Medical Outcome Prediction:

A Hybrid Artificial Neural Networks Approach

By

Fariba Shadabi

A thesis submitted in fulfilment of the requirements of the
Degree of Doctor of Philosophy in Information Sciences and Engineering

The University of Canberra

January 2007

Abstract

This thesis advances the understanding of the application of artificial neural networks ensemble to clinical data by addressing the following fundamental question: What is the potentiality of an ensemble of neural networks models as a filter and classifier in a complex clinical situation?

A novel neural networks ensemble classification model called Rules and Information Driven by Consistency in Artificial Neural Networks Ensemble (RIDC-ANNE) is developed for the purpose of prediction of medical outcomes or events, such as kidney transplants. The proposed classification model is based on combination of initial data preparations, preliminary classification by ensembles of Neural Networks, and generation of new training data based on criteria of highly accuracy and model agreement. Furthermore, it can also generate decision tree classification models to provide classification of data and the prediction results. The case studies described in this thesis are from a kidney transplant database and two well-known collections of benchmark data known as the Pima Indian Diabetes and Wisconsin Cancer datasets. An implication of this study is that further attention needs to be given to both data collection and preparation stages. This study revealed that even neural network ensemble models that are known for their strong generalization ability might not be able to provide a high level of accuracy for complex, noisy and incomplete clinical data. However, by using a selective subset of data points, it is possible to improve the overall accuracy.

In summary, the research conducted for this thesis advances the current clinical data preparation and classification techniques in which the task is to extract patterns that contain higher information content from a sea of noisy and incomplete clinical data, and build accurate and transparent classifiers. The RIDC-ANNE approach improves an analyst’s ability to better understand the data. Furthermore, it shows great promise for use in clinical decision making systems. It can provide us with a valuable data mining tool with great research and commercial potential.

Acknowledgements

First and foremost, I would like to thank my supervisor, Associate Professor Dharmendra Sharma, who has been my advisor throughout my graduate studies at the University of Canberra. His advice and critical comments were invaluable during the years that it took to bring this study to completion. In spite of a busy schedule, he always found time to answer my questions. I am deeply grateful to him.

I would also like to thank the other members of my thesis committee, Professor Nikolai Petrovsky and Professor Simon Hawkins, for their helpful discussions on the ideas in the work and their support and encouragement over the years.

I would like to express my gratitude to the members of the School of Information Sciences and Engineering and School of Health Sciences at the University of Canberra who provided support and useful information along the way. In particular, I would like to express my deep gratitude and appreciation to Mr Robert Cox for the encouragement, insightful technical comments and help with neural networks related matters. Special thanks must also go to Ms Rebecca Booth for her editorial services.

Last, but not least, a great deal of thanks must go to my family, particularly to my parents for the encouragement of my educational pursuits throughout my life and my husband Mehrdad for his empathic understanding and constant support throughout my graduate education. One of the best experiences that we had during this period was the birth of our first child, Anahita. It was truly a memorable and wonderful experience.

Contents

ABSTRACT ..... **III**

ACKNOWLEDGEMENTS ..... **V**

LIST OF FIGURES ..... **VIII**

LIST OF TABLES ..... **IX**

1. INTRODUCTION **1**

1.1. MOTIVATION 2
1.2. OBJECTIVES 5
1.3. THESIS OUTLINE 6

2. KNOWLEDGE SYSTEMS FOR CLINICAL DECISION MAKING **7**

2.1. INTRODUCTION 7
2.2. KNOWLEDGE DISCOVERY SYSTEMS FOR MEDICAL TASKS 8
2.3. DATA MINING TECHNIQUES FOR SUPPORTING MEDICAL TASKS 13
2.4. MACHINE LEARNING TECHNIQUES 20
2.4.1. Artificial neural networks (ANN) for clinical data *21*
2.4.2. Decision trees for clinical data *23*
2.5. RECENT RESEARCH DIRECTIONS: ML-BASED HYBRID SYSTEMS 23
2.5.1. Extracting rules from neural networks *26*
2.5.2. Extracting rules from a neural network ensemble *32*
2.6. SUMMARY 35

3. PROPOSED HYBRID ANN APPROACH: RIDC-ANNE **39**

3.1. INTRODUCTION 39
3.2. MATERIALS AND METHODS: 40
3.2.1. Clinical Data *40*
3.2.2. Artificial neural networks (ANN) *41*
3.2.3. Decision trees *45*
3.2.4. Ensemble classifiers *47*
3.3. THE RIDC-ANNE METHODOLOGY 48
3.4. THE EVALUATION AND ASSESSMENT PROCEDURE 52
3.5. DISCUSSION: WHY THE MODEL WORKS 53
3.6. SUMMARY 54

4. USE OF ANN FOR MEDICAL OUTCOME PREDICTION **57**

4.1. INTRODUCTION 57
4.2. CLINICAL PROBLEM DOMAINS 57
4.3. RESULTS FOR KIDNEY TRANSPLANT DATA USING SINGLE MLP 61
4.3.1. Experiment 1: replicate the previous experimental results *62*
4.3.2. Experiment 2: using the previous pre-processed inputs and a MLP *62*
4.3.3. Experiment 3: pre-processing with slightly different methodology *63*
4.3.4. Experiment 4: search for a Subset using a single MLP *64*
4.3.5. Experiment 5: pre-processed the data further using a single MLP *65*

4.4. RESULTS FOR KIDNEY TRANSPLANT DATA USING RIDC-ANNE 66
4.4.1. 100-MLP classifiers without bagging 67
4.4.2. 100-MLP classifiers with bagging 68
4.4.3. 100-MLP classifiers with all available success points without bagging 70
4.4.4. 100-MLP classifiers with all available success points and bagging 72
4.4.5. 500-MLP classifiers with all available success points and bagging 74
4.5. RESULTS FOR UCI DATASETS USING RIDC-ANNE 77
4.6. SUMMARY 81
5. CONCLUSION AND FUTURE WORK 83
APPENDICES 91
APPENDIX A 92
LIST OF PUBLICATIONS 92
APPENDIX B 93
DESCRIPTIVE STATISTICS OF DATASETS 93
APPENDIX C 95
A SAMPLE OF KIDNEY TRANSPLANT DATA 95
APPENDIX D 96
GRAPHIC DEMONSTRATIONS OF RIDC-ANNE 96
APPENDIX E 101
DETAILED EXPERIMENTAL RESULTS 101
BIBLIOGRAPHY 110

List of Figures

Figure 2.1: Some of the application areas of artificial intelligence 8
Figure 2.2: Basic components of an expert system (Giarratano and Riley, 1998). 9
Figure 2.3: What data mining can offer. 13
Figure 2.4: Data preparation process of data mining. 14
Figure 2.5: Bagging (bootstrapping aggregates) 17
Figure 2.6: Estimating classifier accuracy with the holdout scheme. 17
Figure 2.7: Overview of Machine Learning Strategy. 21
Figure 2.8: The TREPAN algorithm 29
Figure 2.9: A representative tree for the DNA coding domain (Craven, 1996) 30
Figure 2.10: A rule set example generated by GP algorithm (Blackmore and Bossomaier, 2003). 31
Figure 2.11: A black box (global) explanation approach 33
Figure 2.12: A component-based (local) explanation approach 34
Figure 3.1: The structure of a neuron with three inputs and one output 42
Figure 3.2: A simple classification tree for two classes (Accept for a home loan or Reject) 46
Figure 3.3: The general structure for a neural network ensemble 53
Figure 4.1: The percentage of missing data points in fail and success target category for the 23 variables 63
Figure 4.2: The percentage of missing data points in fail and success target category for the 23 variables after the new pre-processing strategy. 64
Figure 4.3: The results for 100-MLP using the first pre-processing strategy. 68
Figure 4.4: The results for 100-MLP using the first pre-processing strategy (with bagging). 70
Figure 4.5: The results for 100-MLP using the second pre-processing strategy. 72
Figure 4.6: The results for 100-MLP using the second pre-processing strategy (with bagging). 74
Figure 4.7: The RIDC-ANNE algorithm (where n or No. of Networks is 500) 75
Figure 4.8: The results for 500-MLP using the second pre-processing strategy (with bagging). 76
Figure 4.9: The back-end of RIDC-ANNE process 77
Figure 4.10: The results for the Pima Indian diabetes data with 100 bagging. 78
Figure 4.11: The results for the Wisconsin cancer dataset with 100 bagging 81
Figure D. 1: 100- MLP classifiers without bagging (Exp 4.3.1) 97
Figure D.2: 100- MLP classifiers with bagging (Exp 4.3.2) 98
Figure D.4: 100- MLP classifiers with all success data & without bagging (Exp 4.3.3) 99
Figure D.5: 100- MLP classifiers with all success data & with bagging (Exp 4.3.4) ... 100

List of Tables

Table 3.1: The RIDC-ANNE algorithm 51
Table 4.1: The main variables selectd for the purpose of this study. 60
Table 4.2: The results for experiment 4, using MLP network. 65
Table 4.3 : A comparison table of sensitivity, specificity and accuracy 80
Table B.1: Descriptive statistics for the main input variables used from Kidney Transplants database 94
Table B.2: Descriptive statistics for the input variables used from the Pima Indian Diabetes dataset 94
Table B.3: Descriptive statistics for the input variables used from the Wisconsin Breast Cancer dataset 94
Table E.1: The results for 100-MLP classifiers without bagging- Kidney Transplant 102
Table E.2: The results for 100-MLP classifiers with bagging- Kidney Transplant 103
Table E.3: The results for 100-MLP classifiers with all available success points without bagging- 104
Table E.4: The results for 100-MLP classifiers with all available success points and bagging- Kidney Transplant 105
Table E.5: The results for 500-MLP classifiers with all available success points and bagging- Kidney Transplant 106
Table E.6: The results for the Pima Indian Diabetes dataset with 100 bagging 108
Table E.7: The results for the Wisconsin Cancer dataset with 100 bagging 109

Chapter 1

Introduction

The economic and social benefits of accurately predicting medical outcomes are very high. As a result, the problem of improving predictive models has attracted many researchers. Over the past few years there has been great interest in the use of data mining tools across the healthcare spectrum.

Until now, most clinical data analysis has largely relied on the use of standard traditional strategies such as logistic regression (Hosmer and Lemeshow, 1989) for knowledge discovery or predicting medical outcomes. Statistical techniques have been used successfully in a number of medical domains (McCance, et al., 1993; Bagley, White and Golomb, 2001). However, they do not always have the capacity for solving problems of high complexity (Freeman, et al., 2000; Heckerling, et al., 2003; Santos-García, et al., 2004; Jaimes, et al., 2005). Consequently, researchers turned their attention to the search for alternative data modelling and prediction techniques that would result in better performance, simpler implementation and adoption in clinical practices.

Many researchers have recently turned their attention to Machine Learning (ML) methods to mimic human functions (such as learning capacity and adaptation to changes) and to create computer programs for analysis of datasets (Lette, et al., 1994; T. K Sen, Oliver and N Sen, 1995; Tu, 1996; Pesonen, 1997; Schwartz, et al., 1997). ML largely overlaps with Knowledge Discovery from Databases (KDD) and Data Mining (DM) in the sense that they all deal with extraction of previously unknown knowledge and patterns from large databases and using it in the decision-making process. As a result, these terms are often used interchangeably in the literature. More information about these terms and techniques can be found in Sections 2.3 and 2.4.

Research shows that ML techniques can be applied in healthcare environments where an automated process must adapt to changing conditions, improve its performance based on previous data, extract knowledge from examples in a database,

and deal with uncertain and incomplete medical knowledge (Safavin and Landgrebe, 1991; Rudolfer, Paliouras and Peers, 1999; Berrar, et al., 2003; Li, et al., 2004).

There are many different types of clinical tasks to which ML tools can be applied. For example ML tools can assist detection of microcalcifications in mammography (Woods, et al., 1993), analyse Sudden Infant Death Syndrome (SIDS) (Wilks and English, 1994) and diagnose thyroid disorders (File, 1994).

The underlying purpose of this thesis is to investigate, implement, and customise a ML technique namely, Artificial Neural Networks (ANN) for the purpose of predicting medical events. ANN models have been used by many researchers and provided good predictive accuracy in a wide variety of domains (Baxt, 1991; Ashutosh, et al., 1992; Tu and Guerriere, 1993; Baxt, 1995; Mobley, Leasure and Davidson, 1995; Ortiz, Ghefter and Silva, 1995; Itchhaporia, Snow and Almassy, 1996; LaPuerta, L'Italien and Paul, 1998; Pantazopoulos, et al., 1998; Dayhoff and DeLeo, 2001). Unfortunately they often produce models that are very difficult to understand. This thesis also tackles the problem of complexity of clinical data and non-transparency of neural network models by developing an algorithm with reference to the bagging-based ensemble and the hybrid decision tree-neural networks ensemble. This strategy tries to identify the regions in the data space that have high impact on system performance and consider the diversity and expertise of the component networks in the rule generation process. The case studies described in this thesis are from a kidney transplant database (a complex scenario) and two well-known collections of benchmark data known as the Pima Indian Diabetes (a semi-complex scenario) and Wisconsin Cancer datasets (a non-complex scenario). The Pima Indian Diabetes and Wisconsin Cancer datasets are stored in the UCI repository (Mangasarian and Wolberg, 1990; UCI Repository). These case studies provide examples of the challenges involved in real life data mining in clinical settings.

1.1. Motivation

Data mining and ML tools can support clinical data analysis. For the decision process in medicine the ideal ML tool should be able to learn the underlying relationships presented in the clinical data, extract maximal predictive information and clearly reveal new patterns that indicate certain outcomes or treatments.

The motivation for this thesis is to develop new techniques by expanding and reusing existing ML techniques for the purpose of predicting the outcome of medical procedures or events. The main reasons for this include:

i) A great deal of research effort has been spent on the use of standard statistical models. ML technology is substantially less mature than the more commonly used statistical models. However, research shows that traditional statistical techniques can not always solve clinical problems. This is usually due to the inherent complexity contained in the clinical data and the considerable amount of normal disparity across patients. A literature search on ML technique revealed that ANN models are powerful techniques and becoming widely accepted in many medical sectors (Doyle, et al., 1994; Ortiz et al., 1995; Ennett, Frize and Walker, 2001; Ramesh, et al., 2004; Gabutti, et al., 2006; Mueller, et al., 2006). Recently, Rajimehr et al. (2002), successfully used an ANN model to predict lupus nephritis in patients with systemic lupus erythematosus. Other groups (Freeman et al., 2000; Heckerling et al., 2003; Eftekhar, et al., 2005; Jaimes et al., 2005) have also reported success when comparing ANN to traditional logistical regression models for prediction of complex medical outcomes. These outcomes include in-hospital death after percutaneous transluminal coronary angioplasty, community-acquired pneumonia, mortality in head trauma, and mortality in patients with suspected sepsis in the emergency room. Therefore, this thesis is well placed to further explore the potential of ANN models in medical analysis for decision support and complex prediction tasks.

ii) Little evidence exists about the application of ANN in the task of kidney transplant outcome prediction. As outlined previously, ANN models are powerful tools for prediction. Furthermore, they have the ability to provide good solutions in complex situations where a large number of variables contribute to an outcome but their individual influence is not well understood. Clinical data gathered from patients with diabetes and kidney diseases or patients who have undergone kidney transplant surgery have this

characteristic and are known to be complex (Doyle et al., 1994; Liberati and Setti, 1994; Matis, et al., 1995; Sheppard, et al., 1999). Over the years there has been substantial research into predicting graft outcomes and detecting key parameters influencing the graft outcome (Doyle et al., 1994; Rapaport, 1995; Dorsey, et al., 1997; Michael, et al., 2003). However, literature search revealed little evidence about application of ANN models for the prediction of graft outcomes after kidney transplantation (Petrovsky, et al., 2002; Michael et al., 2003). Michael et al. (2003), compared ANN with traditional logistical regression models for prediction of the occurrence of delayed graft function in cadaveric renal transplants. Their results revealed that logistic regression was 36.5% sensitive and 90.7% specific compared to the ANN which was 63.5% sensitive and 64.8% specific. Other groups (Matis, 1995) used a MLP network on 290 liver transplantation patients to predict graft outcomes. However, they used the networks for liver transplant outcome prediction only. They used 240 records for training purpose and 55 records for testing purposes. Their system was able to accurately predict 88% of graft failures and 98% of graft survival cases. Overall, they concluded that the ANN model was able to provide a better level of sensitivity than logistic regression methods in these cases.

iii) There is a need for identifying suitable observations in large and noisy clinical datasets. Clinical data are also known to be complex due to genetic and biological diversity of individuals, disease marker variability amongst individuals, the variability in the combination of drugs given to each individual and missing data, to name a few. The rapid development of data collection and storage technologies has led to the formation of large number of clinical databases. Large clinical databases might contain much useful knowledge; however, computational efficiency, especially for powerful but processing-intensive methods, such as ANN can be a major concern. Also, it is well known that employing a methodology to develop a best possible representation of the structure of the data and choosing the right set of data is the key to a successful data mining. Therefore, it is desirable to have a

method at hand to extract data points that contain higher information content from a sea of noisy and incomplete clinical data and build better classifiers.

iv) There is a need for extracting explanations from several 'black box' connectionist learning systems. Powerful ML techniques such as neural network ensembles have the ability to provide good solutions for complex situations. However, ANN models are generally known as 'black boxes' as the process by which the outputs are produced is not obvious. Recently, there has been extensive research on hybrid decision tree-neural networks approaches, in which the goal is to extract useful explanations from individual ANN models by combining the power of black box connectionist learning systems with transparent rule based decision-making methodologies. As yet, there has been little research and development in the explanation of several combined networks.

1.2. Objectives

This thesis presents and evaluates a novel hybrid neural networks model, called Rules and Information Driven by Consistency in Artificial Neural Networks Ensemble (RIDC-ANNE) that has been designed and developed during this period of study for the purpose of prediction of medical outcomes, such as kidney transplants. The proposed classification model is based on combination of initial data preparations, preliminary classification by ensembles of ANNs, and generation of new training data based on criteria of highly accuracy and model agreement. Furthermore, the model will also generate the decision tree models to provide classification of data and the prediction results.

This approach attempts to extract high quality data by configuring an ensemble of bagged networks to filter and identify the regions in the data space that have been consistently misclassified or have high impact on system performance. The RIDC-ANNE approach has the ability to extract the input patterns (examples) that were included across the neural networks series in the final results. It can also be used to provide some clarity for the general behaviour of an ensemble. Overall, this study is

concerned with a fundamental question: What is the potentiality of an ensemble of neural networks models as a filter and classifier in a complex clinical situation?

This thesis also investigates the advantages and disadvantages associated with selective ML techniques to better understand their potential.

1.3. Thesis outline

The remainder of this thesis is organised as follows:

Chapter 2 introduces further background material by concentrating on important concepts and techniques used in further chapters. It starts with an overview of work reported in the scientific literature followed by a description of knowledge discovery and data mining approaches, especially those which make use of ML techniques, and discusses some of their advantages and limitations. By a critical discussion of these techniques, we provide a prelude and justification for the research presented in the later chapters.

Chapter 3 introduces the novel hybrid learning approach, RIDC-ANNE, and describes methods and real life clinical datasets used as examples in this thesis.

Chapter 4 applies a selection of data mining approaches described in the previous chapters to the selected clinical datasets. In addition, it provides some comments about the results and shows how prediction models can be improved by placing more attention on input data selection and pre-processing techniques.

Chapter 5 discusses the contribution of this thesis, identifies the limitations of the work presented, and identifies opportunities for future work.

Finally, five appendices list further details. Appendix A provides a list of publications that have been published to date from the thesis and during this period of study. Appendix B displays the description summary of the main variables of the databases used in this study. Appendix C demonstrates a small sample of kidney transplant data used in this thesis. Appendix D graphically demonstrates the front end of the RIDC-ANNE technique. Appendix E demonstrates the results of this study in detail.

Chapter 2

Knowledge systems for clinical decision making

We are drowning in information and starving for knowledge.

Rutherford D. Roger (1985)

2.1. Introduction

In recent years there has been rapid growth in the successful use of Machine Learning (ML) systems in many diverse areas such as science, medicine and commerce. The main contributing factor influencing the development of such systems in medicine has been the additional demands for a more powerful, yet flexible and transparent technique, to cope with the extensive amount of data and knowledge stored in clinical databases.

This chapter considers a longstanding problem of clinical knowledge discovery and decision making process that is, intelligently processing large amounts of clinical and medical data. The primary aim of this chapter is to draw together previous relevant works and recent research results, with a view towards developing an ML based methodology for the generation of better classification and prediction systems in medical domains. We begin with a brief background of traditional computer-based medical decision making and knowledge discovery systems followed by the statistical and ML based knowledge discovery and data mining techniques. Having posed the idea, let us study how these techniques could be improved within an actual working environment and shed light on many common challenges and expectations. It is not the intention of this study to describe all knowledge discovery and data mining techniques but to concentrate only on relevant concepts and techniques.

2.2. Knowledge discovery systems for medical tasks

From the early 1950s, scientists and healthcare professionals have been aware of the potential of computer systems in medicine. In 1959, Ledley and Lusted published a paper entitled, "*The reasoning foundations of medical diagnosis*" (Ledley and Lusted, 1959). In this paper they point out the potential for computer programs to assist physicians with diagnosis tasks. They applied their idea by using mathematical methods of Boolean algebra and symbolic logic to solve medical problems.

One of the earliest computer-based knowledge discovery systems was DENDRAL (Buchanan and Feigenbaum, 1978; Lindsay, et al., 1980). DENDRAL (DENDRitic Algorithm) was developed between 1965 and 1983 by a group of chemists, geneticists, and computer scientists. In particular, the DENDRAL project was started by Edward Feigenbaum, Joshua Lederberg (a Nobel Prize-winning geneticist and biochemist) and Bruce Buchanan as an effort to aid organic chemists in predicting the molecular structures of unknown organic compounds. DENDRAL uses ML methodology to manipulate the rules and decide which candidates can be considered as a plausible candidate structures for new or unknown chemical compounds.

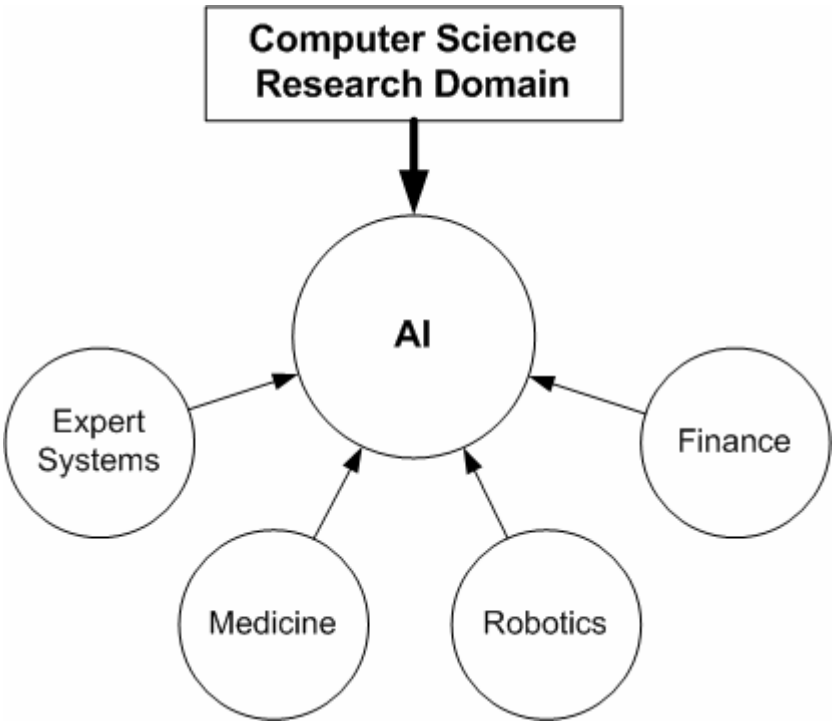

Figure 2.1: Some of the application areas of artificial intelligence

The DENDRAL project pioneered the use of rule-based reasoning strategy which then grew into knowledge engineering tools, today known as expert systems. The terms expert system, knowledge-based system, or knowledge-based expert system are often used synonymously (Waterman, 1985). Expert systems are computer programs that are derived from a branch of computer science research domain known as Artificial Intelligence (AI). AI is an attempt to build intelligent computer programs that behave like humans. AI is a broad discipline with many applications, some of which are shown in Figure 2.1.

Expert systems are designed to act as an intelligent assistant to a user or human expert in one problem domain (Giarratano and Riley, 1998). The general structure of an expert system is shown in Figure 2.2. An expert system usually has three components:

i. External data sources.
ii. A knowledge base (supplied by users), which stores the facts, related to the problem domain, their relations to one another, and perhaps some heuristic knowledge (less rigorous and more experiential). The knowledge within an expert system is generally represented in the form of a set of rules.
iii. A reasoning engine, which operates on the knowledge base and external data sources (performs the knowledge based reasoning) to draw conclusions, answers questions, and gives advice.

The DENDRAL project led to the development of other knowledge-based systems such as META-DENDRAL (Buchanan and Feigenbaum, 1978) and MYCIN (Shortliffe, et al., 1973; Shortliffe, 1976).

