. *Neural Networks, IEEE Transactions on*, 8(2):227 –236. 100

[Zhang and Benveniste, 1992] Zhang, Q. and Benveniste, A. (1992). Wavelet networks. *Neural Networks, IEEE Transactions on*, 3(6):889 –898. 100

[Zhang et al., 2001] Zhang, X., Qi, J., Zhang, R., Liu, M., Hu, Z., Xue, H., and Fan, B. T. (2001). Prediction of programmed temperature retention values of naphthas by wavelet neural networks. *Computers and Chemistry*, pages 125–133. 100, 102

[Zhao et al., 2006] Zhao, X., Zeng, J., Gao, Y., and Yang, Y. (2006). Particle swarm algorithm for classification rules generation. In *Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications - Volume 02*, ISDA '06, pages 957–962, Washington, DC, USA. IEEE Computer Society. 24