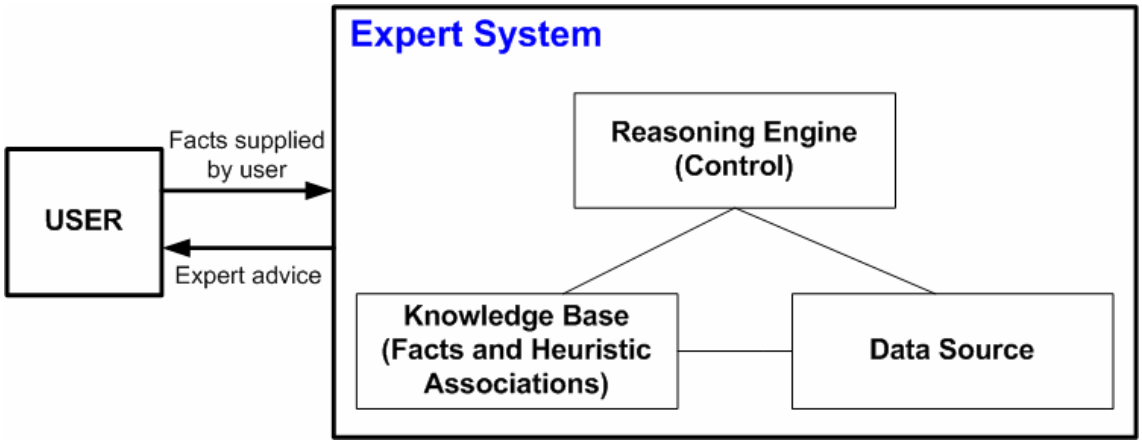

Figure 2.2: Basic components of an expert system (Giarratano and Riley, 1998).

META-DENDRAL (Buchanan and Feigenbaum, 1978) was developed in the early 1970s. This inductive program uses a scientific theory and methodology similar to DENDRAL itself to discover molecular fragmentation rules. META-DENDRAL uses an heuristic search (plan, generate and test) to automatically generate rules for DENDRAL to use in justifying information about unknown chemical compounds. The learning strategy employed in META-DENDRAL was designed to be able to deal simultaneously with multiple concepts and to cope with noisy and incomplete chemical data. Although META-DENDRAL is no longer a productive tool, the learning and discovery ideas that META-DENDRAL introduced into rule-based expert systems have been applied to many new domains.

Another important program, carried out in the early 1970s, was MYCIN (Shortliffe, 1976). MYCIN is one of the most famous examples of early expert systems. MYCIN is an AI based interactive diagnostic program that was developed between 1972 and 1980 at Stanford University to assist practitioners in the selection of an appropriate therapy for patients with certain infectious diseases (Musen, 1999). The system has the ability to prescribe drug treatment and clearly explain its reasoning as a set of IF-THEN rules. In 1979, nine researchers compared the performance of MYCIN system to that of physicians in a small range of infection cases. The study considered ten types of infection such as viral, fungal, and bacterial. For each of these cases, appropriate therapies were obtained from MYCIN, a member of the Stanford Infectious Diseases faculty, a resident medical officer, and a medical student. The evaluation results revealed that MYCIN performed remarkably well, and in some cases, it even outperformed the human doctors (Yu, Fagan and Wraith, 1979). Sadly, although MYCIN performed as well as some domain experts and produced highquality results, due to ethical and legal issues associated with the use of computers in medicine, it was never actually used as a production tool.

Although the MYCIN system was never actually used in diagnosis by clinicians, many other clinical decision support systems were created from this system. NEOMYCIN is an example of one of these support systems (Clancey and Letsinger, 1981). NEOMYCIN is a tree-based problem solving strategy that was developed upon a MYCIN type rule-based system (Sotos, 1990). NEOMYCIN trains doctors by taking them through a broad range of examples of diseases. The system employs a tree-based problem solving strategy that looks at known facts about different kinds of diseases. The

diseases near the top of the tree usually belong to general classes of diseases. NEOMYCIN is then able to differentiate between two disease subclasses by moving down a tree from very general classes of diseases to more precise classes.

EMYCIN (Empty or Essential MYCIN) shell is another additional of MYCIN (Van Melle, 1980; Van Melle, Shortliffe and Buchanan, 1981). EMYCIN is a domain-independent tool that can be used to build rule-based expert systems in which the user must supply only all the necessary knowledge about the task domain. The system was made by removing the medical knowledge base of the MYCIN expert system. The inference engine that applies the knowledge to the task domain is usually built into a shell. Overall, EMYCIN was designed to provide:

i. An easy way for knowledge base development and refinement.
ii. Certainty factor associated with each predication (rather than reporting only true or false for all predications).
iii. Better explanations for its behaviour.

In the late 1970s a program called PUFF was developed from EMYCIN (Aikens, et al., 1983). The program is able to diagnose the presence and seriousness of lung diseases. The PUFF system has the ability to automatically interpret pulmonary function test results and produce a clear report to be kept for future use. The PUFF system is one of the most successful examples of the areas in which expert systems are used and is still in routine use in the clinical laboratories around the world.

Quick Medical Reference (QMR) is another diagnostic decision support system that was developed in the 1970s at the University of Pittsburgh to help physicians diagnose adult disease (Miller, Masarie and Myers, 1986). However, it was only in the early 1980s that the major progress in the development and use of QMR in hospitals and health office practices occurred. The diagnosis process of QMR is based on historical diseases and findings. The QMR knowledge base contains information on more than 700 diseases and more than 4500 clinical findings. Like many other earlier expert systems, this program is currently used by medical students and professionals as an interactive textbook.

In the 1980s, researchers made major progress in applying the AI-based systems to medical practice (Szolovits, 1982; Clancey and Shortliffe, 1984; Miller et al., 1986). DXplain was developed at the Massachusetts General Hospital (Barnett, et al., 1987)

and is a successful example of 1980's medical textbook and intelligence decision support systems. Given a list of symptoms, signs and laboratory information, DXplain can generate a hierarchical list of all possible diagnoses with clear explanation and suggestions. DXplain is still used by medical students and professionals as a medical education and reference system.

Many expert systems including MYCIN and QMR program were found to provide reliable, automatic methods to deal with the increasing amounts of clinical data. These systems have the ability to generate case-specific advice for clinical decision making (Morio, 1989; Wang and Tseng, 1990). Although there are many variations, expert systems generally combine If-Then production rules with inference engines to create the knowledge base. This rule-based approach is easier to build, and it also has the ability to provide transparent suggestions to the physicians.

However, in the late eighties and early nineties researchers in knowledge-based systems realised that generating and maintaining a large rule-based knowledge acquisition tool can be very labour intensive (Saito and Nakano, 1988; Sestito and Dillon, 1991). This shortcoming was one of the main reasons that limited the broader use of the QMR system. They also discovered that the knowledge acquired from experts alone was unsuitable for solving difficult problems. Hence, there was a need to alter the way in which knowledge was created and stored in expert system, especially for complex fields such as medicine that were known to be knowledge intensive. The newer systems generally required adaptations to changes, better flexibility and possibility of plug-in-able updates and knowledge reuse (Chandrasekaran, 1986; Walther, Eriksson and Musen, 1992). In recent years, knowledge-based systems have begun utilising a set of more powerful AI-based data mining techniques.

The following sections outline the data mining process in general and describe a series of methodologies for training and evaluating data mining models such as neural networks. It also outlines the major reasons for applying data mining techniques to clinical databases and it provides an overview of some of the main data mining techniques. Many of these techniques are currently under active development for use in clinical decision-making and knowledge-based systems.

2.3. Data mining techniques for supporting medical tasks

Data mining can be described as the process of extraction of previously unknown patterns or knowledge from a collection of data (Fayyad, Piatetsky-Shapiro and Smyth, 1996). The Knowledge Discovery in Databases (KDD) process according to (Fayyad et al., 1996) is consisted of the following steps: learning the application domain, creating a target dataset, data cleaning and pre-processing, data reduction and projection, choosing the function of data mining, choosing the data mining algorithm(s), data mining, interpretation and using discovered knowledge.

In practice, the two primary goals of data mining are prediction and description. Prediction involves using defined data variables to predict unknown or future values of other variables of interest. Description focuses on finding human-interpretable patterns (such as rules or data models) describing the data (Fayyad et al., 1996). The goals of prediction and description can be achieved by applying different statistical and machine learning methodologies. Figure 2.3 illustrates what data mining can offer.

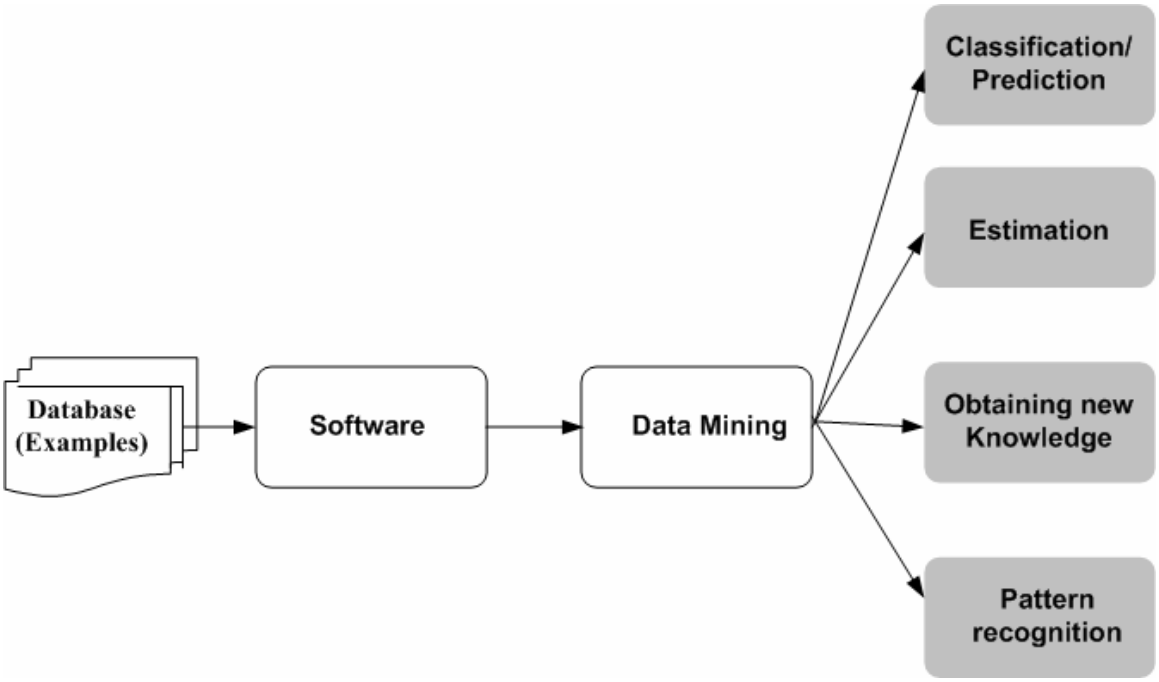

Figure 2.3: What data mining can offer.

The data mining process consists of three stages, data preparation, model learning and model evaluation.

Stage 1: Data preparation

i. The process of analysis usually starts with a set of historical data. The input data can be observational data (e.g. weather records, geographical images), experimental data (e.g. laboratory test results, chemical experimental data), or almost any other types. However, in the real world, data are not always ready for data mining.

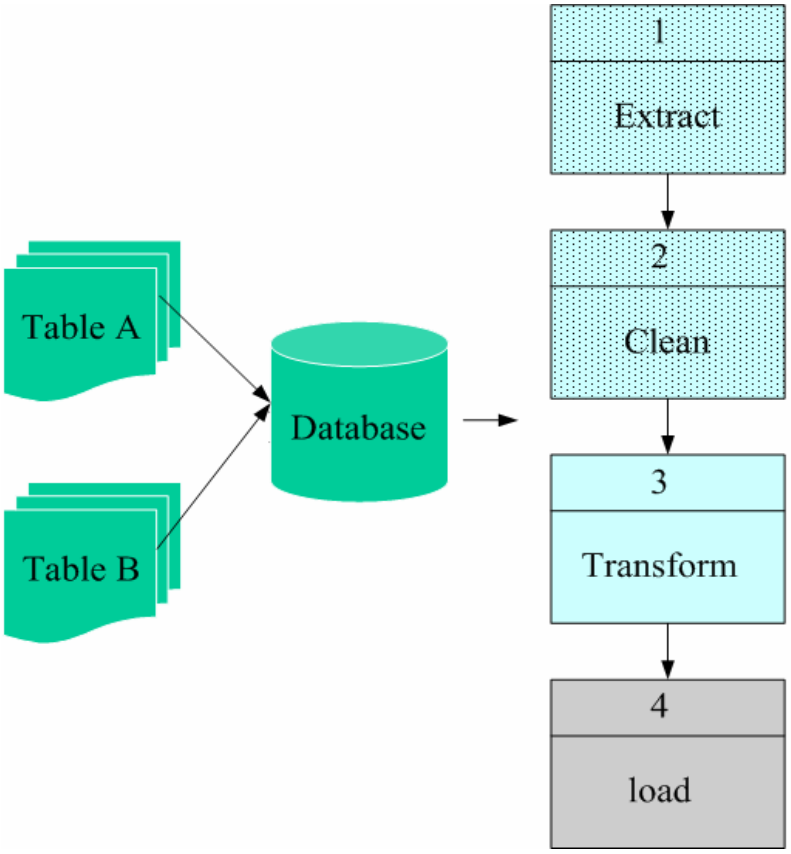

Figure 2.4: Data preparation process of data mining.

As illustrated in Figure 2.4, depending on the nature of the analytic problem, the data preparation stage of data mining may involve:

ii. Gathering data into a central repository

iii. Selecting subsets of records

iv. Selecting subsets of features (variables) that are deemed to be useful. This is particularly useful in situations that datasets are presented with large numbers of

variables and there is a need to bring the number of variables together in a manageable series.

v. Performing some preliminary data cleaning to eliminate possible errors, incomplete data and dubious values

vi. Performing data transformations

vii. Loading the data

Stage 2: Model Learning

This stage involves the application of data mining techniques to learn patterns and potentially useful knowledge from given data. It involves considering different learning techniques and choosing the best techniques based on their performance.

Many current clinical data analyse largely rely on the use of standard statistical techniques (Worsley, et al., 2002; Lange, 2003; Worsley, 2003). Linear regression, non-linear regression, multiple regressions, and stepwise regression are among the most commonly studied form of statistical techniques. In general the purpose of linear regression is to find the best-fit straight line to a set of data points and minimizes the sum of the square of the vertical distances of the data points from the line, while the goal of non-linear regression is to find the values of those variables that form a curve that is closest to the data and minimize the sum of the squares of the vertical distances of the data points from the curve.

Among statistical methods, the most commonly employed in clinical setting is the Logistic Regression (LR). LR (also known as binomial or binary regression) is an effective learning method that can be used on historical data to estimate the probability of a certain event occurring (Hosmer and Lemeshow, 1989). This method is widely used for modelling the relationship between a binary dependent variable (with values such as yes/no or 0/1) and independent (explanatory) variables of any type. Multinomial LR is another type of logistic model that can be used when the dependent variable has more classes than two (for detailed discussions on logistic regression models see (Hosmer and Lemeshow, 1989; Tu, 1996).

The following equation specifies a simple form of logistic prediction model.

$$\log[p/(1-p)] = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + ... + b_n x_n$$

where:

- p is the probability of the outcome of interest (e.g. failure and success events), $x_1, x_2, x_3, ..., x_n$ are the independent variables
- b_0 is an intercept term, and $b_1, b_2, b_3, ..., b_n$ are the coefficients associated with each independent variable

This technique has been used successfully in a number of medical domains, see for example (Long, et al., 1993; McCance et al., 1993; Rudolfer et al., 1999; Bagley et al., 2001; Heckerling et al., 2003).

The choice of a particular algorithm or combination of techniques to apply in a particular situation depends on both the nature of the problem under investigation and the nature of the available data. There is no universally best data modelling or analysis technique across all application domains. However, there are many techniques that can be used to deal with the analysis of data and help improve the accuracy of a given data mining model. Bagging (bootstrapping aggregates) and boosting are two such techniques (Section 2.4.3 gives detailed discussion of these techniques). Bagging is one of the currently favoured methods that has been the subject of many research papers (Baxt and White, 1995; Quinlan, 1996). The basic idea is to combine the outputs of many weak classifiers to generate a powerful committee. As demonstrated in Figure 2.5, each individual classifier is ideally trained with different training data (using a random drawing with replacement strategy). As a result, each classifier may produce a different prediction. Voting strategies are used to combine the predictions of the component classifiers for a given unknown (new) case.

Bagging is particularly useful for an unstable learning algorithm, where a small change in the training set can lead to an obvious change in the model produced.

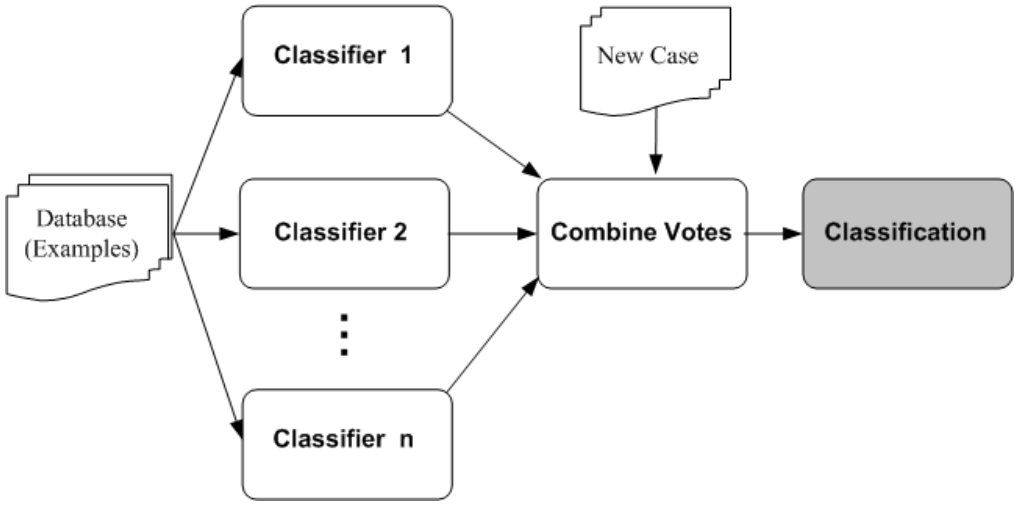

Figure 2.5: Bagging (bootstrapping aggregates)

Stage 3: Model evaluation

This stage involves evaluating the knowledge and prediction output discovered by the given data mining system. This stage is very important as it guides the choice of learning models. In classification and predictions, the performance of a model can be judged based on the percentage of cases in the test dataset that have been correctly classified. This is also known as the measurement of classifier accuracy. Estimating classifier accuracy is important as it facilitates the comparison of different classifiers and algorithms. The methodology of measuring classifier accuracy is divided into two major types: the holdout (single training/testing data) scheme and the random sub-sampling scheme.

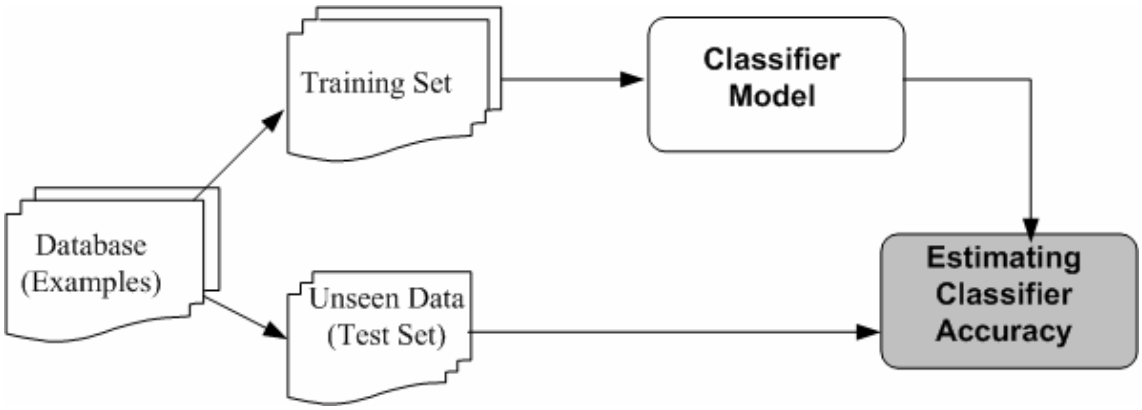

Figure 2.6: Estimating classifier accuracy with the holdout scheme.

In the holdout scheme, the given data are randomly partitioned into two independent datasets, a training and a test set. In this scheme, two thirds of the data are assigned to the training set and the remaining one third is allocated to the test set. The training dataset is used for modelling the classifier, whose accuracy is determined with the test data (see also Figure 2.6).

A similar approach is the split-sample scheme that is commonly used for early stopping in ANN models. In the split-sample method, the given data is used to form the training, validation and test partitions, however only a single subset (the validation set) is used to estimate the generalisation error of a given model.

The random sub-sampling scheme is a different form of the holdout method in which the holdout method is repeated N times. The idea is to minimise the classification bias by selecting random training/testing data in the database. In general, there are two main kinds of random sub-sampling scheme: N-fold cross-validation and leave-one-out cross-validation.

Cross-validation (Stone, 1974) can be used for estimating the generalisation ability of a given classifier (i.e. the performance on previously unseen data) or it can be used for model selection by choosing one of several classifiers that has the lowest estimated generalisation error.

In the N-fold cross-validation, the available data are separated into N parts; *N* models are then trained each on a different combination of N-1 partition and tested on the remaining partition to obtain an unbiased estimate of performance. These unbiased estimates are then averaged to yield an overall estimate of likely future performance.

The leave-one-out cross-validation scheme is an extreme type of cross-validation. It involves omitting each data point from the dataset in turn and using the remainder of the dataset to generate a model that predicts the class label for the absent data point. In the case of ANN models, for example, the remainder of the dataset can be used as a training set and the dropped data point can be used to test the resulting model. This data point then must be returned to the dataset and the next data point must be withdrawn. The process is repeated until all the data points have been used in both training and test data.

In practice the cross-validation is popular, largely because it is simple and makes good use of the available data. However, leave-one-out cross-validation is rarely applied in large datasets simply because it is computationally expensive. This strategy can be

mainly useful where the amount of available data is severely insufficient to form the usual training, validation and test partitions required for split-sample training (Goutte, 1997). Another issue with leave-one-out cross-validation is that it often performs poorly for discontinuous error functions such as the number of misclassified cases. In this case, N-fold cross-validation is usually preferred, provided N does not become too small (Shao, 1993; Breiman, 1996).

Estimating classifier accuracy is a popular scheme for evaluating different classifiers. However, it should be mentioned that this scheme is not always a reliable strategy (Andrews, Diederich and Tickle, 1995; Provost, Fawcett and Kohavi, 1998). Using classification accuracy as a measure assumes equal misclassification costs (i.e. a false positive has the same significance as a false negative) and balanced class distribution (i.e. classes are presented in a constant and relatively balanced fashion). However, these assumptions are not always valid in real-world classification tasks. For example, when the aim of a study is to enable medical staff to choose a graft for transplantation, the consequences of incorrectly predicting that a patient is not at risk of rejection are far more serious than of incorrectly predicting that a patient is at risk. In the latter case, the patient may not be given the graft. In the former case, a patient with the wrong graft may die or require subsequent grafts. As subsequent grafts generally have poorer outcomes and less chance of survival (Opelz, Gustafsson and Terasaki, 1976). Also, given the critical shortage of grafts available for transplantation, it is reasonable to assume a false positive classification error may be more serious.

The limitations of using classification accuracy (particularly when the task is to evaluate different classifiers) can be overcome by using other performance measurements, such as sensitivity, specificity and receiver operating characteristic (ROC) analysis (Kukar, 1997). These schemes provide a variety of perspectives on the performance of classifiers.

Sensitivity and specificity are two important performance measurements that are used as frequently as the classifier accuracy scheme. Sensitivity measures the performance fraction of the undesirable cases that are correctly identified (e.g. diseased patients or unsuccessful transplants). On the other hand, specificity measures the performance fraction of the desirable cases that are correctly identified (e.g. disease free patients or successful transplants).

2.4. Machine learning techniques

Statistical techniques are proven methods of solving and handling a wide range of problems and applications, although they do not always have the capacity for solving problems of high complexity (Heckerling et al., 2003; Santos-García et al., 2004).

Recently, ML tools have been drawn to the attention of computer scientists and medical professionals. Several earlier studies have compared ML classifiers with statistical techniques (Lette et al., 1994; T. K Sen et al., 1995; Tu, 1996; Pesonen, 1997; Schwartz et al., 1997). Chen (1999) compared ANN models with the statistical method of the nearest neighbour decision rule, using active sonar waveforms data. In this study neural network models outperformed the statistical approach. Freeman et al. (2000) compared LR analysis with ANN models in the prediction of in-hospital death after percutaneous transluminal coronary angioplasty. Input variables were 38 clinical signs, history, age and sex of patients admitted to an academic tertiary referral centre between July 1994 and July 1997. The results revealed that in most cases the ANN models were more accurate than LR models. The results also suggested that further investigations are needed to better understand the impact of neural network models on clinical outcomes analysis.

ML systems are developed for many domains including medicine. Examples include: *Medical Diagnosis* (Ledley and Lusted, 1959; Scott, et al., 1984; Barnett et al., 1987; Scott, 1993), *Medical Imaging* (Rotolo LS, et al., 1963; Miller, Blott and Hames, 1992) and *Drug Development* (King, et al., 1992; Weinstein, Kohn and Grever, 1992).

ML research represents an important and promising direction as ML-based techniques such as Case-Based Reasoning (CBR)[1], ANN and decision trees have been used successfully for intelligent information retrieval methods and prediction tasks (Safavin and Landgrebe, 1991; King et al., 1992; Rudolfer et al., 1999; Berrar et al., 2003; Shadabi and Khodai-Joopari, 2003; Li et al., 2004; Shadabi and Khodai-Joopari, 2004). Machine Learning Techniques (MLT) usually learn a task from a series of examples (Michie, Spiegelhalter and Taylor, 1994; Mitchell, 1997). An overview of the ML learning strategy is demonstrated in Figure 2.7.

[1] Case-based reasoning (CBR) is a problem-solving strategy that finds solutions to new problems by analysing previously solved problems or historical data, called cases.

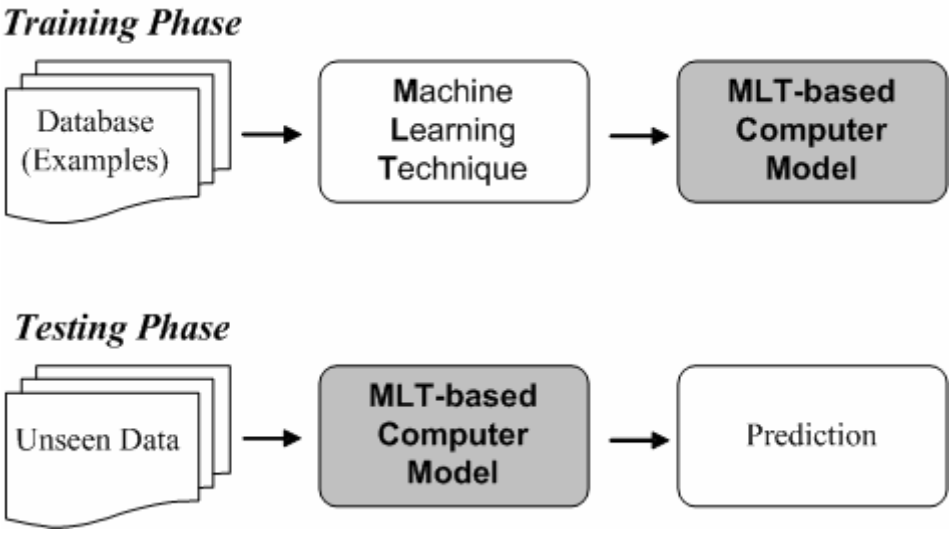

Figure 2.7: Overview of Machine Learning Strategy.

MLT can be applied in healthcare environments where an automated process must adapt to changing conditions. MLT provide a great opportunity for researchers to enhance the information processing and retrieval capabilities of current knowledge-based systems.

This section describes three ML strategies, namely ANN (a sub-symbolic approach), decision trees (a symbolic approach) and ensemble classifiers, as a prelude and justification for the research presented in the later chapters.

2.4.1. Artificial neural networks (ANN) for clinical data

Neural networks models (see Section 3.2.2) are examples of *sub-symbolic* methods. Unlike *symbolic* methods, sub-symbolic methods do not have the ability to induce symbolic representations from data and generate a set of rules that can be understood by humans. However, in positive side, they are known to provide very good solutions to many difficult medical problems (Baxt, 1991; Ashutosh et al., 1992; Tu and Guerriere, 1993; Baxt, 1995; Mobley et al., 1995; Ortiz et al., 1995; Itchhaporia et al., 1996; LaPuerta et al., 1998; Pantazopoulos et al., 1998; Dayhoff and DeLeo, 2001). A study conducted by Lisboa (2002) provides a good review of recent real world applications of neural network models in healthcare domain.

Heckerling et al. (Heckerling et al., 2003) employed a feed-forward back-propagation network when they predicted the present or absent of pneumonia among adults presenting with acute respiratory illness. In some cases, their network was able to

discriminate between patients with or without pneumonia with high level of accuracy of 94% to 95%. They found that in many situations neural networks outperformed the LR models.

Michael et al. (2003), compared ANN with traditional logistical regression models for prediction of the occurrence of delayed graft function in cadaveric renal transplants. Their results revealed that logistic regression was 36.5% sensitive and 90.7% specific compared to the ANN which was 63.5% sensitive and 64.8% specific. Other groups (Matis, 1995) used a MLP network on 290 liver transplantation patients to predict graft outcomes. They used 240 records for training purpose and 55 records for testing purposes. Their system was able to accurately predict 88% of graft failures and 98% of graft survival cases. Overall, they concluded that the ANN model was able to provide a better level of sensitivity than LR methods in these cases.

Santos-Garcia et al. appear to be more imaginative in their approach to use MLP networks for the purpose of predicting cardio-respiratory morbidity after lung resection (Santos-Garcia, et al., 2004). They employed a variety of training functions such as the Conjugate Gradient Descent and the Levenberg-Marquardt. They concluded that neural networks models are able to provide more sensitive and accurate results than LR.

More recently, Ramesh et al. (2004) conducted a review of different AI methods (including fuzzy expert systems, evolutionary computation, ANN and hybrid intelligent systems) and their application in the clinical settings. Their study showed that the ANN methods are among the most commonly studied form of AI techniques in medicine and that AI techniques such as ANN are a promising technique with the potential to be applied in almost any real world clinical decision support system. A similar conclusion were reached by (Piccolo, et al., 2002) when the diagnoses of an ANN-trained computer, a dermatologist experienced in dermoscopy (five years of experience) and a clinician with minimal training in this field were compared. The comparison results showed that analysis either by a trained dermatologist or an ANN-trained computer had higher diagnostic accuracy compared to that of an inexperienced clinician. It also demonstrated that AI-based computer diagnosis systems can have the potential to play an important role in future medical decision support, particularly at centres where clinicians are not experienced in a specific field.

2.4.2. Decision trees for clinical data

Research shows that decision tree (see Section 3.2.3) programs are reasonably powerful prediction techniques and not very difficult to construct (Shavlik, Mooney and Towell, 1991; Murthy, 1998; Gaudart, et al., 2005; Gericke, et al., 2005). They also have the ability to explain their reasoning processes. However, to construct useful decision trees from real-world data, a fair amount of understanding of the problem under investigation and the methodology behind this technique is required. Furthermore, where a decision tree model is presented with small, incomplete and noisy training data, it may not generate reasonable predictions. The constructed decision tree may fit the training dataset very well; in fact a large tree may separate all data into classes (i.e. a tree model with zero misclassification rates). However, such model could seriously overfit the data by leaving very small amount of data samples in the leaf nodes (in the worst scenario leaving only a single data sample). Of course such model will not generalise well on new and unseen data. A number of techniques exist for solving overfitting issue, a well known example is pruning (Quinlan, 1986; Quinlan, 1987a; Mingers, 1989; Breslow and Aha, 1997). This process removes thin branches (branches that assess only a small number of training data samples). Breiman, et al. (1984) suggest a cost complexity approach by first growing an excessively large tree and then finding a nested sequence of sub-trees by continually pruning branches of the tree. Their pruning procedure includes misclassification rates. Additionally, pruning includes a cost complexity criterion that reflects the cost associated with the complexity of the tree, based on the number of terminal nodes in a sub-tree or branch. Apart from the overfitting issue, smaller (less complex) trees are more easily used by decision-makers to study the underlying relationships in the data.

2.5. Recent research directions: ML-based hybrid systems

In recent years there has been a rapid growth in the successful use of hybrid intelligent systems in many diverse areas such as science, medicine and commerce. The main contributing factor influencing the development of hybrid systems in medicine has been the demand for a more powerful yet flexible and robust technique. This is in order to

cope with the extensive amount of data and knowledge stored in clinical databases, and more importantly, the complexity of interpretation.

ML-based techniques such as ANN, Decision trees and recently support vector machines (SVMs)[2] are able to accept numerous input variables and adapt their criteria to better match the data they analyse (Ribeiro, 2003; Ribeiro, 2005). However, unlike decision tree methods, both ANN and SVM technique are known as black box models, which lack the explanation capability on how to reach a decision.

An expert system in the healthcare sector ideally needs to be able to deal with ever increasing amounts of data, especially in the complex fields of medicine and biology. In most expert systems, information and knowledge are represented as a set of rules. Theses rules are usually easily understood by users; however generating and maintaining a large rule-based, knowledge-acquisition tool can be labour intensive. In fact, as mentioned previously, issues relating to the so-called, knowledge-acquisition, task have been raised over the years by many researchers in knowledge-based systems (Saito and Nakano, 1988; Sestito and Dillon, 1991). Furthermore, the brittleness that can happen when a system is presented with noisy data, unusual values or incomplete data with missing values (Holland, 1986). This can usually occur when a system tries to perform the data mining task by using only symbolic data mining algorithms such as decision tree models (Breiman et al., 1984; Quinlan, 1993; Quinlan, 1996).

Research shows that with the growing power of neural networks tools, neural networks are a promising method in the prediction of medical outcomes. The idea of using more powerful methods such as ANN for improving the learning performance of knowledge acquisition tools has been around for many years (Gallant, 1988b; Medsker, 1994).

Over time, neural networks have proven to be very powerful tools for pattern recognition and classification tasks. The power of ANN in comparison to other symbolic ML techniques such as decision trees has been well documented by numerous researchers (Shavlik et al., 1991; Thrun, et al., 1991; Diederich, Hild and Bakiri, 1995). Their study revealed that for most problem domains ANN models are considered to be a very good choice. ANN models are able to accept numerous input variables, adapt their

[2] Support vector machines are an inductive machine learning technique based on the structural risk minimization principle, which can be used for classification of both linear and nonlinear data and aims at minimizing the true error.

criteria to better match the data they analyse. Given enough training times and appropriate numbers of hidden units and layers, ANN models are able to produce a high level predictive accuracy rate, solve complex problems and learn interesting linear or nonlinear relationships. These interesting features of ANN models provide a powerful and compact knowledge representation tool, with an efficient storage and individual patterns recall system. The learned patterns and relationships are stored as a set of values across all weights and thresholds. Theses values are usually incomprehensible for human users.

Over the past few years, information researchers in data mining and knowledge-based systems have turned to a multi-strategy learning approach (Katz, et al., 1994; Koutroumbas, et al., 2001). Theses approaches facilitate data mining by combining the power of connectionist learning systems such as ANN with rule-based learning methodologies. There are several reasons for the combination of neural network and symbolic techniques within a hybrid based system, these include:

i) These techniques can translate a neural network system into an alternative, more understandable model.

ii) In critical problem domains, such as medical diagnosis, it is not acceptable for a computer-aided medical diagnosis or a decision support system to be just accurate, systems are required to provide transparent justifications for any given decision or output generated by the system. These justifications are crucial to user acceptance of intelligent systems.

In some situation the users may need to determine any possible interactions between the attributes and discover new concepts from data stored in databases, rather than simply obtain a high level of classification or prediction accuracy. Over time, a wide variety of methods have been proposed to extract knowledge (or explanations) from complex classifiers such as ANN using concepts drawn from *fuzzy logic* (Bellman and Zadeh, 1970; Zadeh, 1983; Masuoka, et al., 1990; Mitra, 1994; Castro, Mantas and Benitez, 2003) and *rule based methods* (Fu, 1991; Thrun et al., 1991; Craven and Shavlik, 1996a; Krishnan, 1997; Setiono, 1997; Zhou, Chen and Chen, 2000).

Next section provides an overview of a variety of hybrid intelligent techniques that might be utilised for extracting rules from ANN, particularly for clinical knowledge discovery and decision making processes.

2.5.1. Extracting rules from neural networks

An important reason for the extraction of explanations and rules from neural networks in expert systems is to directly add the extracted knowledge to the knowledge base system (Saito and Nakano, 1988; Gallant and Hayashi, 1991; Sestito and Dillon, 1991; Sestito and Dillon, 1994). Gallants Connectionist Expert Systems (Gallant, 1988a) and Matrix Controlled Inference Engine (Gallant and Hayashi, 1991) are examples of early models where expert system rules are extracted from a neural network. The basic idea is to combine the explanation facility of rule-based systems with the ANN in order to generate the knowledge (d'Avila Garcez, Broda and Gabbay, 2001). Coupling ANN and rule extraction algorithms can significantly improve the learning performance of knowledge acquisition tools, enhance the overall utility of ANN and reduce the brittleness of rule-based systems. Rule extraction algorithms can be categorized into four categories, namely the *decompositional*, *pedagogical*, *eclectic* and *compositional* algorithms (Andrews et al., 1995; Tickle, et al., 1998). These classification schemes are based on the approach used to study and analyse the underlying ANN architecture or/and the classification given by the network for the processed input vectors.

Decompositional (local) methods start by extracting rules from each +unit (hidden and output) in a trained neural network. The rules extracted at the individual unit level are then combined to form a global relationship and the final rule base for the ANN architecture as a whole. Some examples of this style of algorithm are KT (Fu, 1994), Subset (Towell and Shavlik, 1993), COMBO (Krishnan, 1997) and RX (Setiono, 1997), RULEX (Andrews and Geva, 1994; Andrews and Geva, 1995).

KT (Fu, 1994) is the earliest implementation of this style of algorithm. This approach focuses on mapping individual (hidden and output) units into conventional Boolean rules. The following example of rules generated from KT is taken from (Fu, 1994):

If $(0 \leq Output \leq Threshold_1) \Rightarrow$ *no, and*

If $(Threshold_2 \leq Output \leq 1) \Rightarrow$ *yes, where* $(Threshold_1 < Threshold_2)$

Similar to the KT algorithm is the Subset algorithm by Towel and Shavlik (1993). The Subset algorithm searches for subsets of incoming weights that exceed the bias of a particular unit. In many situations, this style is capable of delivering a set of rules that yield exact representation of the underling ANN architecture, however as reported by

Towel and Shavlik (1993), the computational time for algorithms such as Subset and KT may grow exponentially with the number of inputs. Consequently, this can impose great restriction on network architecture and training procedures.

Towel and Shavlik (1993) tried to reduce the complexity of rule searches by introducing a second rule extraction approach, namely M of N, which is an alternative to the if-then form of rules. The stages of the M of N approach are outlined below:

1. Generate an ANN and train it using back propagation. With each hidden and output unit, form groups of similarly weighted links.
2. Set the link weights of all group members to the average of the group.
3. Eliminate any groups that do not significantly affect whether the unit will be active or inactive.
4. Holding all link weights constant, optimize the bias of all hidden and output units using the back propagation algorithm.
5. Form a single rule for each hidden and output unit. The rule consists of a threshold given by the biases and weighted antecedents specified by the remaining links.
6. Where possible, simplify rules to eliminate superfluous weights and thresholds.

The Boolean expression or rules that are generated by this algorithm states:

> *If (M of N antecedents/conditions, a_1, a_2,....,a_n, are true) Then (the conclusion b is true).*

This approach can be useful for problems that can be expressed in the form of the M of N rules. The most impressive feature of the M of N algorithm is its ability to analyse clusters of similarly weighted links. Most of the previously reported algorithms conduct searches by exploring and testing a space of conjunctive rules against the network in order to find out whether they are valid rules or not. However, the M of N algorithm starts by analysing each cluster of similarly weighted links as whole and eliminating groups that have little significant effect on the result. In the final stage it creates rules from the remaining clusters through an exhaustive search. Approaching the rule extraction problem in this way can reduce the search space.

Similarly in the late 1990s, Taha and Ghosh (1996) proposed three new techniques with different power of rule extraction, namely BIO-RE (Binarized Input-Output Rule Extraction), Partial-RE and Full-RE for extracting rules from trained feed-forward

ANN. BIO-RE is a binary rule extraction technique that generates a truth table from the binary inputs and simplifying the resultant Boolean function. However in practice, creating a truth table that represents all possible valid input-output mapping of the trained network based on this approach can only work for small size problems. Another limitation to this approach is the requirement for Binarized Input-Output variables. If the original inputs are not binary, they have to be converted to binary values with great amount of care. Partial-RE and Full-RE use a similar link rule extraction approach to the Subset algorithm with some improvements and simplifications. As the name suggest, theses two rule extraction techniques can be used to extract partial or full rule sets from trained feed-forward. The Partial-RE algorithm is well suited for larger networks where time complexity can be a real issue. These algorithms provide users with some degree of control over the extracted rule set and information they obtain from the learner.

The second category of rule extraction algorithms is pedagogical. The core idea in the pedagogical (global) approach is to treat the trained neural network as a black box. It aims to extract rules that map inputs directly into outputs (Tickle et al., 1998). The pedagogical algorithm uses a trained neural network to only generate test data for the rule generation algorithm. In this strategy the target concept is computed by the network and the input vectors are the actual network's input vectors (Craven and Shavlik, 1994).

At the end of 1980s, Saito and Nakano (1988) proposed a medical diagnosis expert system based on a multi-layer neural network. They implemented a pedagogical algorithm and used it to observe the changes in the levels of input and output units. In order to explain the inference process in the system, they presented a rule extraction routine that characterises the output classes directly from the inputs of simple networks. They were particularly interested in implementing a routine for extracting compact and propositional rules from a simple network. Interestingly, their research revealed that even for a relatively simple problem domain, the number of extracted rules can be large. VIA (Thrun, 1994), TREPAN (Craven, 1996; Craven and Shavlik, 1996a), STARE (Zhou et al., 2000) are other examples of this style of algorithm.

TREPAN (Craven, 1996; Craven and Shavlik, 1996a) is one of the most significant developments in the pedagogical approach. The TREPAN algorithm is a relatively new data mining approach that operates on neural networks and uses the M of N rule extraction strategy. Figure 2.8 shows the TREPAN algorithm described in

(Craven and Shavlik, 1999) for converting a Neural Network to an M of N type Decision Tree.

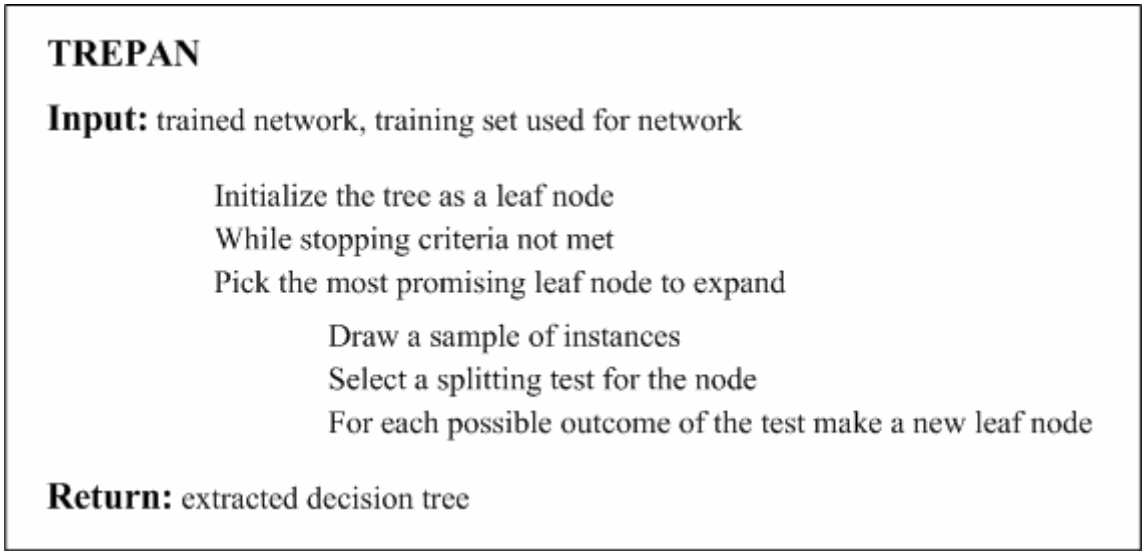

Figure 2.8: The TREPAN algorithm

The TREPAN algorithm is similar to standard decision tree algorithms such as See5 (Quinlan, 2000), and C4.5 (Quinlan, 1993), which learn directly from a training set. Figure 2.9 that has been taken from (Craven, 1996), shows the tree extracted by TREPAN from the DNA coding network. This tree represents the function that the network has learnt. In this scenario, the system generated a decision tree by recursively partitioning the input space in order to recognise protein-coding regions in *E. coli* DNA sequences. The features in the coding domain represent the 64 codons (three-letter words) that can be formed from the DNA bases: fa, c, g, tg. Each Boolean feature indicates the presence or absence of a given codon in-frame in the sequence being classified. Negated features are preceded by a minus sign. The class labels indicate whether a given, fixed-length sequence is predicted to encode a protein or not.

TREPAN does not require any specialised neural network architecture or training algorithm since, like other pedagogical algorithms, it operates on datasets and not the actual underlying architecture. In fact, Craven and Shavlik (1999) believe that TREPAN is so general and flexible it (theoretically) can be applied to wide variety of networks, including ensembles (Craven, 1996).

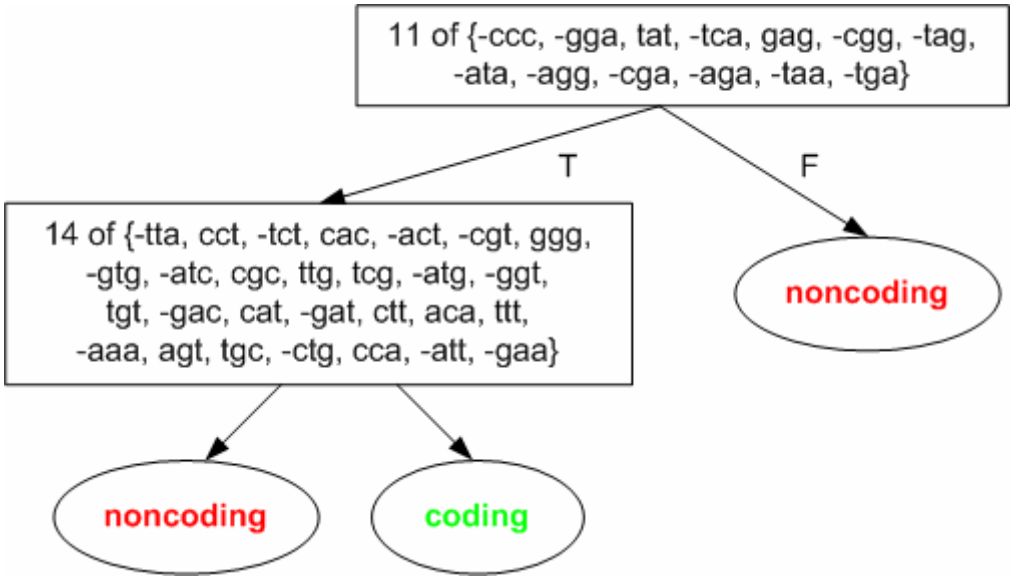

Figure 2.9: A representative tree for the DNA coding domain (Craven, 1996).

Johansson and Niklasson (2001) performed a comparative study between the accuracy of ANN approaches and traditional linear approaches. Not surprisingly, their results clearly revealed the superiority of ANN. Furthermore, in a separate study, Johansson and Niklasson (2002) exploited the potential of neural networks in decision-making problem domains where comprehensibility was also considered to be an important factor for decision makers. In this study they successfully used the TREPAN algorithm (Craven, 1996; Craven and Shavlik, 1996b) to create decision trees from trained ANN models. Their results revealed that the decision tree extracted by the TREPAN algorithm had a higher accuracy rate on unseen data than the trees created by the standard tool See5 (Quinlan, 2000). However, they noted that the trees created by TREPAN were still as complicated as the trees generated by See5 (Quinlan, 2000). This shortcoming was also noted by Boz (2002). This shortcoming was the motivation for another study by Johansson and Niklasson (2003) where they proposed a novel method called G- REX based on Genetic Programming (GP)[3] for rule extraction. The genetic programming rule extraction algorithms attempt to incorporate ideas of natural evolution for classifications as well as optimisation tasks (for detailed discussion on genetic algorithms see Goldberg (1989) and Mitchell (1996)). In their primary study

[3] Genetic programming (GP) is an automated method for creating computer programs from a high-level problem statement of a problem. Genetic programming starts from a high-level statement of the requirements and automatically creates a computer program to solve the problem.

they used a relatively small advertising dataset. The results showed that the rules extracted by G-REX generally outperformed the TREPAN and See5 in terms of both accuracy and comprehensibility. However, the G-REX algorithms can be computationally expensive, even for relatively small datasets.

1. IF behavioural = Condition3, on, AND region = Condition 1, off, AND Age = Condition3, on,
   THEN =outcome1
2. IF mn_problems = Condition3, off, AND reporter = Condition1, on, AND Marital status = Condition2, on,
   THEN =outcome1
3. IF longterm_stressors = Condition2, on,
   THEN =outcome2
4. IF time = Condition2, on, AND risk = Condition8, on,
   THEN =outcome2
5. IF Age = Condition3, on,
   THEN =outcome2
6. IF reporter = Condition2, off, AND dependent = Condition2, off, AND time = Condition4, off,
   THEN =outcome4
7. IF occupation = Condition2, off,
   THEN =outcome1
8. IF time = Condition4, off, AND last_seen = Condition2, off, AND day = Condition4, off,
   THEN =outcome1
9. IF day = Condition1, off,
   THEN =outcome2
10. IF mn_problems = Condition6, on, AND longterm_stressors = Condition4, off, AND day = Condition3, on,
    THEN =outcome2
11. IF risk = Condition4, off, AND character = Condition1, off,
    THEN =outcome1
12. IF Age = Condition2, on,
    THEN =outcome2
13. IF season = Condition3, off, AND day = Condition1, on,
    THEN =outcome2
14. IF occupation = Condition6, on, AND longterm_stressors = Condition2, off, AND reporter = Condition1, off,
    THEN =outcome3
15. IF mn_problems = Condition2, on, AND longterm_stressors = Condition1, off,
    THEN =outcome3
16. IF occupation = Condition2, off, AND occupation = Condition7, on,
    THEN =outcome2
17. IF urban = Condition2, off, AND history = Condition2, on, AND season = Condition1, off,
    THEN =outcome2
18. IF appearance = Condition1, off, AND ph_outcome2 =Condition2, on, AND reporter = Condition4, on,
    THEN =outcome3

Figure 2.10: A rule set example generated by GP algorithm (Blackmore and Bossomaier, 2003).

Further supports for improving the comprehensibility of rules from overly complex models, come from Quinlan (1987b), Domingos (1998), Keedwell, Narayanan and Savic (2000) and Blackmore and Bossomaier (2003). Blackmore and Bossomaier (2003) used genetic algorithms to evolve rules directly from trained neural networks. Figure 2.10 taken from Blackmore and Bossomaier (2003), shows a rule set example generated by a GP algorithm. Their experimental results revealed that the rules extracted by genetic algorithm approach outperform various decision tree based algorithm such as WEKA *J48.PART* (Witten and Frank, 2000) in terms of both accuracy and comprehensibility. However, due to the nature of GP algorithms, the

computational time grows considerably for the large neural networks. A similar shortcoming for the TREPAN algorithm was reported by Golea (1996).

The third rule extraction category is eclectic algorithms. The methods from this category, like the decompositional category, carefully examine the ANN at the level of individual units; they also extract rules at the global relationship within trained neural networks. DEDEC (Tickle, Orlowski and Diederich, 1996) is an example of this style of algorithm.

The final category is compositional algorithm. The compositional algorithms neither focus on local models that mirror the behaviour of individual units, nor treat the network as a black box, like pedagogical approaches. Representatives of this category include algorithms proposed by Omlin et al. (1992) and Giles et al. (1992). This category has been designed for recurrent ANN model where the relationships between different outputs are used to find models of the underlying recurrent neural networks. Theses are typically in the form of finite-state machines that mimic the network to a satisfactory degree.

2.5.2. Extracting rules from a neural network ensemble

Neural network ensembles (see Section 3.2.4) are known to be good predictive models. However, as a neural network ensemble is composed of several independently trained neural network models its comprehensibility is considerably more difficult than its component classifiers.

Over the years, many authors have tackled the problem of rule extraction from individual networks. However, much less work has been done in the explanation of several combined neural networks (Wall and Cunningham, 2000). This is of significance for the acceptance of this technique in medicine.

One example of research on improving the comprehensibility of artificial neural networks ensembles can be found in Craven's recent work (1996). This work uses the TREPAN algorithm (Craven, 1996; Craven and Shavlik, 1996a) and the Addemup algorithm (Opitz and Shavlik, 1996) for generating rules from the ensembles in a telephone domain. TREPAN is a good choice for this task because it does not try to translate the entire individual units (hidden and output) of networks into rules and, more importantly, it can be applied to a wide class of networks. In this study Craven (1996) induces an ensemble of neural networks by using the Addemup algorithm. The

Addemup algorithm uses a genetic search strategy to generate an ensemble of neural networks (Towell and Shavlik, 1994). The solutions provided by Addemup algorithm had high level of accuracy rates, however but these solutions required very large and complex networks. In fact, for this relatively simple problem domain they had to apply the TREPAN algorithm to ensembles of networks that, on average, had more than 10,000 parameters per ensemble. These kinds of networks take much longer to train and the rules are generated at greater computational cost. This can be problematic, particularly since the time complexity of the TREPAN algorithm is known to be problematic in some complex cases (Golea, 1996; Quinlan, 2000; Boz, 2002).

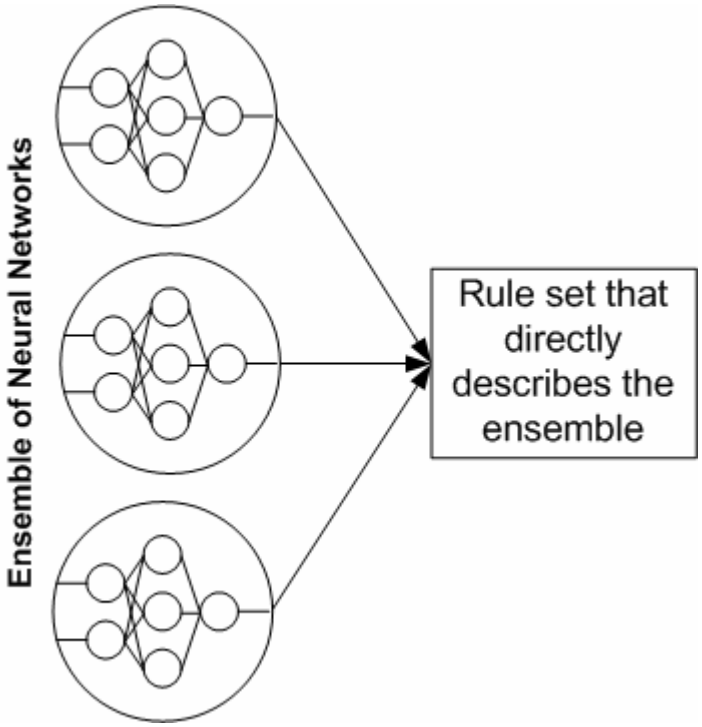

Figure 2.11: A black box (global) explanation approach

Domingos (1998) describes a similar black box (global) decision tree-based rule extraction technique for ensemble models, namely Combined Multiple Models (CMM). In CMM the basic idea is to preserve most of the accuracy gains of multiple model approaches while still producing a comprehensible set of rules (see Figure 2.11 for the general structure of a black box approach). They attempted this by feeding a base model (e.g. a bagged ensemble of models) a new training set that consisted of the original examples plus a large number of artificial examples that had been generated and labelled according to the ensemble. Their evaluation results (on 26 benchmark datasets)

revealed that in some situations it may be impossible to improve accuracy without losing comprehensibility or increasing the complexity. They suggested further research is still needed to determine the conditions in which the use of multiple models such as CMM is beneficial.

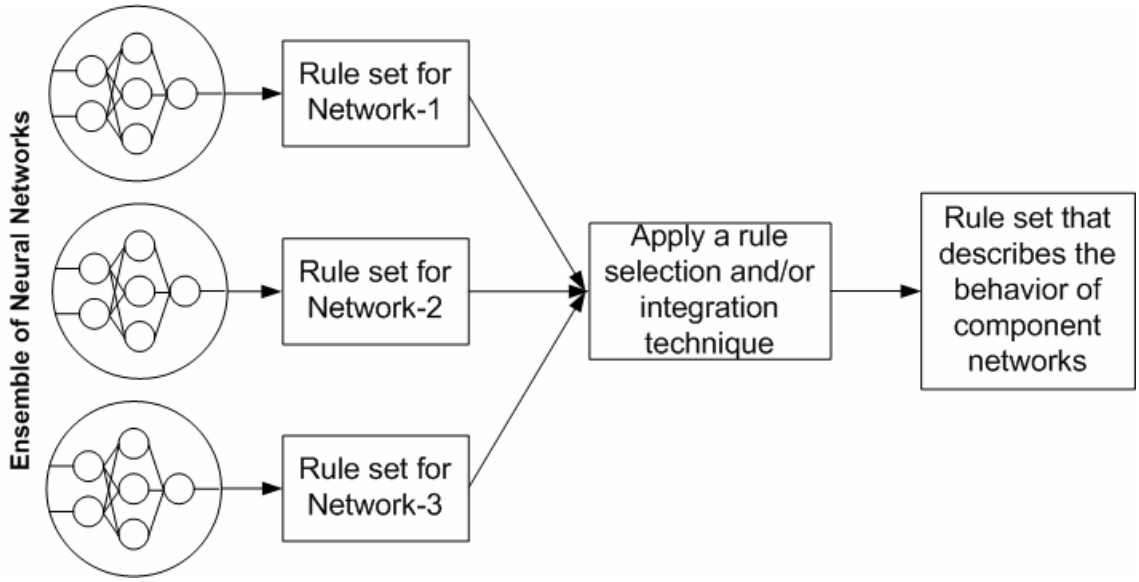

Figure 2.12: A component-based (local) explanation approach

The results of Domingos's study were not surprising for Wall and Cunningham (2000). They argue that using a global explanation strategy can be problematic. They believe that since the multiple ANN models are intentionally trained to increase diversity, this diversity should be considered in the generated rule sets. This point has also been made by Zenobi and Cunningham (2001) where they regard the ensemble members as specialists in sub-regions of the problem space.

Recently, Wall et al. (2003) proposed a local explanation strategy as an alternative to black box or global approaches such as TREPAN and CMM. In this study they tried to explain the ensemble on a case-by-case basis (see Figure 2.12 for general structure of a local approach). This strategy has been pursued successfully by other researchers such as Sima(1995) and Das, et al.(1998). Local explanation strategy can be used to identify the ensemble members that are relevant in explaining the prediction associated with a particular case. In the first step, Wall et al. (2003) trained an ensemble of networks using bootstrapped sets generated from a training set. Then, they used the networks to produce a class label for a new set of data generated from the original training set. This labelled data was fed into Quinlan's C4.5 package (Quinlan, 1993) in order to build

decision trees and extract rules that capture the behaviour of each individual network. Once the ensembles of rule sets had been produced, they applied a rule selection strategy (mainly based on majority voting) and calculated a number of coverage statistics in order to study the fitness of the rules to any unseen example and provide a case-by-case explanation. It should be noted that this approach attempts to discover how the elements of the ensemble contributed to the prediction. This is done by first extracting sets of rules from each of the member of networks in the ensemble and producing ensembles of rule sets. It is important to note that Wall et al. (2003) tested this strategy on relatively small ensembles of networks. Although there is no reason to believe that this approach will not perform well on larger ensembles, its suitability or computation time complexity remains to be verified.

More recently, a series of black box approaches namely REFNE (Rule Extraction from Neural Network Ensemble), C4.5 Rule-PANE (C4.5 Rule Proceeded by Artificial Neural Ensemble), and NeC45.5 (Neural ensemble based on C4.5) have been suggested by Zhou and Jiang (2003) for extracting meaningful knowledge from neural network ensembles. These approaches utilise the trained neural network classifiers to generate instances and then extract rules from them. As described by Zhou and Jiang (2003), algorithms such as C4.5 Rule-PANE were not designed with the improvement of comprehensibility of the Neural Network Ensembles in mind. These style of algorithms use an ensemble of ANN as a data pre-processing strategy for the standard rule induction systems such as C4.5 Rule (Quinlan, 1993; Zhou and Jiang, 2003). Their results reveal that given enough training data, the C4.5 Rule-PANE approach, can provide comprehensible solutions with strong generalisation ability that are significantly more accurate than the standard decision trees.

2.6. Summary

Research shows that the state of the art in computer-based clinical knowledge discovery and decision support research has matured. Over the years, computer-based clinical decision support systems have been used successfully in the clinical setting (McDonald, 1976a; McDonald, 1976b; McDonald, Wilson and McCabe, 1980; Chase, et al., 1983; McDonald, et al., 1984; Rogers, Haring and Goetz, 1984; Brownbridge, et al., 1986; McDowell, Newell and Rosser, 1986; McDowell, Newell and Rosser, 1989; Petrucci, et

al., 1991; Rosser, et al., 1992; Fuchs, et al., 1999; Bates, et al., 2001). Expert systems are now the common form of clinical decision support systems in routine clinical use. Many early expert systems that provided good user interface design, reasonably accurate results with clear justification and reasoning are still in routine use in health care systems, clinical laboratories and educational institutions.

Recently, researchers in data mining and knowledge-based systems have turned to ML-based techniques such as neural networks and decision tree algorithms. These techniques can be either used as stand alone systems or deeply integrated into many existing expert systems. This chapter presented selected ML techniques that can be used in clinical knowledge discovery and decision support systems.

More recently, there has been extensive research on hybrid and integrated models. However, literature search revealed that the majority of models used for the prediction of medical events have mainly focused on the mathematical and technical aspects of prediction algorithms and there is little evidence about new algorithms for overcoming the current clinical data issues. The rapid development of data collection and storage technologies has led to the formation of large number of clinical databases and there is a need for identifying suitable observations in large and noisy clinical datasets. Large clinical databases might contain much useful knowledge; however, computational efficiency, especially for powerful but processing-intensive methods, such as ANN can be a major concern. Also, it is well known that clinical data are complex due to genetic and biological diversity of individuals, disease marker variability amongst individuals, the variability in the combination of drugs given to each individual and missing data, to name a few. Therefore, it is desirable to have a method at hand to fucus mainly on data issues and extract data points that contain higher information content from a sea of noisy and incomplete clinical data and build better classifiers.

Furthermore, combinations of techniques or specifically adapted ML and training methods have great potential to be effectively used in predictive modelling tasks. Multiple techniques can be potentially integrated to improve the transparency of results, ease of use and the accuracy of medical prediction tools. Of these techniques some are being actively developed (notably hybrid neural network-decision trees), where the goal is to extract useful explanations from individual neural network classifiers. As yet, little evidence exists about research and development in the explanation of several combined neural networks. In this vein, Wall and Cunningham (2000) noted that: “Without a

method of extracting reliable rules from an ensemble, we are once again faced with the black box problem that once plagued neural networks."

A literature search also revealed little evidence of the application of neural network models for the prediction of graft outcomes following kidney transplantation, in particular for matching donor-recipient pairs with a high accuracy rate. Successful kidney transplantation will not only extend the length and quality of life for the recipient but also reduce medical expenses and increase access (for patients in kidney transplant waiting list) to donor kidneys by reducing the need for multiple kidney transplants.

There are still many challenges to be overcome. Each technique has its own advantages and disadvantages under different circumstances. Although, it is clear that integrating AI modules into computerised patient records and clinical findings can provide a great chance for improving the quality of care.

The work of this thesis extends current data mining research further through a new model called RIDC-ANNE. The RIDC-ANNE model is based on neural network ensemble technology. This model treats the ensembles as a black box or data pre-processing tool in order to provide a body of new examples to feed standard rule induction or decision tree systems such as C4.5 (Quinlan, 1996). It is important to note that the RIDC-ANNE technique incorporates elements of both local and global explanation strategies. Like other black box or global approaches (Zhou and Jiang, 2003; Zhou and Jiang, 2004), in this strategy the target concept is computed by the ensemble of networks and the input vectors are the actual network's input vectors. This technique considers the diversity and expertise of the component networks in the rule generation process. However, it is not purely a local strategy (Wall et al., 2003) because it does not focus on identifying the ensemble members that are relevant in explaining the prediction (output) associated with a particular case. Instead it explains the output of the ensemble based on a cluster of cases that consistently generate agreement across the classifiers with similar expertise. Rule-extraction provides an explanation capability to the black box models, hence a check on their core logic. This capability is important for the acceptance of the black box machine-learning techniques in medical community.

The RIDC-ANNE approach has the ability to extract the input patterns that were included across the neural networks series in the final results. It can also be used to provide clarity for the general behaviour of an ensemble. Overall, the goal of this model is to identify the regions in the data space that have high impact on system performance

and narrow the increasing gap between research direction in algorithm development and explanation facility enhancement, with a view towards the development of more efficient and transparent intelligent systems.

Chapter 3

Proposed hybrid ANN Approach: RIDC-ANNE

When told, "I am too busy treating patients to do research." I answer: "When you treat a patient, you have treated a patient. When you do research, you have treated ten thousand patients. "

Robert H. Riffenburgh (2006)

3.1. Introduction

Over the past few years there has been a rapid growth in the successful use of advanced data mining techniques with specially adapted ML and training schemes across the health spectrum. It has been proven that by introducing such powerful tools into medical analysis, new treatments and better quality of care become available for patients. These newer techniques can be used to identify diseases or model certain medical outcomes.

The current chapter discusses a novel hybrid technique called Rules and Information Driven by Consistency in Artificial Neural Networks Ensemble (RIDC-ANNE). The RIDC-ANNE approach has been designed and developed during this period of study based on combination of initial data preparations, preliminary classification by ensembles of ANNs, and generation of new training data based on criteria of highly accuracy and model agreement. The model will also generate the decision tree models to provide classification of data and the prediction of results. Furthermore, the chapter outlines the key components of the research design and reveals how the experimental design will be used to analyse and evaluate the results of this study.

3.2. Materials and methods:

3.2.1. Clinical Data

The original kidney transplants database used in this study is a series of text files, with each file containing the data of a table. There are 35 tables in the database. A data dictionary is available to be used for interpreting the data contained in the tables[4]. The parameters used for the training and testing tasks are extracted from relevant tables. Some variables from these tables were removed because they were actually an indication of the outcomes of the transplant, and as such they were measured after the transplant had been made. The challenge for this study is to select an appropriate kidney from the available pool of organs for a particular patient.

The Pima Indians Diabetes dataset and The Wisconsin Breast Cancer dataset are standard data sets available at UCI machine learning repository and had been studied extensively by the machine learning community. The Pima Indians Diabetes data set is based on the personal data of the Pima Indians, which originally was obtained from the US National Institute of Diabetes, Digestive and Kidney Diseases. All patients in this dataset are females of Pima Indian heritage and at least 21 years old. The database contains 768 data samples taken from patients who might have shown signs of diabetes. All the data samples in this dataset have no missing attributes. There are 8 attributes (inputs) in this database and two output classes, diabetes and non-diabetes. The Wisconsin Breast Cancer dataset is a relatively clean and non-complex dataset. The dataset has nine attributes (inputs) and two output classes. All nine inputs are continuous and range from 1 to 10. The database contains 699 samples, with 683 samples of complete data and 16 samples with missing attributes. Each of the 683 available instances is labelled as either Benign (444 instances or 65% of data) or Malignant (35%). The task is to predict benign or malignant classes.

[4] See ANZDATA (2000), "Data Dictionary: ANZDATA Registry Database." URL: www.anzdata.org.au/.

3.2.2. Artificial neural networks (ANN)

The inspiration for ANN models came from a desire to simulate the features of biological neural networks and learning systems, which show high power in pattern recognition tasks and adaptability.

Neural networks architecture generally divides into two categories namely, feed-forward networks (networks without any loops in their path) and feedback networks (networks with recursive loops). Feed-forward neural networks are the most widely used form of network architecture. They are built of a set of interconnected processing neurons (nodes) that store the knowledge and operate in parallel. Networks can be structured with different number of nodes (also known as processing elements), layers (built by nodes), connections that bond the nodes together, and connection weights which are similar to regression model coefficients (Guerriere and Detsky, 1991; Simpson, 1992; Zurada, 1992; Cross, Harrison and Kennedy, 1995; Ripley, 1996). A simple neural networks model has many input-nodes and one output. The output node provides the network's response. The number of input and output units must be set based on the number of input and output (target) attributes respectively. A model of a neuron with three inputs is demonstrated in Figure 3.1. In this model the set of input signals are $I_i\,(i=1,2,3)$. A set of input signals is supplied either from the outside or from a previous layer. The inputs are multiplied by the weight $W_i\,(i=1,2,3)$. The weighted inputs and bias (denoted by W_0) are summed by neuron and produce the net input (denoted by S_1). W_0 is an offset that can be used to scale the result. The next step is to use the net input as the argument of the transfer function f. The actual output depends on the particular transfer function (also known as an activation function) that is specified for the units (Zurada, 1992). The output (denoted by O_1) might be used as an input to other units or neurons in the network or as the network's response.

Once the network architecture is defined, the connection weights between nodes are modified until the network output is similar to the actual target output values or gets to a point where the network cannot improve its performance any further. The process of adjusting network scalar parameters of weight and bias to improve performance is known as *learning*. Weights and bias are determined by the user or the neural network software as part of calibration. The amount that the weights are changed during each of

the iterations is usually defined by two additional parameters known as *learning rate* and *momentum*. A larger learning rate causes a larger weight change. Momentum allows the weight change to be proportional to the previous weight change. The actual value of the weight specifies the strength of the positive or negative influences. The weights determine whether it is possible for one unit to impede or activate the receiving unit. A positive weight causes the associated input to have an increasing effect on the weighted sum and therefore increase the chance of that sum exceeding the threshold value. In contrast, if the weight is negative, then it causes the associated input to have a decreasing effect on the weighted sum and therefore decrease the chance of that sum exceeding the threshold value.

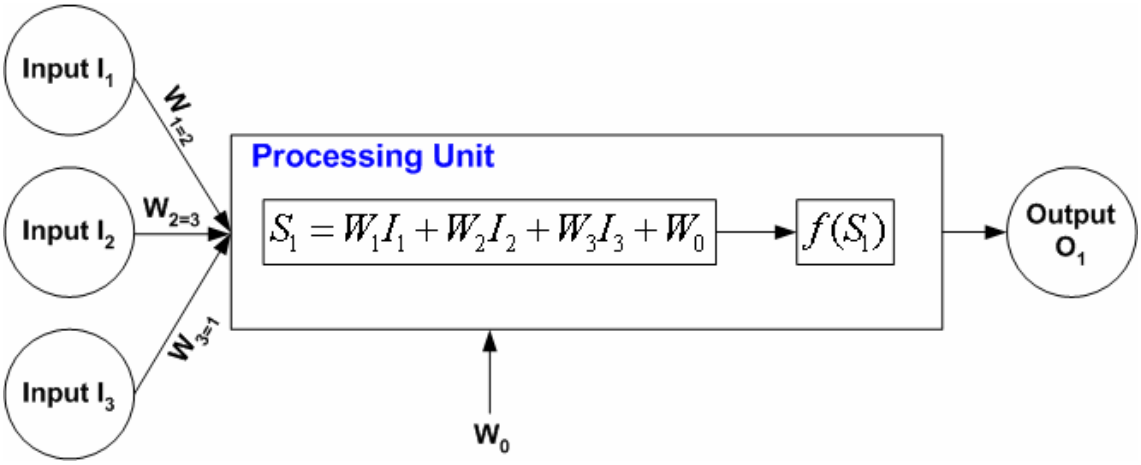

Figure 3.1: The structure of a neuron with three inputs and one output

If the weights are modified in comparison to the desired outputs, learning is *supervised* (Reed and Marks, 1999). A supervised learning strategy is based on the definition of an error measure. This procedure is performed on historical data and normally the error is defined as the difference between the output of the ANN and a pre-specified externally desired signal or output. *Unsupervised* learning strategy also known as self-organisation learning or learning by observation and discovery is another learning method. This strategy has no external teacher or supervisor and normally modifies the network weights based on pre-specified internal rules of interaction (Deco and Obradovic, 1996).

There are number of training algorithms that can be used to train a network, each specialising in a particular area. The Back Propagation algorithm (a supervised training procedure) is the most commonly used learning algorithm (Rumelhart, Hinton and

Williams, 1986; Zurada, 1992; Werbos, 1994). This algorithm is based on a very simple mathematical operation and is not difficult to implement. However, one major problem in the use of the Back Propagation algorithm is the long training time. For any given problem, the network must have many pre-specified external desired examples presented repeatedly during the learning procedure before the network learns the underlying relationships in the data. This process can be shortened by the use of an optimisation algorithm such as the Conjugate Gradient Descent and the Levenberg-Marquardt algorithms (Gill, Murray and Wright, 1981; Levenberg, 1994; Bertsekas and Tsitsiklis, 1996). Overall, these algorithms have their own computation and storage requirements, and no one algorithm is best suited for all problems. An algorithm needs to be chosen based on the characteristics and the requirements of the problem to be solved.

Briefly, a simple neural networks model can gain an insight into the relationships and patterns hidden in a set of data by using the following procedure:

i. Start training the network to do a particular job by presenting the network with a combination of inputs and outputs (training examples) from historical data.
ii. Each processing unit computes a weighted sum of its inputs and generates an output by a transfer function.
iii. The model determines how closely the actual output of the network matches the desired output.
iv. If the response is not the desired output, the user can change the weight or bias parameters of each connection, or perhaps the network itself will adjust these parameters to achieve a better approximation of the actual target output value.
v. Repeat the process until the model can associate input vectors with specific output vectors, or perform other desired tasks.

To avoid overfitting issue and provide optimal solutions, a validation (tuning) set or lot of trial and error experimentation can be used to determine the optimal network topologies and training parameters (Tu, 1996). The final state of the network can be saved to do a particular job (e.g. to classify unseen input vectors or to make predictions).

Of the different network topologies, the feed-forward multilayer perceptron (MLP) is the most commonly used. Figure 3.2 shows a simple MLP model for a scenario in kidney transplant outcomes prediction. In this scenario, the input signals are Patient Blood Group, Graft Number, Sex and Age. As can be seen in the Figure 3.2, the first layer consists of four nodes (also known as input signals). The second layer (also known as the hidden layer) consists of three processing units. The outputs of the four input signals are used as inputs to each of the units in the second layer. The inputs to the units in second layer are associated with weights of each of the units in the first layer. The receiving nodes multiply each of their input values by the associated weights and then the weighted inputs are summed by neurons. After the sum is calculated, an activation function is applied to convert the signal to an output. The outputs of the hidden nodes are used as the inputs to a node in the last layer (output layer). This node is the last processing element in this example.

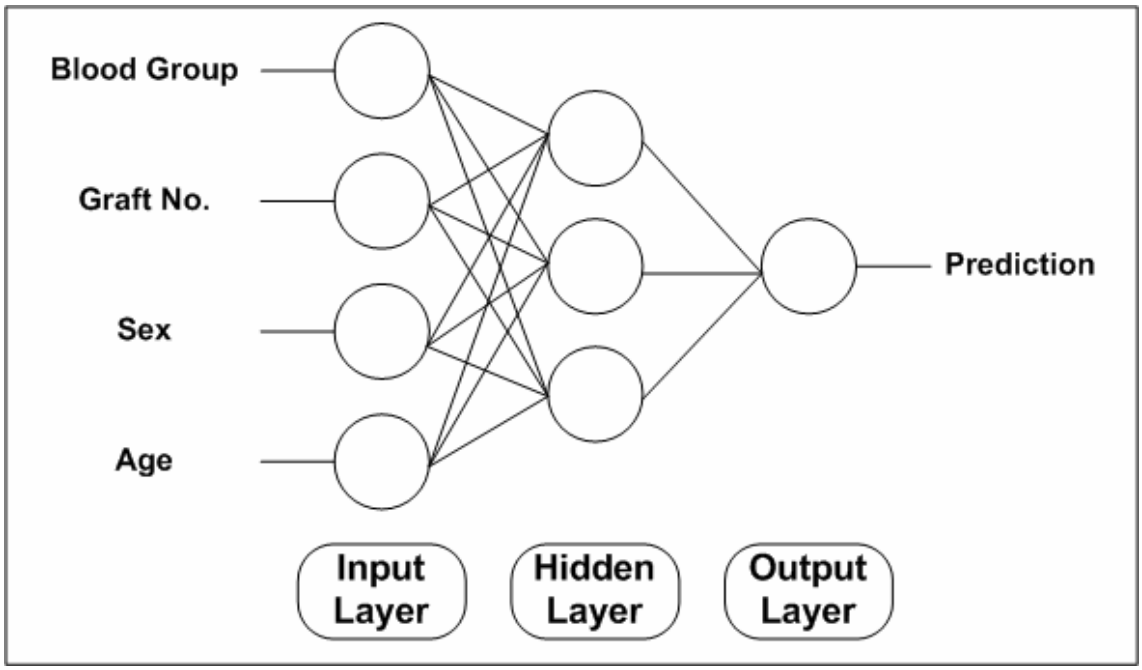

Figure 3.2: Network architecture for a kidney transplant outcome prediction

It should also be noted that there is a fair amount of judgment in the use of ANN classifiers, which generally can be classed into two distinct categories:

i) The pre-processing of the dataset into a form suitable for input into the ANN as training data. This includes decisions about appropriate parameters to be

included in the dataset and the input representation, as well as the size of training, testing and validation set.

ii) The selection of the ANN training parameters (number of nodes, training epochs, the training constants and output representation).

3.2.3. Decision trees

ANN techniques have made a significant contribution in a number of different medical domains. The ANN models are powerful prediction techniques, however, they are known as black boxes as how the outputs that are produced are not obvious. Due to this limitation ANN models are not widely used by medical professionals.

Over the past twenty years, decision tree learning models have received a great deal of attention in the ML field. Decision tree algorithms usually require shorter training times than other ML algorithms such as neural networks. They also have the ability to model linear or non-linear relationships with logical rules (Mitchell, 1997; Witten and Frank, 2000).

Decision trees and rule generating data mining systems have been implemented successfully in a number of different domains. A survey paper on decision trees by Murthy (1998) provides a good multidisciplinary survey of recent real world applications of decision trees. This study revealed that decision trees can assist detection of microcalcifications in mammography (Woods et al., 1993), analyse Sudden Infant Death Syndrome(SIDS) (Wilks and English, 1994) and diagnose thyroid disorders (File, 1994).

A decision tree model is represented by a tree structure. In general, a learning model is a multi-stage decision process that starts with an initial set of datasets, which consists of various observations or cases for which a known class label has been assigned. In any given dataset, the segmentation algorithms look at known facts stored in a knowledge base and perform a series of tests in a specific order. At each stage of this process a binary decision is made and some data samples are separated into subsets with greater purity in terms of the class membership. This process usually continues until no more rules can be found or some stopping criterion is fulfilled.

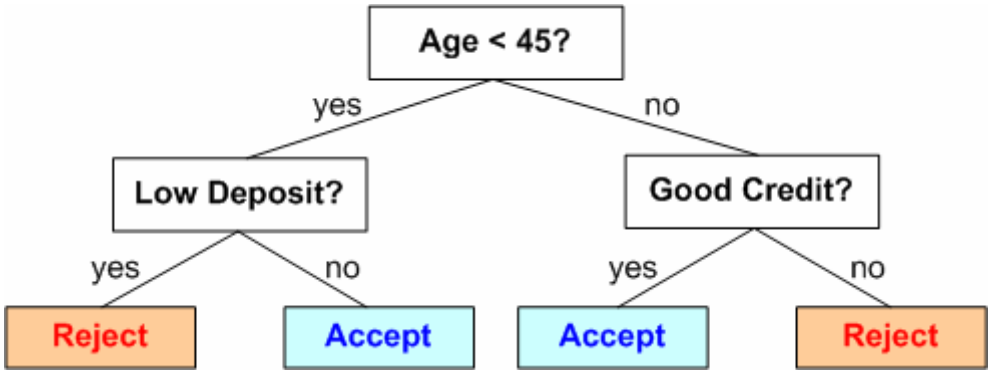

Figure 3.2: A simple classification tree for two classes (Accept for a home loan or Reject)

Generally, tree structures are represented by a root node, internal nodes (i.e. nodes that are divided into two child nodes), terminal nodes (nodes with no child) and branches. The terminal nodes of the tree are usually known as leaves. After presenting a feature vector to a classification tree, a decision or classification can be made by going down from the root node of the tree towards the leaf nodes (see Figure 3.2).

As demonstrated in Figure 3.2, if for example the value for the Age feature is less than 45, then the user needs to move to the left. The features that are used to split nodes near the top of the tree are considered to be the most important variables. This process continues until the user reach one of the terminal nodes. At this stage a new fact is determined. Leaf nodes can give a classification that applies to all instances that reach the leaf or a probability distribution over all possible classifications. The constructed tree can be converted into an equivalent set of If-Then type rules, which can be used to predict properties based upon the values of various features. For example, the constructed tree in Figure 3.2 can be converted into the following set of if-then type rules:

If (Age < 45 and Low Deposit = yes) Then Reject,

If (Age < 45 and Low Deposit = no) Then Accept,

If (Age ≥ 45 and Good Credit = yes) Then Accept,

If (Age ≥ 45 and Good Credit = no) Then Reject.

Recording the test outcomes and the leaf-node classification as a set of rules improves the readability of results, as they are easier for human user to understand.

3.2.4. Ensemble classifiers

An ensemble classifier consists of several individually trained classifiers that are jointly used to solve a problem. This approach is to focus on altering the training process and combine the output of several classifiers (for example neural network models). This technology has already been successfully applied to many domains such as *Face Recognition* (Gutta and Wechsler, 1996; Huang, et al., 2000), *Image Analysis* (Cherkauer, 1996; Asker, 1997) and *Medical Diagnosis* (Zhou, et al., 2002).

In general, constructing ensembles consists of two phases: a training phase and a combining phase (Zhou and Tang, 2003).

Training phase- The most popular methods for generating training sets for classifiers are boosting (Efron and Tibshirani, 1993; Freund and Schapire, 1996) and bagging (Brieman, 1996). These two methods generate multiple classifiers by resampling (Quinlan, 1996; Bauer and Kohavi, 1999).

Boosting was originally proposed by Schapire (1990) and later was improved by Freund *(1995).* It generates chains of component classifiers whose training sets are determined by the performance of former ones. Training instances that are misclassified by former classifiers will contribute more to the training of later classifiers.

Bagging (bootstrapping aggregates) was originally proposed by Breiman (1996). Bagging is a popular method for training component neural networks. This technique generates several training sets using random sample drawing (with replacement) from the original training set. Consequently, in every new training set there are data points which appear more than once while others do not appear at all. Each individual classifier is trained with each of the training sets. As a result, each classifier may produce a different prediction. Bagging is a good technique for improving a poorly performing classifier, especially where a classifier has been presented with a small training sample set or training set that includes misleading data points (Cox, Clark and Richardson, 1999).

Combining phase- In this phase, a variety of approaches can be used in order to combine the predictions of the component classifiers.

Averaging and majority voting strategies are the most popular approaches for classification tasks. In averaging strategy (Opitz and Shavlik, 1996), all predictions of the multiple classifiers contribute to final prediction by computing their average.

Majority voting (Hansen and Salamon, 1990) is similar to averaging, however, in this strategy each component classifier votes for a category and the category with the majority of votes defines the ensemble.

Using the results of several classifiers is a technique which has been shown to offer significant improvement in prediction accuracy (Brieman, 1996; Freund and Schapire, 1996; Cox et al., 1999). This area of research is currently under active development (Dietterich, 1997). Berrar et al. (2003) performed a comparative evaluation of six different ML techniques namely, decision trees, ensemble of decision trees using boosting, support vector machine (SVM), k-nearest neighbour classifier (k-NN), probabilistic neural networks (PNNs) and MLP ANN. They used lung cancer data to predict the survival of risk groups. Their study revealed that the best overall classification on unseen test data was obtained by an ensemble of decision trees using boosting strategy. A similar study by Katz, et al. (1994) confirms that better performance can be achieved by combining different techniques.

3.3. The RIDC-ANNE methodology

Neural networks models have been used by many researchers and provide good predictive accuracy in a wide variety of domains (Baxt, 1991; Ashutosh et al., 1992; Tu and Guerriere, 1993; Doyle et al., 1994; Baxt, 1995; Mobley et al., 1995; Ortiz et al., 1995; Itchhaporia et al., 1996; LaPuerta et al., 1998; Pantazopoulos et al., 1998; Dayhoff and DeLeo, 2001; Ennett et al., 2001; Ramesh et al., 2004; Gabutti et al., 2006; Mueller et al., 2006). This thesis presents and evaluates a novel hybrid neural networks model, called RIDC-ANNE for the purpose of prediction of medical outcomes. This approach attempts to improve data quality by configuring an ensemble of bagged networks as a filter and identifying the regions in the data space that have been consistently misclassified or have high impact on the system performance. Furthermore, it provides some clarity for the general behaviour of an ensemble. Overall, for the purposes of this study, the following methodology was employed (see also the pseudo-code of RIDC-ANNE algorithm in Table 3.1):

1. Pre-process the dataset. This includes: extracting the data from different tables, cleaning the data, transforming the nominal attributes into numeric attributes, removing variables that are actually an indication of the

outcomes, choosing the appropriate parameters to be included in the dataset with the help of domain expert and normalisation. This step is also known as data discovery and cleaning, where the user have to decide whether quality of data is satisfactory for the goal or problem that has defined in one of first steps of KDD. Split the dataset into three equal parts for training, overtraining prevention (also known as tuning) and testing (mainly with balanced distribution of success and failure cases).

2. Perform classification using a series (n=500 or n=100) of Multilayer Perceptron (MLP) networks that were trained independently, to differentiate between two classes. This is an important phase of KDD that must precede actual analysis of the data. The next step is usually Data Analysis, which is in general an attempt to understand why certain groups of entities are behaving on the way they do, it can also include search for rules of such behaviour. The model can be refined in iterations according to the evaluation results.
3. Generate new training sets by extracting the patterns (examples) that were consistently causing x% agreement (for x= 50 to 100) across the ANN classifiers, in the testing phase.
4. From the new training sets generated in Step 4, choose a reasonably big training set that has provided both a good level of accuracy (based on its corresponding classification table) and a reasonable amount of model agreement.
5. Modify the new training set for the rule generation process by replacing the desired classes of all cases with their corresponding class labels (i.e. the class assigned by the trained ensemble).
6. Grow a decision tree (Quinlan, 1996; Quinlan, 2000) from the selected samples generated in Step 6.
7. Analyse the results.

It is important to note that the method makes no assumptions about how the tree is generated and the decision tree can be any standard classification decision tree (such as generated by C4.5). The method requires (1) a neural network model, (2) a decision tree, (3) a classification cost matrix, and (4) a set of historical data. All neural network

classifiers described in this study are feed-forward, back-propagation networks and are implemented by the Delphi programming language. All networks consist of a set of input neurons, a set of neurons in the hidden layer and two output neurons. Sigmoid transfer functions are used for both the hidden layer and output layer.

Table 3.1: The RIDC-ANNE algorithm

Given a training set *S* (with *n* cases and their desired classes), learning algorithm *L*, number of bootstrap samples *T* and a test set *V*, the following steps was used to identify certain regions in the selected data space and generate rules from an ANN ensemble:

Step 1: Train and test an ANN ensemble.

1. Model Generation and Training

For i = 1 to T:

Generate a new training set (bootstrap sample) with n *cases, using random drawing (with replacement) from* S.

Apply the learning algorithm L *to the bootstrap sample.*

Keep the resulting neural network model M_i *for future use.*

2. Prediction

Repeat the following procedure for every case in the test set V*:*

For i = 1 to T:

Predict class of case using M_i

Return class that appears most frequently.

Step 2: Obtain a list of training data as the possible candidates for rule induction.

For i= T/2 to T

Generate training set N_i *by extracting all cases and their desired classes in test set* V *that have caused agreement on one category (fail or success) across at least* i *models.*

Modify the new training set N_i *by replacing the desired classes of all cases with their corresponding class labels (i.e. the class assigned by the trained ensemble).*

Produce the classification table C_i *by comparing the corresponding class labels output from the ensemble and the desired classes.*

Keep the resulting training data N_i *and* C_i *for future use.*

Step 3: Extract rules from the most appropriate training set.

First, choose a reasonably big training set from N_T, N_{T-1}, N_{T-2}… $N_{T/2}$, *that has provided both a good level of accuracy (based on its corresponding classification table* C*) and reasonable amount of model agreement.*

Then, grow a decision tree (e.g. a C4.5 *tree by using* S-Plus *software system) from samples and their new class labels in the chosen training set. Finally, generate rules by converting the most appropriate leaf in every path into a rule.*

3.4. The evaluation and assessment procedure

The binary classifiers constructed in the previous section were evaluated by using the measurement of classifier accuracy scheme. In the scheme, accuracy is defined as the percentage of the classifier's predictions that are actually correct as measured against the known classes of the unseen evaluation data. As discussed earlier in Section 2.3, this scheme is not always a reliable strategy, particularly when the task is to evaluate different kinds of classifiers (Provost et al., 1998).

For the purposes of the research presented in this thesis, it is possible to simply assume the best classifier is the classifier with the highest proportion of correct classifications and that does not differentiate between false positives and false negatives when a binary classification is made. Furthermore, the focus of this research is on neural network classifiers and, in most cases, all classes (after the pre-processing stage) of the examples in selected databases (particularly for Kidney Transplant data) are distributed in a relatively balanced fashion. Therefore, the measurement of classifier accuracy scheme can be considered as the mainstream evaluation methodology for the results presented in this thesis. For the sake of completeness, the sensitivity and specificity of important experiments has also been obtained and is presented in Section 4.4. The True Positive rate (also called hit rate and recall) of a classier is estimated as:

$$\text{TP rate} = \frac{\text{positives correctly classified}}{\text{total positives}}$$

The False Positive rate (also called false alarm rate) of a classier is estimated as:

$$\text{FP rate} = \frac{\text{negatives incorrectly classified}}{\text{total negatives}}$$

Additional terms associated with the proposed evaluation methods are:

$$\text{Sensitivity} = \text{Recall}$$

$$\text{Specificity} = \frac{\text{True negatives}}{(\text{False positives} + \text{True negatives})} = 1 - \text{FP rate}$$

For example, sensitivity for diabetes can be defined as the probability of the respective systems to classify subjects with high blood pressure as diabetic (true positives). Specificity can be defined as the probability of classifying subjects without high blood pressure as non-diabetic (true negatives).

3.5. Discussion: Why the model works

One of the main reasons for necessity of automated computer systems for intelligent data analysis in medicine can be the rapid development of data collection and storage technologies that has lead to formation of large number of databases. Data mining techniques have the potential to be successfully used to automate the process of finding the underlying relationships and patterns presented in large datasets.

Recently, there has been extensive research on hybrid and integrated systems. The literature reveals that combinations of data mining techniques or training methods have great potential to be effectively used in predictive modelling tasks. In addition, the literature shows that employing a methodology to develop a best possible representation of the structure of the data and choosing the right set of data can be considered key part of a successful data mining.

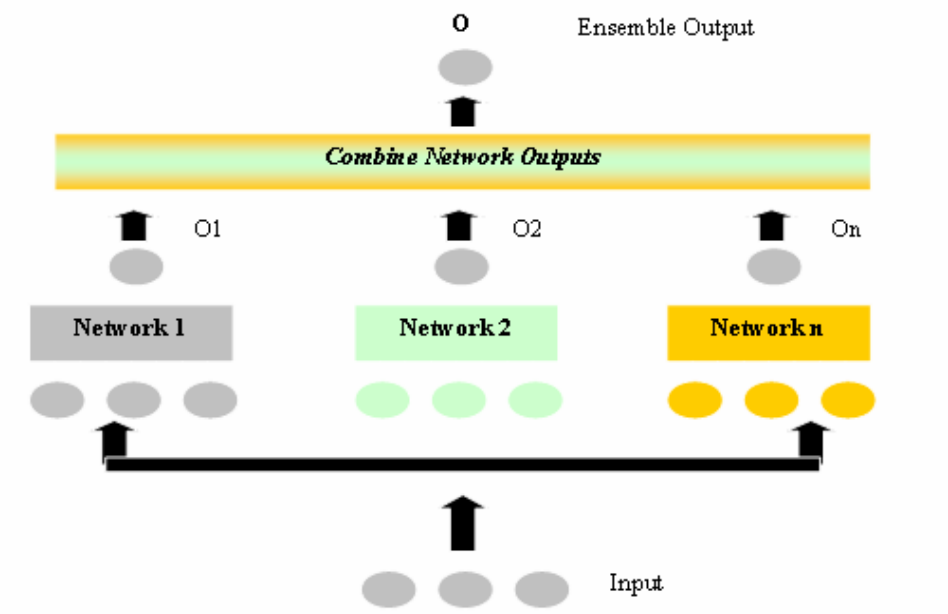

Figure 3.3: The general structure for a neural network ensemble.

Thus, the above factors are the basis of RIDC-ANNE approach. Neural networks have the ability to provide good solutions for noisy clinical data or in situations where large numbers of variables contribute to an outcome but their individual influence is not well understood. To better address the complexity of clinical data, a variance-reducing method known as Bagging (bootstrapping aggregates) algorithm was chosen for training an ensemble of neural networks (see Figure 3.3). Training the network in this way is a

promising approach that can provide better generalisation performance, especially where a classifier has been presented with a small or noisy training dataset (Cox et al., 1999). Unfortunately, neural network models do not have the ability to induce symbolic representation from data and generate a set of rules that can be understood by humans. Users often wish to translate a neural network system into a more understandable model. So it is natural to extend the task of prediction to provide transparent justifications for the outputs generated by the system. Table 3.2 shows summaries of some of the important ML-based techniques that have been described in this study.

3.6. Summary

This section outlined the use of the RIDC-ANNE algorithm for the prediction of outcomes of medical events. The proposed classification algorithm is based on combination of initial data preparations, preliminary classification by ensembles of ANNs, and generation of new training data based on criteria of highly accuracy and model agreement. Furthermore, the model will also generate the decision tree models to provide classification of data and the prediction of results. The RIDC-ANNE approach facilitates data mining by configuring an ensemble of bagged networks as a filter and combining the power of a variance-reducing method such as bagging and black box connectionist learning systems such as ANN with transparent rule based decision-making methodologies.

As described in Section 2.6, the RIDC-ANNE approach is based on the rule extraction approach of (Wall et al., 2003; Zhou and Tang, 2003; Zhou, 2004). This approach attempts to improve data quality by configuring an ensemble of bagged networks as a filter and identifying the regions in the data space that have been consistently misclassified or have high impact on the system performance. The RIDC-ANNE approach has the ability to extract the input patterns (examples) that were included across the neural networks series in the final results. In this strategy the ensemble of networks computes the target concept and the input vectors are the actual network's input vectors. This strategy also considers the diversity and expertise of the component networks in the rule generation process. However, unlike the previous method of Wall et al. (2003), RIDC-ANNE does not focus on identifying the ensemble members that are relevant in explaining the prediction (output) associated with a

particular case. Instead, it uses the output of an ensemble based on a cluster of cases that consistently generate agreement (on their class labels) across the classifiers with similar expertise. This form of approach offers a simpler explanation for the ensemble behaviour and helps the user investigate different regions of data space.

In the next chapter, the RIDC-ANNE approach is tested with different datasets and its results is compared with those from other studies. The RIDC-ANNE is applied to three datasets of clinical relevance. Two of sets, Pima Indians Diabetes and Wisconsin Cancer, are well known sets available at the UCI machine-learning repository, which have been studied extensively by machine learning community. These datasets represent the benchmark cases for the assessment of RIDC-ANNE performance. The third dataset, ANZ Kidney Transplant database, has not been studied extensively by machine learning method and represent a novel application of RIDC-ANNE strategy.

Table 3.2: Summary of the ML-based techniques

Techniques	Examples	Explanation Strategies	Comments
Decision Trees	C4.5	Logical Rules or Decision Trees Knowledge Representations	-Have the ability to model linear or non-linear relationships with logical rules -Not very difficult to construct -Do not require long training time -Overfitting can accrue if a model is presented with small, incomplete and noisy training data
Neural Networks	MLP	N/A	-Very powerful for modelling linear or non-linear relationships but do not have explanation facilities -Good potential for problems of high complexity -Require long training time
Ensemble Classifiers	Neural Network Ensemble (ANNE)	N/A	-Powerful predictive models for improving classifiers with poor performance, especially where classifiers are presented with small training examples with misleading data points - Some models such as ANNE do not have explanation facilities and require very long training time
Hybrid Decision Tree-Neural Networks	KT	Local	-Have the ability to model linear or non-linear relationships and enhance comprehensibility of ANN by extracting symbolic rules from ANN -Model can be used in conjunction with expert judgment -Local strategies may produce rules with better quality, but the execution time of the local algorithms may increase exponentially with the number of units within a trained neural networks
	SUBSET	Local	
	TREPAN	Global	
Hybrid Decision Tree-Neural Networks Ensembles	CMM	Global	-Have the ability to extract symbolic rules from ANNE models -Like other ensemble classifiers, they can be useful where a classifier has been presented with a small training sample set or training set with misleading data points -Local strategies may produce rules with better quality however usually require longer execution time
	Wall's Case-by-Case Strategy	Local	
	C4.5 Rule-PANE	Global	

Chapter 4

Use of ANN for medical outcome prediction

4.1. Introduction

For many classification and prediction modelling tasks obtaining large amounts of training data in order to build accurate models is often difficult due to technical and economic constraints. In some cases, it may be possible to acquire a large amount of data, but high quality training data is often quite limited. Hence, it is important to build accurate classifiers and study the data in order to be able to identify suspect data and generate a subset of the data that has higher information content than the original data.

The present chapter investigates the use of ANN and focuses on adapting the training process and combining the output of several classifiers in the RIDC-ANNE approach. Using the results of several classifiers has been shown to offer a significant improvement in prediction accuracy (Quinlan, 1996; Cox et al., 1999).

The case studies presented in this chapter are from kidney transplant (a complex scenario), Pima Indian Diabetes (a semi-complex scenario) and Wisconsin Cancer (a non-complex scenario) datasets. The last two datasets are stored in the UCI repository[5] and are frequently used as benchmark data (Mangasarian and Wolberg, 1990; William and Mangasarian, 1990; Duch, Adamczak and Grabczewski, 2001).

4.2. Clinical problem domains

AI-based data mining tools have been successfully used in a number of medical domains. A good example is the famous Pima Indians diabetes. This dataset is based on the personal data of the Pima Indians, which originally was obtained from the US National Institute of Diabetes, Digestive and Kidney Diseases. All patients in this dataset are females of Pima Indian heritage and at least 21 years old. The database

[5] http://www.ics.uci.edu/pub/machine-learning-databases

contains 768 data samples taken from patients who may have shown signs of diabetes. All the data samples in this dataset have no missing attributes. There are eight attributes (inputs) in this database and two output classes, diabetes and non-diabetes. The attributes are age, pregnancy information and medical measurements (see also Appendix B). Over the years a large amount of research has been conducted using this database (Cheung, 2001; Duch et al., 2001). Interestingly, a comparison study conducted by Duch and Adamczak (2001), reported only a 77% accuracy rate for the best classification performance for this dataset (see also Ripley, 1996).

The Wisconsin Breast Cancer dataset is another popular dataset that has been used as a benchmark by many researchers (Mangasarian and Wolberg, 1990; William and Mangasarian, 1990; Cheung, 2001; Duch et al., 2001). This is a relatively clean and non-complex dataset. The dataset was originally obtained from the University of Wisconsin Hospitals, in Madison by Dr William H. Wolberg. The dataset has nine attributes (inputs) and two output classes. All nine inputs are continuous and range from 1 to 10. The database contains 699 samples, with 683 samples of complete data and 16 samples with missing attributes. Each of the 683 available instances is labelled as either Benign (444 instances or 65 % of data) or Malignant (35%). The task is to predict benign or malignant classes (see also Appendix B). The classification accuracy of this database has been approximated to 90% or higher. The same comparison study reported by Duch and Adamczak (2001) indicated 97% accuracy for the best classification performance.

Recently, there has been substantial research into predicting graft outcomes and detecting the key parameters influencing graft outcomes (Doyle et al., 1994; Matis et al., 1995; Rapaport, 1995). The economic and social benefits of accurately predicting the success of a graft such as kidney are very high. A kidney transplant is a surgical procedure to implant a healthy donor kidney into a patient with kidney failure. A successful kidney transplant will not only improve the quality of life for the recipient but also reduce medical expenses. Unfortunately, the factors that determine the successful outcome of graft transplants are still unclear. The following factors are known to influence the outcome of transplants:

- the compatibility of blood types between the donor and recipient;

- the number of Human Leukocyte Antigen (HLA) mismatches. This is the general name of a group of genes in the human cells that enable the body to tell the difference between its own cells and foreign ones; and
- the results from basic cross-match tests. This involves a mixing of cells and serum to determine whether or not the recipient of a kidney will respond to the transplanted organ by attempting to reject it.

Currently, the medical professionals world wide use the above factors and tests to determinate the compatibility of donor kidneys and recipients.

This study considers the use of neural networks models for the prediction of kidney transplant outcomes by using a small trial dataset from a kidney transplant database registry (ANZDATA, 2000). The main variables retained for this study are shown in Table 4.1.

The challenge for this study is to select an appropriate kidney from the available pool of organs for a particular patient, thereby maximising the chances of a successful transplantation. The pre-processing stage for kidney transplant data includes: extracting the data from different tables, cleaning the data, transforming the nominal attributes into numeric attributes, and choosing the appropriate parameters to be included in the dataset with the help of a domain expert. This dataset has been greatly corrupted with missing features and random errors in the values of the features. The dataset stands as a good example of clinical data with complex characteristics.

Some variables from these tables were removed because they were actually an indication of the outcomes of the transplant, and as such they were measured after the transplant had been made (see Appendices B and C). The data was also sampled at the six month and the two year rejection horizon. It was anticipated to predict the success or failure of the kidney in the six months or two years following the transplant.

Table 4.1: The main variables selectd for the purpose of this study.

No.	Variable/code	Description	Type and size
1	AGE	Age at transplant (Recipient)	NUMBER (2)
2	MISA	Number mismatches A	NUMBER (3)
3	MISB	Number mismatches B	NUMBER (3)
4	MISD	Number mismatches DR	NUMBER (3)
5	MISDQ	Number mismatches DQ	NUMBER (3)
6	REFHOSP	Referring hospital	CHARACTER (4)
7	REFSTAT	Referring state	NUMBER (1)
8	DONHOSP	Donor hospital	CHARACTER (4)
9	DONSTAT	Donor state	NUMBER (1)
10	TRANHOSP	Transplant hospital	CHARACTER (4)
11	TRANSTAT	Transplant state	NUMBER (1)
12	CMV	Recipient CMV antibody status	NUMBER (1)
13	EBV	Recipient EBV antibody status	NUMBER (1)
14	DONSOURC	Donor source	NUMBER (2)
15	DONAGE	Donor age	NUMBER (2)
16	DONSEX	Donor sex	CHARACTER (1)
17	ISCHEMIA	Total ischemia (to nearest hour)	NUMBER (2)
18	MULTIPLE	Has recipient had another organ transplanted?	CHARACTER (1)
19	BLTRANA	Ever transfused? (Before the first graft only)	NUMBER (1)
20	BLTRANB	Number of units transfused	NUMBER (2)
21	INSITU	Insitu Y?	CHARACTER (1)
22	KIDPRESI	Initial kidney preservation	NUMBER (2)
23	KIDPRESM	Machine kidney preservation	NUMBER (1)
24	TXSTATUS (Class Label)	Did graft succeed or fail? 0=functioning 1= failed	NUMBER (1)

4.3. Results for kidney transplant data using single MLP

This section presents the experimental results for the following experiments based on the kidney transplant data:

Experiment 1

Attempts to reproduce the previous results of by Khum (2001) and Petrovsky et al. (2002), using the same pre-processing strategy reported by them and using the original data and 6-month rejection horizon.

Experiment 2

In order to make a fair comparison, it was decided to contact Khum (2001) and Petrovsky et al. (2002), and obtain their data in a pre-processed format, and then use that data without any modification in the experimental set up described in previous section.

Experiment 3

It was decided to construct a slightly different pre-processing methodology, using the original data. In doing so, the rejection horizon was changed from six months to two years. It was also decided to remove the experimental bias found in Experiment 2.

Experiment 4

In an attempt to determine sensitivity to individual factors, each input variable was systematically removed in turn before re-training the ANN. An important point to note here is that in this experiment it was decided to run 30 experiments (changing the initial random allocation of weights in the ANN) and report the averaged of results.

Experiment 5

The data was pre-processed further by removing some variables and the previous experiments were replicated.

4.3.1. Experiment 1: replicate the previous experimental results

The objective of this experiment was to replicate previous results reported by Khum (2001) and Petrovsky et al. (2002), in which an ANN (MLP network) was trained to predict whether a kidney transplant was likely to be a success or failure. In this experiment they reportedly achieved a prediction accuracy of 84% for successful transplants and 71.7% for unsuccessful transplants.

Using the classification scheme reported in the previous studies (Khum, 2001; Petrovsky et al., 2002), only a prediction accuracy of 58.94% was achieved for successful transplants and 53.75% for unsuccessful transplants. This was considerably below the previous experimental results (of 70% to 84% accuracy).

4.3.2. Experiment 2: using the previous pre-processed inputs and a MLP

Having received the pre-processed data from the original researchers (Khum, 2001; Petrovsky et al., 2002), it was decided to feed this data into a single MLP model constructed for the purposes of this study. The result revealed an accuracy rate of 67% in the prediction of successful transplants and 98.5% for unsuccessful transplants. This was a much better result than the 70% and 84% accuracy rate previously reported.

The data points used in experiment 4.2.1 and this experiment were compared to determine the cause of the large difference in the accuracy rates. This investigation revealed slightly different selection strategy was used by Khum (2001) and Petrovsky et al. (2002) during the pre-processing stage of the successful and unsuccessful records. In their pre-processing stage, they decided to remove the records with missing information in the successful dataset and keep the records with missing information in the unsuccessful dataset. Therefore, the neural network simply had to learn that if a column had a missing data point (which was represented by the number –1), it was a fail point. Figure 4.1 shows the size of missing data points in each target category in their training dataset.

The missing data distribution shown in Figure 4.1 has an impact on the network performance because of the unbalanced missing points in each category. It is easy to see that the missing data is skewed to favour one or the other target category. This is

particularly true for Variable 5 but is also visible in Variables 4, 9,12,13,17,19,20,22 and 23.

The skewing of missing data is a plausible reason for the high neural network accuracy rate reported in the previous study (Khum, 2001; Petrovsky et al., 2002). It was suspected this was done mainly because significant numbers of records belong to the successful transplant category and data cleaning can be done without dramatically reducing the size of data. Therefore, in order to match the size of successful dataset, records containing unsuccessful transplants with low quality points (ones with missing data) were not removed from the unsuccessful dataset.

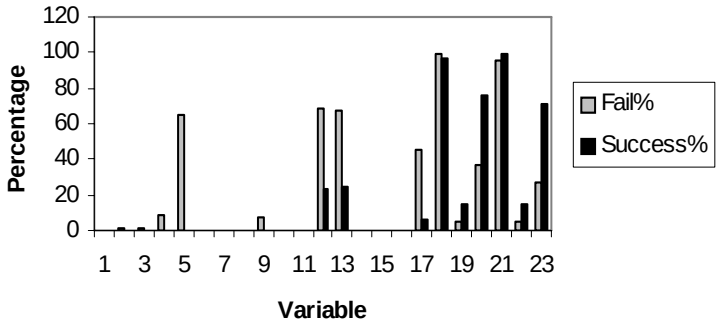

Figure 4.1: The percentage of missing data points in fail and success target category for the 23 variables

4.3.3. Experiment 3: pre-processing with slightly different methodology

In this experiment it was decided to employ a new pre-processing methodology and pre-process the data in a slightly different manner to avoid creating variables with a skewed distribution of missing data. Additionally, the pre-processing rejection period was changed from six months to two years. Overall, the following steps were followed:

Step 1: Remove all non-first graft records

Step 2: Remove all records (for both success and failure categories) that have no HLA-DQ information.

Step 3: In the sex column, replace Male with numeric value of 0 and Female with a value of 1.

Step 4: Remove those patients who were lost in the follow up.

Step 5: Split data into different files such as: success.txt, fail.txt

Step 6: Create a target column with the following values: 1 for failed grafts and 2 for success grafts

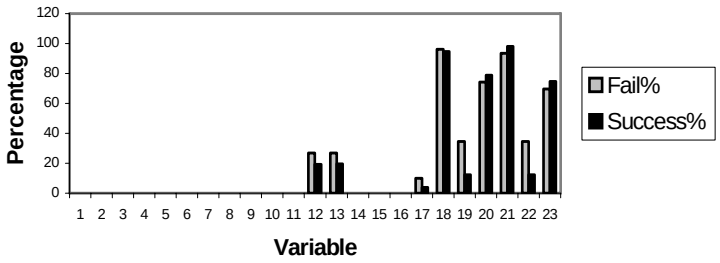

Figure 4.2: The percentage of missing data points in fail and success target category for the 23 variables after the new pre-processing strategy.

This strategy produced a final dataset of 672 fail points. Furthermore, 672 success cases were randomly sampled from the file containing 2371 success cases. A summary of the relevant findings for the final 23 columns used in this experiment's training process are shown in Figure 4.2. In this experiment all 23 attributes described in Table 3.1 were used as inputs of the neural networks. For this experiment the overall accuracy rate for validation set reached only 61.6%.

4.3.4. Experiment 4: search for a Subset using a single MLP

The objective of this experiment was to improve the performance of the network classifier and find the best subset of features by using different combinations of variables. It was decided to remove one column of data at a time to see if any single column of data had a significant influence on prediction accuracy.

As can be seen in Table 4.1, the highest accuracy rate is obtained when the variable number 14 (donor source) was removed. However, no single variable had significant effect on prediction accuracy and the results were still around 62.8%. It was concluded that despite using a range of pre-processing strategies for prediction of outcomes of kidney transplants, the resultant accuracy of approximately 62% was too low to be of any clinical use (Shadabi, et al., 2004).

Table 4.2: The results for experiment 4, using MLP network.

Removed Variable	Final Accuracy Rate (%)	Final Accuracy Rate (%) "After Averaging 30 Run"
1	60.96	61.35
2	61.87	61.60
3	62.10	62.20
4	61.87	61.93
5	61.64	61.58
6	60.50	62.40
7	61.42	61.82
8	62.33	60.92
9	61.64	61.89
10	62.10	61.68
11	64.38	62.06
12	64.84	62.13
13	64.38	62.45
14	**62.79**	**62.86**
15	60.50	60.80
16	59.36	61.86
17	57.99	58.65
18	59.82	61.54
19	59.59	59.03
20	60.13	61.81
21	61.30	61.50
22	59.17	60.45
23	60.38	59.50

4.3.5. Experiment 5: pre-processed the data further using a single MLP

In this experiment it was decided to not include variables with a significant amount of missing data or variables with a slightly skewed distribution of missing points. The final 16 variables that were retained were: AGE (recipient age at transplant), MISA (number of mismatches A), MISB(number of mismatches B), MISDR (number of mismatches DR), MISDQ (number of mismatches DQ), REFHOSP (referring hospital), REFSTAT (referring state), DONHOSP (donor hospital), DONSTAT(donor state), TRANHOS (transplant hospital), TRANSTA (transplant state), DONSOUR (donor source), DONAGE (donor age), DONSEX (donor sex), ISCHEMIA (total ischemia to nearest hour),and KIDPRESI (initial kidney preservation). The previous experiments (i.e. a single MLP with a single hidden layer and two output neurons) were repeated using the

new list of variables. Once again the validation phase gave a low level of overall accuracy (about 61%).

4.4. Results for kidney transplant data using RIDC-ANNE

The possibility of building a model that could provide better prediction for kidney transplants data was investigated by using a hybrid-learning model RIDC-ANNE. This section presents the experimental results for the prediction of kidney transplant outcomes:

Experiment 1

In this experiment, the pre-processing steps provided a final dataset of 657 fail points and 657 success points. After combining these two files, one third of the data was extracted for the validation set and the remaining two third was kept for training (network 1 to 100) and testing.

Experiment 2

This is the same as Experiment 1. However in this experiment, the bagging technique was used to generate different training sets from the original training sets.

Experiment 3

The steps in this experiment were very similar to the methodology used in Experiments 1 and 2. However, in this experiment 100 MLP classifiers (without bagging) were constructed and, rather than extracting 657 success points in the early pre-processing stage, it was decided to balance the data at a later stage and use the 2371 success points.

Experiment 4

This is the same as Experiment 3. However, in this experiment the bagging technique was used to generate different training sets from original training sets for the100 MLP classifiers.

Experiment 5

This is the same as Experiment 4. However, in this experiment 500 MLP classifiers (rather than 100 MLP classifiers) were implemented using the bagging technique. Furthermore, the back-end of the RIDC-ANNE approach was used to extract explanations and knowledge from several combined neural network classifiers.

4.4.1. 100-MLP classifiers without bagging

For this experiment the following algorithm was employed (see also Figure D.1 in Appendix D):

1. Pre-process the dataset. This includes decisions about appropriate parameters to be included in the dataset, the time point, etc. This gave us a final dataset of 657 fail points and 2371 success points.
2. Combine the two files (fail and success files) in a temporary file and extract normalisation parameters.
3. Perform the normalisation, using the extracted normalisation parameters.
4. Randomly extract 657 records from the file containing only success data points.
5. Combine the extracted success records (from Step 4) with the fail records.
6. Extract 1/3 of data for validation set and keep the remaining 2/3 for training and testing.
7. Implement 100 ANN classifiers.
8. For the file containing the remaining 2/3 of data, the data was randomly sampled (randomly drawing without replacement), half for training and the remaining half for testing (tuning) purpose.
9. Each ANN classifier (i.e. network 1 to 100) is trained independently of the others with each of the training sets produced in Step 7, to differentiate between successful and unsuccessful transplants.
10. Each ANN classifier (i.e. network 1 to 100) is tested independently of the others, using the validation set extracted in Step 6, to differentiate between successful and unsuccessful transplants.
11. For each example, the predicted output of each of the networks is recorded to produce the output of the ensemble.

12. The prediction accuracy, the amount of votes (agreements) between classifiers and the data points that were used to produce such agreements are recorded. We also de-normalised the validation set, using the extracted parameters from Step 2.
13. Analyse the results and predictive accuracy of classifiers (Figure 4.3).

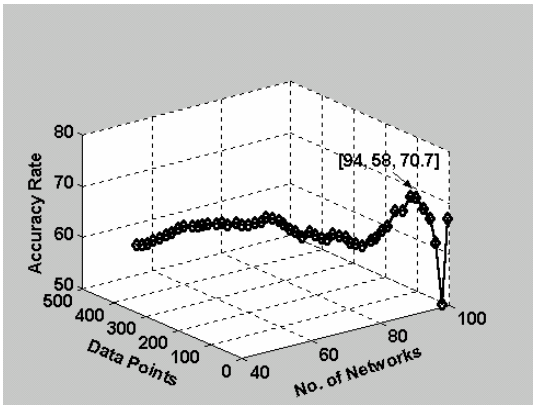

Figure 4.3: The results for 100-MLP using the first pre-processing strategy.

It can be seen as the model approaches 100% agreement it includes fewer patterns (examples) across the ANN series. In this experiment, using majority voting the balanced validation set reached slightly above a 60.5% predictive accuracy rate. However, by using only 59 examples, it reached a 70.7% accuracy rate (with 94% agreement among the networks). The results show that the overall performance of this model is generally similar to that of previous models. However, when the model uses fewer patterns, it can indeed perform slightly better (62.8% versus. 70.7%) than the previous experiments.

4.4.2. 100-MLP classifiers with bagging

The next approach taken was to implement 100 MLP neural network classifiers with bagging. For this experiment the following algorithm was employed (see also Figure D.2 in Appendix D):

1. Pre-process the dataset. This includes decisions about appropriate parameters to be included in the dataset, the time point and etc. This provides a final dataset of 657 fail points and 2371 success points.
2. Combine the fail and success datasets in a temporary file and extract normalisation parameters.

3. Perform the normalisation using the extracted normalisation parameters.
4. Randomly extract 657 records from the file that contain only success data points.
5. Combine the extracted success records (from Step 4) with the fail records.
6. Extract 1/3 of the data for the validation set and keep the remaining 2/3 for training and testing (tuning).
7. For the file containing the remaining 2/3 of the data, randomly sample the data (randomly drawing without replacement), half for training and the remaining half for testing (tuning) purpose.
8. Implement 100 ANN classifiers.
9. **Use bagging to generate different training sets for the classifiers from the training sets generated in Step 7 to increase the diversity of ensembles.**
10. Train each ANN classifier (i.e. network 1 to 100) independently of the others with each of the training sets produced in Step 8, to differentiate between successful and unsuccessful transplants.
11. Test each ANN classifier (i.e. network 1 to 100) independently of the others, using the validation set extracted in Step 6, to differentiate between successful and unsuccessful transplants.
12. For each example, record the predicted output for each of the networks to produce the output of the ensemble.
13. Record the prediction accuracy, the amount of votes between classifiers, and the data points that were used to produce such agreements. De-normalise the validation set, using the extracted parameters from Step 2.
14. Analyse the results and predictive accuracy of classifiers.

The results are shown in Figure 4.4 (see also Table E.2 in Appendix E). In this experiment, the accuracy rate for the balanced validation set (with 438 cases) was less than 60%. However, by using only 48 data points, the validation set reached an accuracy rate of 73% (with 89% agreement among the networks). At one point it reached an accuracy rate of 75% with 99% agreement among the networks, using only four examples. However, in the latter case, the accuracy is based on using only a very small number of examples.

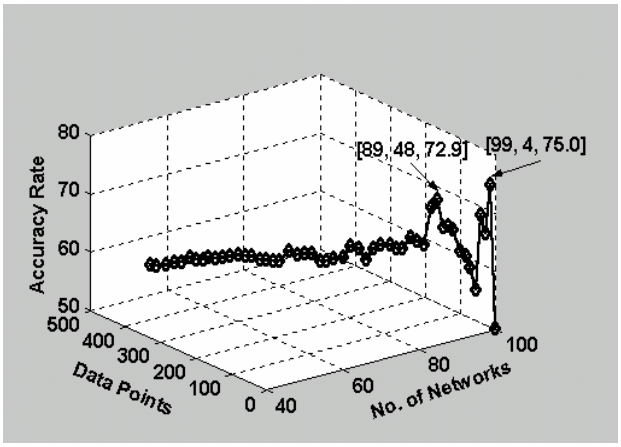

Figure 4.4: The results for 100-MLP using the first pre-processing strategy (with bagging).

4.4.3. 100-MLP classifiers with all available success points without bagging

In this experiment, rather than randomly extracting 657 success points in the early pre-processing stage, it was decided to keep 2371 success points. Therefore, for this experiment the following algorithm was employed (see also Figure D.3 in Appendix D):

1. Pre-process the dataset. This includes decisions about the appropriate parameters to be included in the dataset, the time point and etc. This provides a final dataset of 657 fail points and 2371 success points.
2. Combine the fail and success datasets in a temporary file and extract normalisation parameters.
3. Perform the normalisation, using the extracted normalisation parameters.
4. Randomly extract 1/3 of the data for the validation set from the file containing only **success** data points and keep the remaining 2/3 for training and testing.
5. Randomly extract 1/3 of the data for the validation set from the file containing only **fail** data points and keep the remaining 2/3 for training and testing.
6. For the file containing the remaining 2/3 of **success** data points, randomly sample 219 records (randomly drawing without replacement) for training and keep the remaining 219 for testing (tuning) purpose.

7. For the file containing the remaining 2/3 of **fail** data points, randomly sample 219 records (randomly drawing without replacement) for training and keep the remaining 219 records testing (tuning) purpose.
8. Combine the extracted 219 training success records from Step 6 with the extracted 219 training fail records from Step 7 to produce a training set.
9. Combine the extracted 219 testing success records from Step 6 with the extracted 219 testing fail records from Step 7 to produce a testing set.
10. Repeat Steps 6 to 9 to produce 100 training sets and test sets for ANN classifiers.
11. Implement 100 ANN classifiers.
12. Train each ANN classifier (i.e. network 1 to 100) independently of the others with each of the training sets produced in Step 8, to differentiate between successful and unsuccessful transplants.
13. Test each ANN classifier (i.e. network 1 to 100) independently of the others, using the validation set extracted in Steps 4 and 5, to differentiate between successful and unsuccessful transplants.
14. For each example, record the predicted output of each of the networks to produce the output of the ensemble.
15. Record the prediction accuracy, the amount of votes between classifiers and the data points that were used to produce such agreements. Optional: de-normalised the validation set using the extracted parameters from Step 2.
16. Analyse the results and predictive accuracy of classifiers.

The results are shown in Figure 4.5 (see also Table E.2 in Appendix E). In this experiment, using majority voting, the balanced validation set (with 438 cases) reached an accuracy rate of 61%. However, by using only 88 data points, the validation set reached a 71% accuracy rate (with a 93% agreement among the networks). As can be seen from Figure 4.5, the accuracy rate also reached approximately 75% when there was a 97% agreement among the networks, using 41 examples.

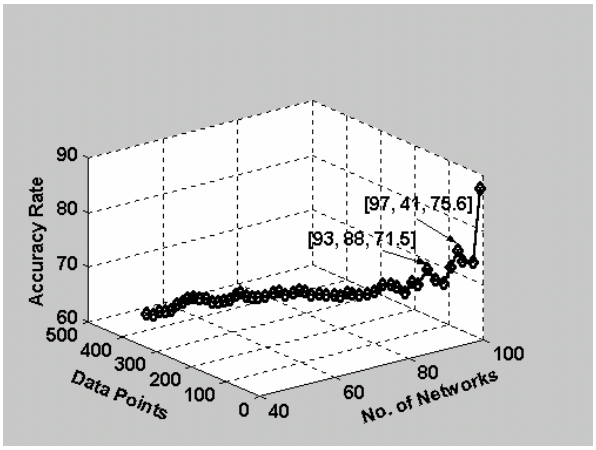

Figure 4.5: The results for 100-MLP using the second pre-processing strategy.

4.4.4. 100-MLP classifiers with all available success points and bagging

In this experiment it was decided to implement 100 MLP neural network classifiers using the same pre-processing scheme (as described in previous experiment) but this time with bagging technique to generate different training sets from the original training sets. For this experiment the following methodology was employed (see also Figure D.4 in Appendix D):

1. Pre-process the dataset. This includes decisions about the appropriate parameters to be included in the dataset, the time point and etc. This provides a final dataset of 657 fail points and 2371 success points.
2. Combine the fail and success datasets in a temporary file and extract normalisation parameters.
3. Perform the normalisation using the extracted normalisation parameters.
4. Randomly extract 1/3 of the data for the validation set from the file containing only **success** data points and keep the remaining 2/3 for training and testing.
5. Randomly extract 1/3 of the data for the validation set from the file containing only **fail** data points and keep the remaining 2/3 for training and testing.
6. For the file containing the remaining 2/3 of **success** data points, , randomly sample 219 records (randomly drawing without replacement)

for training and keep the remaining 219 records for testing (tuning) purpose.

7. For the file containing the remaining 2/3 of **fail** data points, randomly sample 219 records (randomly drawing without replacement) for training and keep the remaining 219 records for testing (tuning) purpose.
8. Combine the extracted 219 training success records from Step 6 with the extracted 219 training fail records from Step 7 to produce a training set.
9. Combine the extracted 219 testing success records from Step 6 with the extracted 219 testing fail records from Step 7 to produce a testing set.
10. Repeat Steps 6 to 9 to produce 100 training sets and test sets for ANN classifiers.
11. Implement 100 ANN classifiers.
12. **Use bagging to generate different training sets for the classifiers from the training sets generated in Step 10 in order to increase the diversity of ensembles.**
13. Train each ANN classifier (i.e. network 1 to 100) independently of the others with each of the training sets produced in Step 12, to differentiate between successful and unsuccessful transplants.
14. Test each ANN classifier (i.e. network 1 to 100) independently of the others, using the validation set extracted in Step 4 and 5, to differentiate between successful and unsuccessful transplants.
15. For each example, record the predicted output of each of the networks to produce the output of the ensemble.
16. Record the prediction accuracy, the amount of votes (agreements) between classifiers and the data point that were used to produce such agreements. Optional: de-normalise the validation set, using the extracted parameters from Step 2.
17. Analyse the results and predictive accuracy of classifiers.

The results are shown in Figure 4.6 (see also Table E.4 in Appendix E). In this experiment, using majority voting the balanced validation set (with 438 cases) reached accuracy rate of 60.5%. However, by using only 123 data points, the validation set reached an accuracy rate of 70.7 % (with a 83% agreement among the networks).

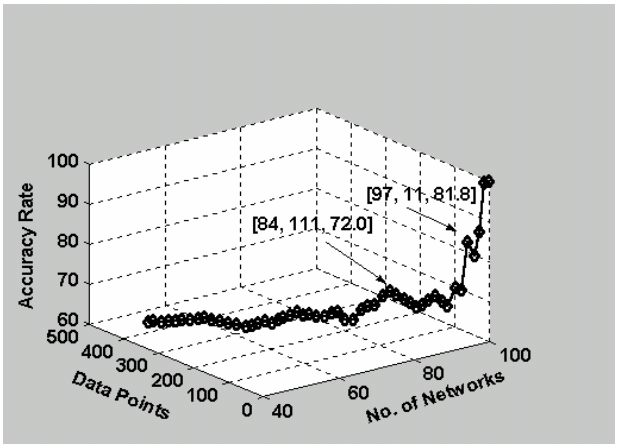

Figure 4.6: The results for 100-MLP using the second pre-processing strategy (with bagging).

4.4.5. 500-MLP classifiers with all available success points and bagging

In this experiment it was decided to use the same pre-processing scheme as described in previous experiment, however in Step 11, 500 MLP (instead of 100-MLP) classifiers were constructed (see also Figure 4.7).

This model required more training time. In this experiment, using majority voting the balanced validation set reached an accuracy rate of 60.5%. However, by using 19% of data points (84 cases); the validation set reached an accuracy rate of 70.2% (with a 87% agreement among the networks). This model was able to classify about 87% of successful transplants and 54% of unsuccessful cases. This shows that the majority of networks misclassified the unsuccessful transplants. This indicates the need for the user to examine the unsuccessful records to see whether they are mistakes that need to be fixed or if there is a need for more suitable input vectors or attributes. In this case, the investigation revealed that due to the poor quality of data, it was not possible to include in the prediction models all the variables that are known (at least theoretically) to be important factors influencing the outcome of unsuccessful kidney transplants. To improve the accuracy or the quality of the data, there are many other factors, such as the quality of post-transplant care or the cause of the original kidney failure that need to be recorded and perhaps be included in the prediction models for the kidney transplants (or for many other similar medical scenarios).

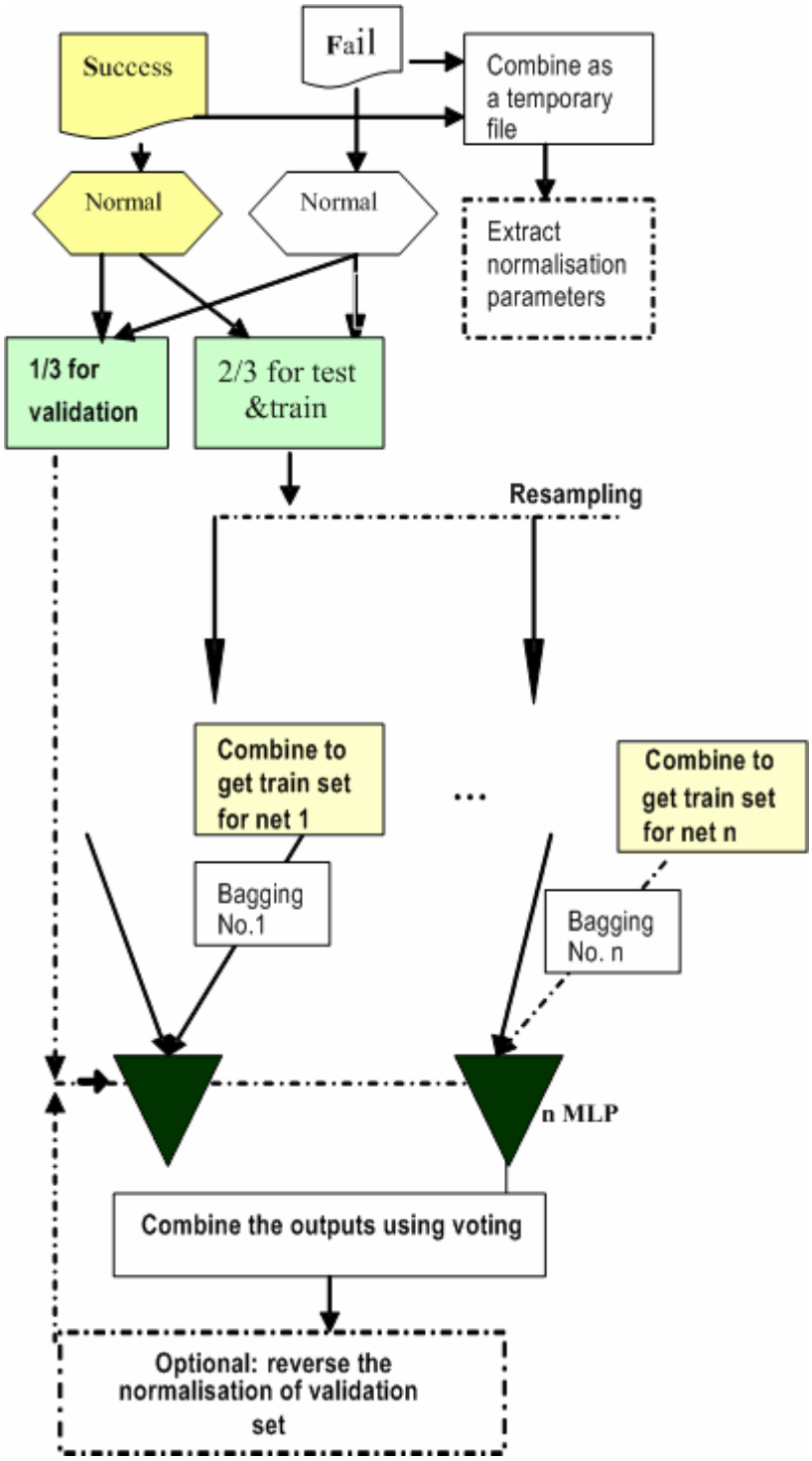

Figure 4.7: The RIDC-ANNE algorithm (where n or No. of Networks is 500)

As it can be seen from Figure 4.8, the accuracy rate also reached 76.1% with a 89% agreement among the networks (446 out of 500 networks), using 63 examples (see also Table E.5 in Appendix E).

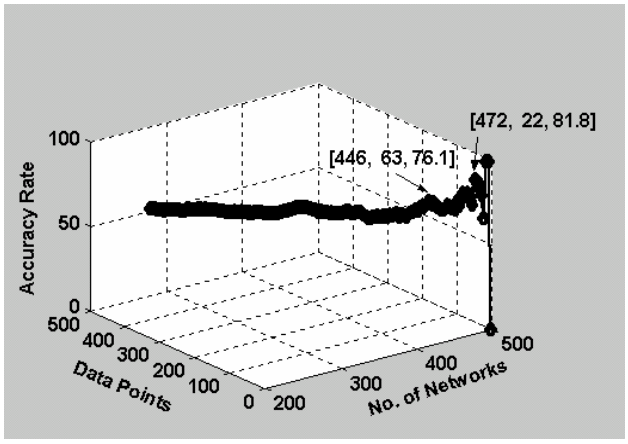

Figure 4.8: The results for 500-MLP using the second pre-processing strategy (with bagging).

Figure 4.9 demonstrates the rule generation stage of RIDC-ANNE. As can be seen in Figure 4.9, after the model generation and prediction stage we can generate a new training set. The set is generated by extracting all cases and their desired classes in a test set that have caused agreement on one category (fail or success) across at least x ($\frac{T}{2} \le x \le T$, where T = Total number of classifiers) classifiers in an ensemble model. Then replace the desired classes of all cases with their corresponding class labels (i.e. the class assigned by the trained ensemble). Finally, grow a decision tree (e.g. a C4.5 tree by using S-Plus software system) from samples and their new class labels in the chosen dataset.

It should be noted that for the rule generation process it is possible to use a more conventional strategy and generate a set of training data just by feeding the examples in the validation set to a trained ensemble and replacing the true class labels of the original test (validation) instances with the class labels assigned to them by the ensemble. However, in order to study different regions of the data space, it was decided to enforce the model to consider the examples whose class labels consistently caused agreement across the ANN classifiers. This strategy attempts to remove some of the branches of rules and to identify regions that have a strong impact on the system performance (this also means identifying the data spaces that have been consistently misclassified). This has the additional benefit of making the rule set substantially easier to understand.

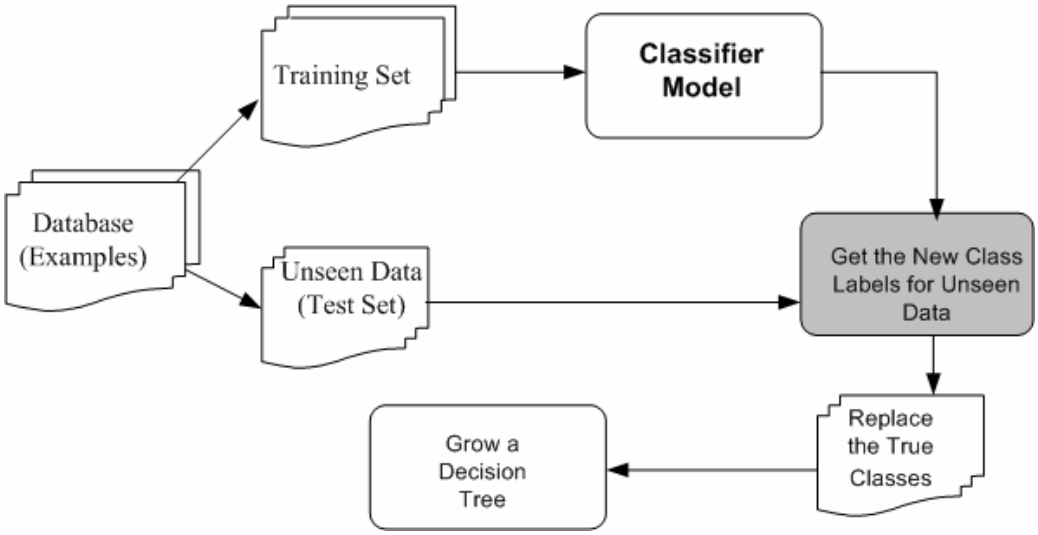

Figure 4.9: The back-end of RIDC-ANNE process

The following rule set was produced by applying the RIDC-ANNE approach (to kidney transplant data) based on the 84 examples whose class labels (outputs from the ensemble) were in agreement across 87% of classifiers:

1. donstate <= 4 AND donage <= 43: Success
2. donhosp <= 109 AND misa <= 1 AND refstate > 3 AND misb > 0 AND refhosp > 94: **Failure**
3. misa <= 1 AND misb <= 1: **Success**
4. donsex > 0 (i.e. Female): **Failure**
5. misa > 1: **Failure**
6. age > 27: **Success**

As demonstrated above, the rule set produced only six rules. These rules were valid for 97% of cases (82 cases) in the dataset. In the above rule set, the donor state and age were both considered to be important factors for the success of kidney transplants.

4.5. Results for UCI datasets using RIDC-ANNE

The investigation methodology of the kidney transplant data described in Section 5.3.9 was also applied to the “Pima Indian diabetes” and “The Wisconsin cancer” datasets. It was decided to choose the 100-MLP model, since it requires less training time. The

initial experiments revealed that for this group of datasets, increasing the number of networks did not significantly improve the system's performance.

In this experiment, using the Pima Indian diabetes data, which used the majority voting, the balanced test set reached above a 77% accuracy rate. In a related study, Newton Cheung (2001) reported around a 75% accuracy rate using the Pima Indian diabetes data (see also Table 4.2). Other group (Duch et al., 2001) implemented a neurofuzzy system and managed to achieve only a 73.8 % accuracy rate for this dataset.

Further investigation of the Pima Indian diabetes data revealed that the accuracy rate can reach an accuracy rate of 80% with a 68% agreement among the networks (68 of 100 networks), but only by using 93% (239 cases) of test data points. The results are shown in Figure 4.10 (see also Table E.6 in Appendix E). This model was successfully able to classify about 94% of patients who developed diabetes (i.e. those that will develop diabetes) and 48% of non-diabetes cases. It should be mentioned that the accuracy rate also reached 87% with a 99% agreement among the networks, using only 152 examples (59% of data points).

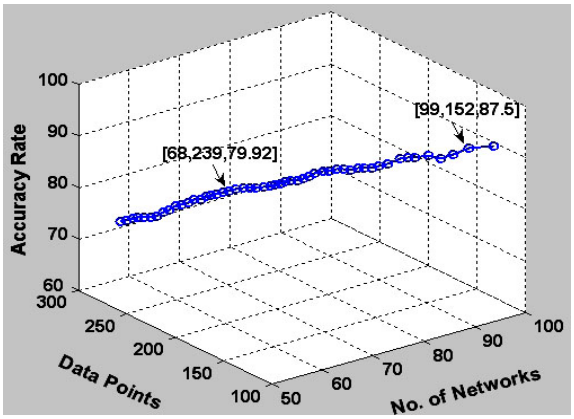

Figure 4.10: The results for the Pima Indian diabetes data with 100 bagging.

Moreover, the RIDC-ANNE approach can be used to generate a set of rules approximated by the trained ANNE to predict the diabetes and non-diabetes cases. The following rule set was produced by applying the RIDC-ANNE approach to classifying the diabetes and non-diabetes samples using the 239 examples that their class labels (outputs from the ensemble) were in agreement across 68% of classifiers:

1. 2-hour OGTT plasma glucose < 139.5 AND diabetes pedigree function < 0.76: **Diabetes**

2. 2-hour OGTT plasma glucose < 139.5 AND diabetes pedigree function > 0.76 AND Body Mass Index (BMI) < 36.5: **Diabetes**
3. 2-hour OGTT plasma glucose < 139.5 AND diabetes pedigree function > 0.76 AND Body Mass Index (BMI) > 36.5: **Non-Diabetes**

This means, for this selected dataset, the first two rule sets are roughly used by ANNE to predict diabetes cases with an accuracy rate of 94%. However, it should also be noted that for the healthy cases (those who did not develop diabetes) presented in the above samples, the ANNE consistently misclassified the samples (only a 48% accuracy rate). In this case 68% of classifiers made the same mistake and reported 52% of healthy cases as diabetes cases. Domain expert can use this information and rules to highlight and study the regions in the data space that have negative impacts on the generalisation ability of classifiers and more suitable input vectors may be supplemented accordingly.

Not surprisingly, for the Wisconsin Cancer dataset, using majority voting the balanced test set reached a 98% accuracy rate. This was slightly better than the results of a comparison study reported by Duch et al., (2001), who reported a 96.5% accuracy rate for their neurofuzzy system. As demonstrated in Table 4.3, a related study by Cheung (2001) reported an accuracy rate of 97% for his proposed Bayesian networks.

The model used in this study was able to classify approximately 98% of cancer cases and 97% of non-cancer cases. The accuracy rate also reached 98.2% with a 65% agreement among the networks (68 of 100 networks), based on 99.5% (227 cases) of test data points. The results also revealed that it is possible to achieve a 98.6% accuracy rate with a 100% agreement among the networks based on 96.5% (220 cases) of test data points. In other words, by removing only 0.5% (eight cases) of test data points, it was possible to generate 100% agreement among the networks and a higher level of accuracy. The complete results for the Wisconsin cancer dataset are shown in Figure 4.11 (see also Table E.7 in Appendix E).

Table 4.3 : A comparison table of sensitivity, specificity and accuracy

Data Base	Sensitivity	Specificity	Accuracy
RIDC-ANNE			
Breast Cancer, 50%-Net Agreement	97.97%	97.50%	97.81%
Breast Cancer, 65%-Net Agreement	97.97%	98.73%	98.24%
Breast Cancer, 100%-Net Agreement	98.64%	98.63%	98.64%
Pima Indians Diabetes, 50%-Net Agreement	47.73%	92.86%	77.34%
Pima Indians Diabetes, 68%-Net Agreement	51.25%	94.34%	79.92%
Pima Indians Diabetes, 99%-Net Agreement	48.39%	97.52%	87.50%
Kidney Transplant 500-Bagging (Section: 4.3.5), 50%-Net Agreement	57.99%	63.01%	60.50%
Kidney Transplant 500-Bagging, 87%-Net Agreement	54.55%	87.50%	70.24%
Kidney Transplant 500-Bagging, 89%-Net Agreement	54.84%	96.88%	76.19%
Bayesian Network with Naïve Dependence (BNND) (Cheung, 2001)			
Breast Cancer	98.53±1.05	96.38±0.24	97.13±0.50
Pima Indians Diabetes	60.63±1.31	83.14±1.86	75.29±1.01
Bayesian Network with Naïve dependence & Feature Selection (BNNF) (Cheung, 2001)			
Breast Cancer	98.49±0.68	96.19±0.42	97.00±0.36
Pima Indians Diabetes	60.52±1.42	83.64±0.96	75.57±0.86
C4.5 decision tree (BNND)(Cheung, 2001)			
Breast Cancer	-	-	85.65±1.82
Pima Indians Diabetes	-	-	75.13±1.52

The following rule set was produced by applying the RIDC-ANNE approach using the 220 examples whose class labels (outputs from the ensemble) were in agreement across 100% of classifiers:

1. Uniformity of Cell Shape < 3.5 AND Bland Chromatin > 4.5: **Cancer**
2. Uniformity of Cell Shape < 3.5 AND Bland Chromatin < 4.5: **Non-Cancer**
3. Uniformity of Cell Shape > 3.5: **Cancer**

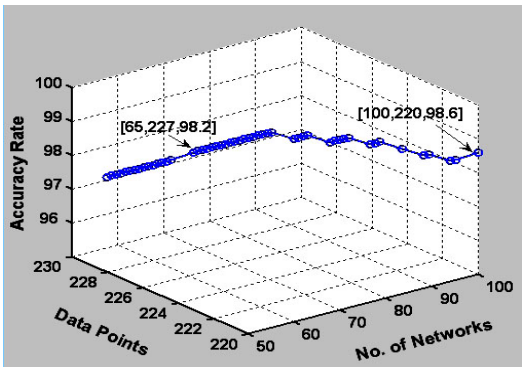

Figure 4.11: The results for the Wisconsin cancer dataset with 100 bagging.

4.6. Summary

This study began with an investigation of the use of the ANN technique in predicting the outcomes of kidney transplants. Then, to improve the prediction accuracy rate it was decided to use a novel neural networks ensemble technology, namely, RIDC-ANNE. This strategy focuses on altering the training process and combining the output of several classifiers. Furthermore, it identifies suspect data points and provides some clarity for the behaviour of classifiers.

Research shows that the neural network ensemble strategy offers good generalisation performance, even in the situations where a network is presented with incomplete data and needs to be trained to recognise small differences. However, the classification results of the network ensembles using the kidney transplant database, demonstrate that a significant improvement in classification accuracy can not be achieved over single Multilayer Perceptron (MLP) networks. This shows that clinical databases such as kidney transplant data, which contain a considerable amount of noisy and suspect data points, can significantly deteriorate the generalisation ability of prediction models. Furthermore, it is important to note that kidney transplant data do not contain the complete set of determinants, which affect the outcomes (e.g. Surgery condition, the quality of graft, some patient-specific data). This makes the predictions more difficult and emphasises the quality of predictions by RIDC-ANNE. These results were confirmed by using relatively cleaner data in the Pima Indian Diabetes and Wisconsin Cancer datasets.

This study revealed that networks generated a relatively low level of accuracy using all noisy or misleading examples in the validation set. However, by applying the RIDC-ANNE strategy and only using a selective subset of data points, it was possible to

improve the accuracy rate. For example, the results in Section 4.3.4 demonstrated that using 70% prediction accuracy as a benchmark, for 20% of records; RIDC-ANNE can achieve a higher accuracy rate. Having posed this idea, it allowed us to identify the patterns that were consistently misclassified across the classifiers. Moreover, the primary experimental results revealed that the rule generation process of the RIDC-ANNE approach can be used to generate a set of rules for each subset of data (generated from trained ANNE). This process can then be used to translate a neural network ensemble into an alternative, more understandable model. Overall, a system user or a domain expert may be able to use the extracted information and rules to identify unrealistic predictions and predict in which section of input vectors, or under which circumstances, the selected classifier may perform poorly.

Chapter 5

Conclusion and future work

Over the past few years there has been great interest in the development of robust and efficient data mining algorithms and a variety of approaches have been proposed. More recently, researchers in data mining and knowledge-based systems have turned to research on hybrid and integrated techniques. These newer algorithms and approaches have provided opportunities for researchers to carry out a variety of clinical trials and improve the earlier prediction techniques. However, their success is usually dependant on the quality of the data available. If the data is inadequate, or contains incomplete, irrelevant and misleading information, the data mining algorithms may produce less accurate and less transparent results, or even fail to discover any useful information. Therefore, the data pre-processing stage is one of the most important factors that can affect the success of data mining on a given task. Data pre-processing usually includes data cleaning, normalisation, transformation, feature extraction, data filtering, and so on. Successful data pre-processing not only leads to better quality results but it can also reduce the cost of data mining.

The work reported in this thesis started with an investigation into the potential of ANN models in application to clinical data. The first investigation undertaken in this thesis was the use of a single ANN classifier for kidney transplant outcome prediction where the main goal was to predict whether a kidney transplant was likely to be a success or a failure. Currently, medical professionals around the world use blood types, the number of Human Leukocyte Antigen (HLA) mismatches and the basic cross-match tests[6] to determinate the best match between kidney donors and recipients. However, the current matching and testing scheme do not always provide a successful outcome.

[6] The HLA is the general name of a group of genes in human cells that enable our body to tell the difference between our own cells and foreign ones. The cross-match test involves a mixing of cells and serum to determine whether or not the recipient of a kidney will respond to the transplanted organ by attempting to reject it

Literature search revealed the chance of a successful cadaver transplant surviving for one year after the transplant is about 85% and 50% at five years[7].

The initial results of this work revealed that despite using a range of standard pre-processing strategies for prediction of kidney transplant outcomes, the resultant accuracy was too low to be of clinical use. It should be noted that in a related study, Michael et al (2003) reported success when compared ANN with traditional LR models for the prediction of delayed graft function in cadaveric renal transplants. However, their resultant accuracy was still a low 64%. This could be due to the inherent complexity contained in the kidney transplant data.

Due to the successful track record of Artificial Neural Network Ensembles in application to difficult and complex problem, we decided to investigate the possibility of using neural network ensembles for complex and noisy clinical data by addressing the following fundamental question: What is the potentiality of an ensemble of neural networks models as a filter and classifier in a complex clinical situation?

To answer this question, a set of networks were trained independently to differentiate between successful and unsuccessful kidney transplant data. In addition, as part of a major investigation, it was decided to consider the problem of dealing with complex clinical data and rule extraction from individual networks by studying the data and designing a novel hybrid-learning model called RIDC-ANNE.

This novel algorithm was successfully developed with reference to the data filtering, bagging-based ensemble and the hybrid decision tree-neural networks ensemble theories described in Chapter 2. The RIDC-ANNE approach assists the data-preparation process by configuring an ensemble of bagged networks as a filter and identifying the regions in the data space that have been consistently misclassified or have high impact on the system performance. The RIDC-ANNE technique treats the ensembles as a black box or data filtering tool in order to provide a body of new training examples to feed standard rule induction or decision tree systems. This strategy considers the diversity and expertise of the component networks in the rule generation process. However, unlike other rule generation techniques that have been described in Section 2.5.2, it does not focus on the identifying ensemble members that are relevant in

[7] Transplant Service of the University of Texas Medical Branch at: ttp://www.utmb.edu/renaltx/srate.htm (last accessed: 30 November 06)

explaining the prediction (output) associated with a particular case. Instead RIDC-ANNE explains the output of the ensemble based on a cluster of cases that consistently generate agreement across the classifiers with similar expertise. The use of RIDC-ANNE not only produces rule sets that are substantially easier to understand but it also reduces the computational cost, since it creates a cluster of new training examples that allow faster rule generation process.

The RIDC-ANNE approach uses the Bagging algorithm to train neural networks and extracts the input patterns (examples) that were included across the ANN series in the final results. In this strategy (see also Section 3.3) for each of the patterns or examples in the validation set, the votes were recorded first. Then the system searched for patterns or examples that caused at least 50% of the networks to agree on one category (fail or success). Then, it searched for patterns that caused 51% agreement among the networks. This procedure was repeated until the model reached the 100% agreement among the networks for one category (fail or success). It should be noted that as the model approaches the 100% agreement condition, fewer patterns were able to satisfy the condition and be included in the report. Finally, for each condition (i.e. network agreement), the prediction accuracy and the number of examples that satisfy the condition were reported by the system.

This investigation revealed that ANN models generated relatively low level of predictive accuracy using all the examples. However, by using only a selective subset of data points, it was possible to improve the accuracy rate. For example, an impartial investigation into the potential of the RIDC-ANNE strategy when applied to kidney transplant data (see also Section 4.3.10.) revealed that when using the majority voting, the balanced validation set (almost the same amount of negative and positive cases) can reach an accuracy rate of 60.5%. However, by using 19% of data points (84 cases), the validation set can also reach a 70.2% accuracy rate (with 87% agreement among the networks). Furthermore, a close study of the rule set generated in this scenario revealed that the donor state and age were considered two important factors in determining the success of kidney transplants.

Among the networks (446 of 500 networks) the accuracy rate also reached 76.1% with 89% agreement, using only 63 examples (14% of data points).

Furthermore, this study investigated the potential of the RIDC-ANNE strategy in the application of two well-known medical datasets, Pima Indian Diabetes and

Wisconsin Cancer (Jayadeva, Khemchandani and Chandra) and compared our results with the results obtained by other researchers whenever possible. For the Pima Indian Diabetes data, using the majority voting, the balanced test set reached around a 77% accuracy rate. As demonstrated in Section 4.4, this result was slightly better than previous comparison studies (Cheung, 2001; Duch et al., 2001). However, by using only 93% (239 cases) of data points, it was possible to achieve an 80% accuracy rate with a 68% agreement among the networks. The accuracy rate also reached 87% with a 99% agreement among the networks, when only 152 examples (59% of data points) were used. Not surprisingly, by removing only 0.5% of the Wisconsin breast cancer (which is known to be a well recorded and non-complex dataset) data points, it was possible to generate 100% agreement among the networks and achieve an accuracy rate of 98.6%.

Recently, the front end of the RIDC-ANNE technique has been successfully tested by Szukalski et al. (2005) on a set of artificial data[8]. The results of this study revealed that the RIDC-ANNE approach has a great potential for the classification tasks and can successfully be used to extract the subsets of data that in effect have a higher signal to noise ratio than the original data.

For the purposes of this study it was further decided to compare the study findings with those results obtained by other researchers (for accuracy) using the same datasets whenever possible. The results in Table 4.3 suggest that overall, the proposed RIDC-ANNE approach performs better than the previous studies (Cheung, 2001; Duch et al., 2001).

In the rule generation stage of RIDC-ANNE approach, it is important to note that in general, different rule structures may be extracted for different data partitions (Duch et al., 2001). Therefore, even for the same databases, it was not always possible to make an equitable comparison of the rules which have been generated for different regions of the data space in this study and the rules obtained by other researchers. In addition, it was found that there is a need for further research into designing more efficient data pre-processing techniques that can guarantee the preservation of more data.

Although ANN models are known to be remarkably robust and forgiving to the presence of missing values in the training set (even when the missing values are simply replaced by zeros), however, the kidney transplant analysis demonstrated in Sections

[8] The artificial datasets were generated in order to know exactly which data has information content and which contains noise.

4.3.2 and 4.3.3, showed that it was necessary to remove records containing unsuccessful transplants in order to avoid a skewed distribution of missing data. Also, the records of those patients who were lost in the follow up were removed. Instead, the records where the information (for important variables) was available for both successful and unsuccessful transplants were selected. As a result, as demonstrated in Table 4.3 in most cases, the proposed RIDC-ANNE approach performed differently in terms of sensitivity and specificity (e.g. 54.84% versus 96.88% for Kidney transplant data with a 89% agreement among the networks). The study showed that the RIDC-ANNE was able to predict the desirable cases (e.g. successful transplants). This shows that the majority of networks misclassified the unsuccessful transplants. This is an indication of the need for users or domain experts to examine unsuccessful records further to see whether they are mistakes that need to be fixed or if there is a need for more suitable input vectors or attributes. In this case the investigation revealed that due to poor quality of data, it was not possible to include all the variables that are known (at least theoretically) to be important factors influencing the outcome (particularly unsuccessful outcomes) of kidney transplants in the prediction models. There are many factors such as quality of post transplant care or the cause of original kidney failure that need to be recorded and perhaps be included for the kidney transplants (or for many other similar medical scenarios) in the prediction models. These results show that the data must be carefully collected to ensure completeness. However, in routine practice, this is not always possible and some information is typically missing in clinical databases. One implication of this study is that more attention needs to be given to both data collection and preparation stages.

In summary, the results (as shown in Sections 4.4 and 4.5) showed that the application of the RIDC-ANNE model in predicting outcomes for a complex dataset such as kidney transplantation is promising. The method presented in this thesis proved useful in explaining the kidney transplant data and also provides basis for improved methods for matching the donor-recipient pairs with higher probability of successful transplant. The model has been ordered by the steps of KDD process as defined by (Fayyad et al., 1996) and designed to:

1. Provide a strategy to deal with complex clinical data gathered from patients with diabetes and kidney diseases or patients who have undergone kidney transplant surgery.
2. Deal with the 'black boxes' issue of connectionist learning systems such as ANN.
3. Construct useful decision trees from real-world data.
4. Translate a neural network system into an alternative, more understandable model.
5. Demonstrate how coupling ANN and rule extraction algorithms can improve the learning performance of knowledge acquisition tools, enhance the overall utility of ANN and reduce the brittleness of rule-based systems.
6. Provide a method of extracting reliable rules from an ensemble. An ensemble of bagged networks can indeed be successfully configured to act as a filter using the RIDC-ANNE method.
7. Translate a neural network ensemble into an alternative, more understandable model.
8. Produce rule sets that are substantially easier to understand but it also reduces the computational cost, since it creates a cluster of new training examples that allow faster rule generation process.
9. Show that the data must be carefully collected to ensure completeness.

A system user or a domain expert may be able to use the extracted information and rules to:

- Identify unrealistic predictions and pinpoint the source by studying different regions of the data space and their corresponding rule sets.
- Identify data points that have been consistently misclassified. These points can provide valuable insight into particular observations. The user may examine these data points to see whether they are mistakes that need to be fixed, unusual circumstances that differ dramatically from the study objectives, or they are just unusual cases that must be discussed further.
- Highlight the regions in the data space that have negative impacts on the generalisation ability of predictive models and supplement more suitable input vectors. For example, for neural networks classifiers the idea is to come up with

a model that predicts future observations well, therefore in some situations it might be necessary to delete unusual cases or circumstances in order to gain improvements in out-of-sample prediction accuracy.

- Predict in which section of input vectors or under which circumstances the neural network model may perform poorly.
- Shed light on the overall quality of the data under investigation and decide what actions need to be taken next. For example, in circumstances where obtaining new data may take days or be relatively expensive, it is wise to invest in a more reliable tool or even chose techniques with a higher computation time, to ensure that those examples are as useful as possible and the predictions are reasonable. In other situations, however, obtaining new data may be inexpensive and hence this option may be the wiser choice.

The research conducted in this thesis advances current clinical data preparation and classification techniques where the goal is to extract patterns that contain higher information content from a sea of noisy and incomplete clinical data, and building accurate and transparent classifiers. This new approach shows great promise for clinical decision making systems and can provide a valuable data mining tool with both research and commercial potential.

The research has raised several issues for future work. The major future extensions identified include the following:

1. Designing a more efficient data pre-processing technique (particularly for the rule extraction process) that can guarantee the preservation of more data and studying the value of extracted information and rules in details with the help of a domain expert. This can help to gain more information on the factors that have had some influence on the outcome of transplants. This step is important since it is important to have a dataset at hand that contains sufficient relevant characteristics to perform rule extraction and the classification of graft outcome prediction with a high level of accuracy.
2. The possibility of modification of the RIDC-ANNE strategy by using a combination of both neural networks and fuzzy logic technologies in which membership for classes of cases is a matter of degree and not necessarily 0 or 1 (Bellman and Zadeh, 1970; Zadeh, 1983).

3. Developing a fully automated tool that can be used to generate predictions for kidney transplant procedures by using the proposed hybrid classification scheme. It will be necessary to train and validate the tool on more complete data that has been recently collected from relevant clinical environments.
4. Employing a N-fold cross-validation scheme. It would have been interesting to employ, for example, a 10-fold cross-validation with some of the smaller datasets used in the present study. As described in Section 2.3.3, in N-fold cross-validation, the available data are separated into N parts. Then a classification model is constructed from N-1 of the subsets and tested on the one withheld subset to obtain an unbiased estimate of performance. This process is repeated ten times, each with a different subset withheld. Accuracy across the N parts is averaged to yield an overall estimate of likely future performance.

APPENDICES

Appendix A

List of publications

This appendix displays the list of refereed publications and book chapters that have been achieved to date from the thesis work and during this period of study.

[1] Shadabi, F., Sharma, D., Cox, R. (2006), "Learning from Ensembles: Using Artificial Neural Network Ensemble for Medical Outcomes Prediction", *Proceedings of the 3rd International Conference on Innovations in Information Technology (IIT 06),* Dubai, UAE - The full text access will also be provided by IEEE Explore: 103.

[2] Shadabi, F., Sharma, D., Cox, R., and Petrovsky, N. (2006), "Data Extraction for Improved Prediction Outcomes in Organ Transplantation." *Proceedings of the International Statistics Workshop, Contributions to Probability and Statistics-Application and Challenge*s: 276-288.

[3] Shadabi, F., Cox, R., Sharma, D., and Petrovsky, N. (2005), "A Hybrid Decision Tree - Artificial Neural Networks Ensemble Approach for Kidney Transplantation Outcomes Prediction." *Proceedings of the 9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 05),* Melbourne, Australia 2: 116-22.

[4] Shadabi, F., Cox, R., Sharma, D., and Petrovsky, N. (2004), "Use of Artificial Neural Networks in the Prediction of Kidney Transplant Outcomes." *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES 04),* Wellington, New Zealand 3: 566-572.

[5] Shadabi, F., Cox, R., Sharma, D., and Petrovsky, N. (2004), "Experiments with a Neural Network Ensemble to Predict Renal Transplantation Outcomes." *Proceedings of the 2nd International Conferences on Artificial Intelligence in Science and Technology (AISAT 04),* Hobart, Australia: 271-76.

[6] Shadabi, F. and Khodai-Joopari, M. (2004), "An Investigation of Prediction Techniques in Management Application." *Proceedings of the 2nd International Conferences on Artificial Intelligence in Science and Technology (AISAT 04),* Hobart, Australia: 47-50.

[7] Shadabi, F. and Khodai-Joopari, M. (2003), "Case Based Reasoning Models in Management Application. " *Proceeding of the 8th Australian and New Zealand Intelligent Information Systems Conference, (ANZIIS03),* Sydney, Australia**:** 457-62.

[8] Shadabi, F. and Sharma, D. (2007) "The Use of Data Mining Techniques in Clinical Decision Making Systems" book chapter in: *Success in Evolutionary Computation*, Ang Yang and Yin Shan (Eds), UNSW, Springer publishing.

Appendix B

Descriptive statistics of datasets

This appendix describes the descriptive statistics of the main input variables used from the Kidney Transplants Database, the Pima Indians diabetes dataset and the Wisconsin Breast Cancer dataset.

Table B.1: Descriptive statistics for the main input variables used from Kidney Transplants database

Variables	N	Missing	Minimum	Maximum	Mean	Std. Deviation
age	3028	0.00	1.00	73.00	41.60	14.90
misa	3028	0.00	0.00	2.00	0.99	0.68
misb	3028	0.00	0.00	2.00	1.06	0.67
misd	3028	1.00	0.00	2.00	0.85	3.16
misdq	3028	0.00	0.00	2.00	0.48	0.58
refhosp	3028	0.00	5.00	309.00	86.91	64.68
refstate	3028	0.00	0.00	8.00	4.43	2.15
donhosp	3028	0.00	1.00	319.00	82.51	65.70
donstate	3028	0.00	0.00	8.00	4.39	2.13
tranhosp	3028	0.00	5.00	307.00	83.85	57.52
transtate	3028	0.00	0.00	8.00	4.39	2.08
donsource	3028	0.00	1.00	14.00	1.82	2.17
donage	3026	1.00	1.00	76.00	36.16	15.94
donesex	3028	0.00	0.00	1.00	0.42	0.49
ischemia	2859	95.00	0.00	45.00	13.55	7.99
kidpresI	2458	315.00	1.00	23.00	6.56	6.08

Table B.2: Descriptive statistics for the input variables used from the Pima Indian Diabetes dataset

Variables	N	Missing	Minimum	Maximum	Mean	Std. Deviation
n.preg	768	0.00	1.00	17.00	3.84	3.36
xhrOGTT.plasm	768	0.00	0.00	199.00	120.85	31.97
blood.pressure	768	0.00	0.00	122.00	69.10	19.35
skin.fold	768	0.00	0.00	99.00	20.53	15.95
xhrSer	768	0.00	0.00	846.00	79.79	115.24
bmi	768	0.00	0.00	67.10	31.99	7.88
diabet.sped	768	0.00	0.07	2.47	0.47	0.33
age	768	0.00	21.00	81.00	33.24	11.76

Table B.3: Descriptive statistics for the input variables used from the Wisconsin Breast Cancer dataset

Variables	**N**	Missing	Minimum	Maximum	**Mean**	Std. Deviation
Clump Thickness	683	0.00	1.00	10.00	4.44	2.82
Uniformity of Cell Size	683	0.00	1.00	10.00	3.15	3.06
Uniformity of Cell shape	683	0.00	1.00	10.00	3.21	2.98
Marginal Adhesion	683	0.00	1.00	10.00	2.83	2.86
Single Epithelial Cell Size	683	0.00	1.00	10.00	3.23	2.22
Bare Nuclei	683	0.00	1.00	10.00	3.54	3.64
Bland Chromatin	683	0.00	1.00	10.00	3.44	2.44
Normal Nucleoli	683	0.00	1.00	10.00	2.86	3.05
Mitoses	683	0.00	1.00	10.00	1.60	1.73

Appendix C

A sample of kidney transplant data

This appendix displays a small sample of Kidney Transplant data.

```
1. age
2. misa
3. misb
4. misd
5. misdq
6. refhosp (codes)
7. refstate (codes)
8. donhosp (codes)
9. donstate (codes)
10. tranhosp (codes)
11. transtate (codes)
12. donsource (codes)
13. donage
14. donsex (0=male, 1=female)
15. ischemia (hrs)
16. kidpresi
17. target (1, 2)

Data:
37 1 1 1 1 187 3 187 3 187 3 4 58 1 1 1 2
11 0 1 1 0 5 5 5 5 5 5 4 35 1 1 2 2
27 2 1 0 0 75 3 83 7 75 3 1 29 1 17 9 2
56 1 0 1 2 10 3 78 5 10 3 1 23 0 15 2 2
59 1 2 2 1 65 5 78 5 65 5 1 50 0 30 9 2
62 1 1 1 1 80 4 78 5 72 4 1 25 1 10 7 2
43 1 0 1 0 91 2 108 2 69 2 1 34 1 17 7 2
56 2 1 0 0 9 8 9 8 9 8 1 51 0 12 2 2
27 1 1 1 0 74 6 74 6 74 6 1 36 1 ? 2 2
56 1 0 2 2 307 6 74 6 307 6 1 61 1 16 2 2
44 1 2 1 1 307 6 74 6 307 6 1 46 0 7 9 1
41 1 1 1 0 72 4 72 4 72 4 1 42 0 8 3 1
21 2 2 2 0 75 3 75 3 75 3 14 60 1 1 1 1
43 1 1 1 0 74 6 307 6 74 6 1 55 1 15 2 1
44 2 2 0 0 69 2 89 2 69 2 1 52 0 23 2 1
57 0 1 1 1 107 8 107 8 107 8 9 29 0 1 ? 1
50 1 2 0 0 107 8 107 8 107 8 1 34 1 13 ? 1
14 0 1 0 0 112 8 9 8 9 8 5 51 0 1 ? 1
25 1 1 1 1 84 3 84 3 84 3 5 53 0 1 ? 1
57 1 1 0 0 112 8 88 8 112 8 1 21 0 12 2 1
30 1 1 0 0 107 8 107 8 107 8 4 65 1 1 ? 1
51 2 2 2 1 112 8 112 8 112 8 2 46 1 1 5 1
60 1 1 1 1 72 4 72 4 72 4 1 59 1 14 ? 1
38 2 2 1 1 26 3 13 3 75 3 1 30 1 14 ? 1
35 1 1 2 1 107 8 58 8 107 8 1 10 0 32 2 1
16 1 0 1 1 89 2 27 5 89 2 1 12 1 ? 2 1
58 2 1 2 2 72 4 250 4 72 4 1 72 0 20 ? 1
29 0 0 1 1 112 8 30 8 112 8 1 50 0 ? 2 1
26 2 2 2 1 101 4 14 1 72 4 1 25 0 22 2 1
```

Appendix D

Graphic demonstrations of RIDC-ANNE

This section shows a graphic representations of the front end of the RIDC-ANNE technique that have been implemented with slightly different data pre-processing and training schemes, as described in Sections 4.3.1 to 4.3.4.

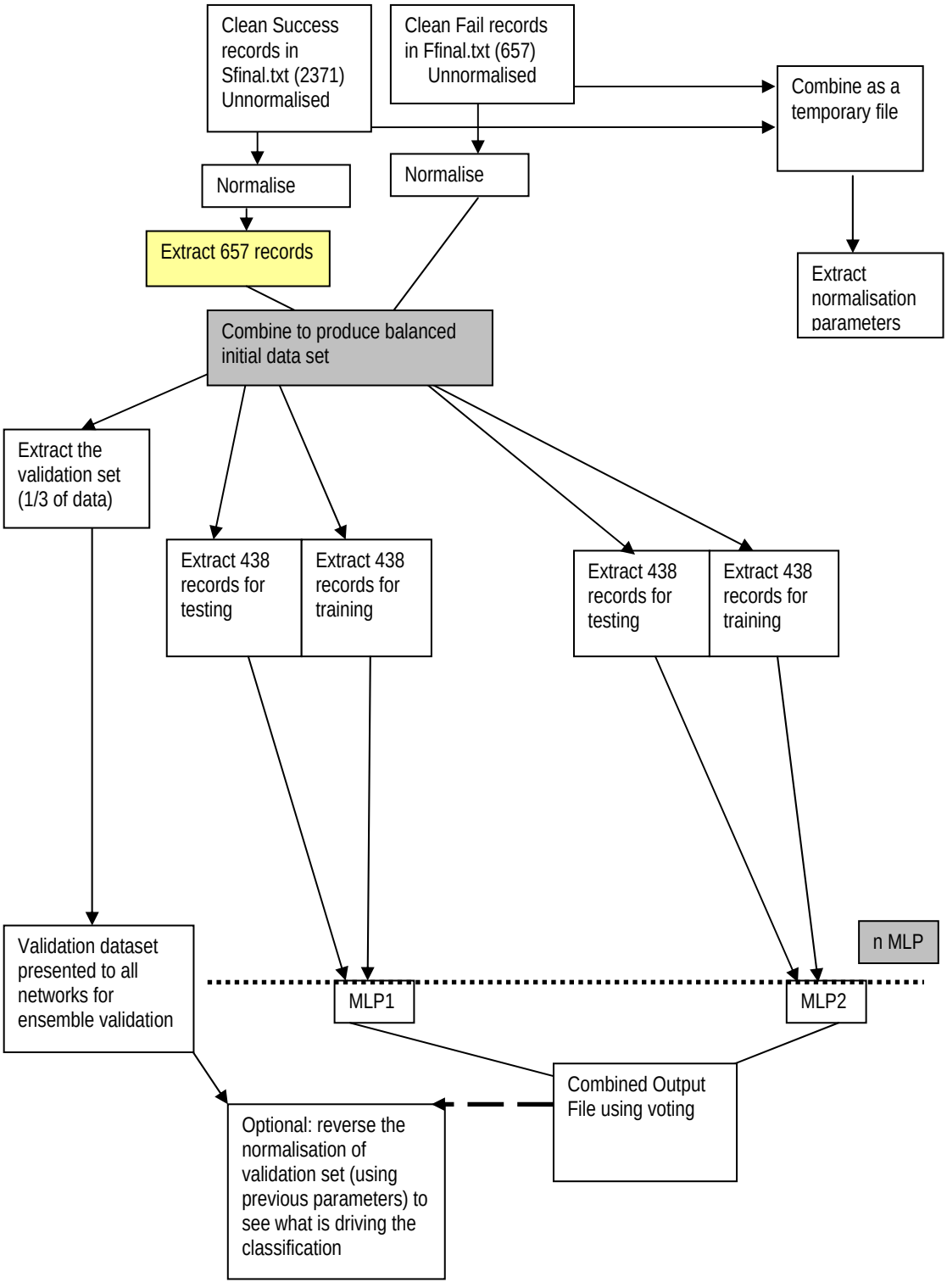

Figure D. 1: 100- MLP classifiers without bagging (Exp 4.3.1)

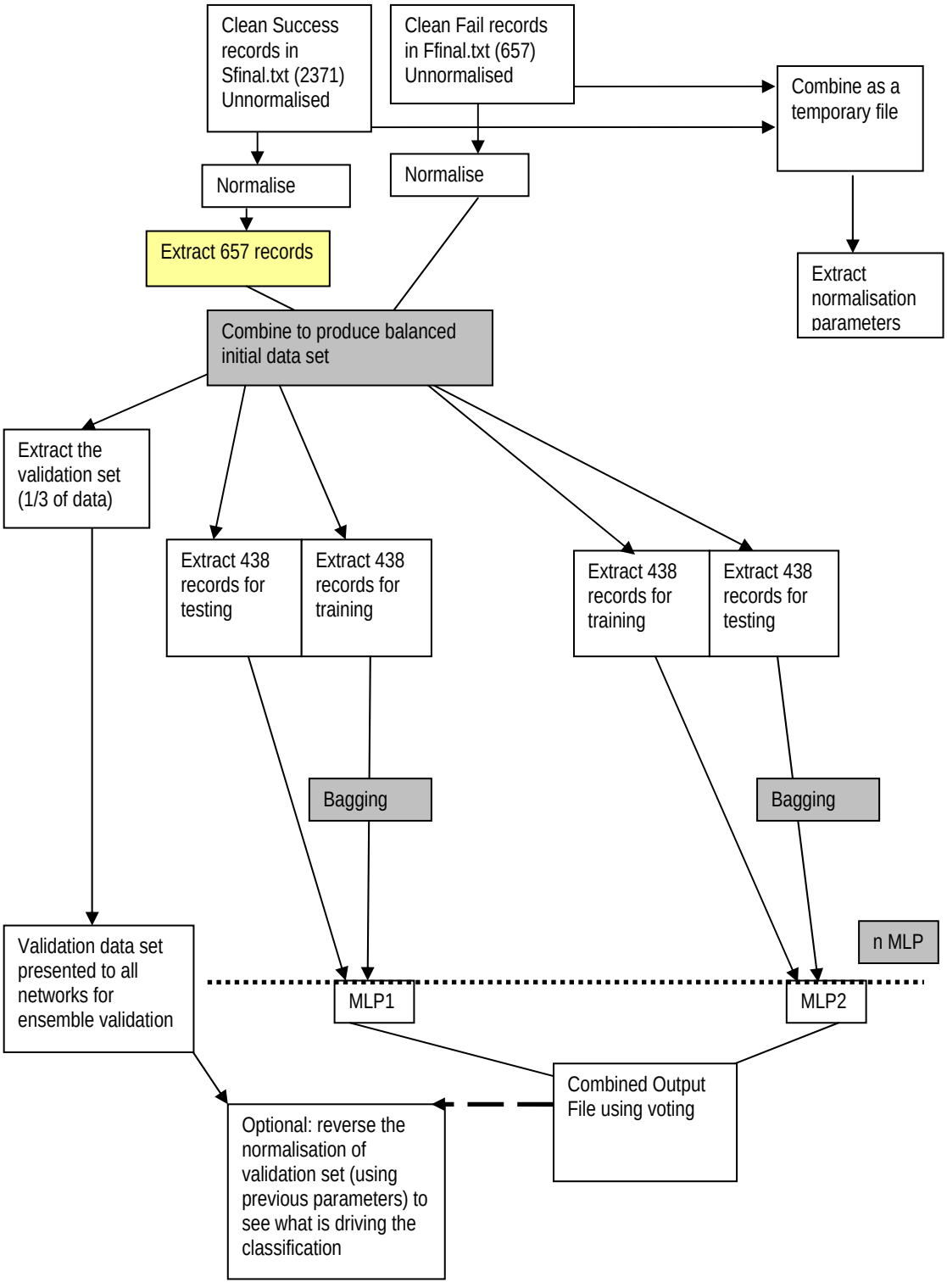

Figure D.2: 100- MLP classifiers with bagging (Exp 4.3.2)

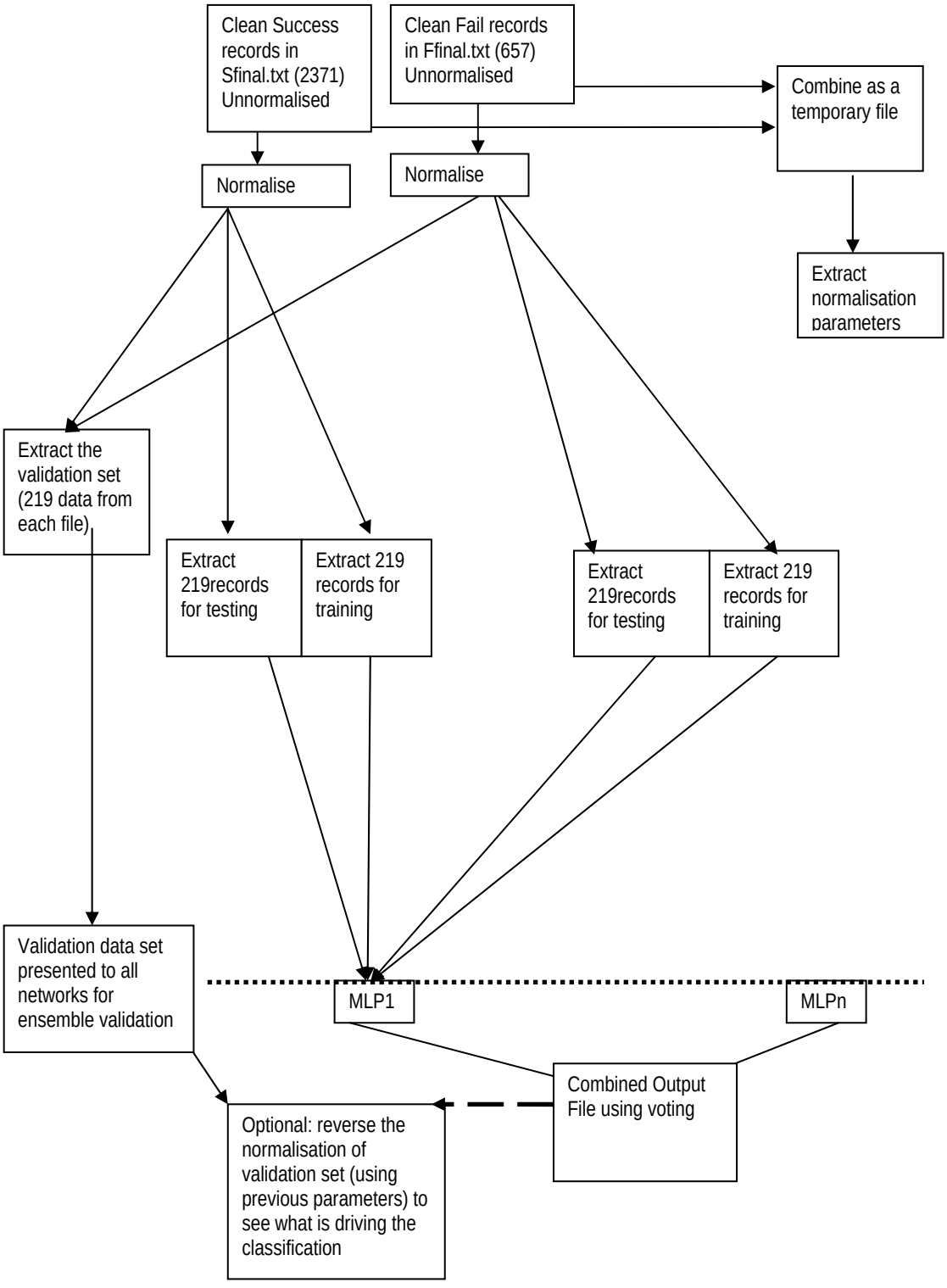

Figure D.3: 100- MLP classifiers with all success data & without bagging (Exp 4.3.3)

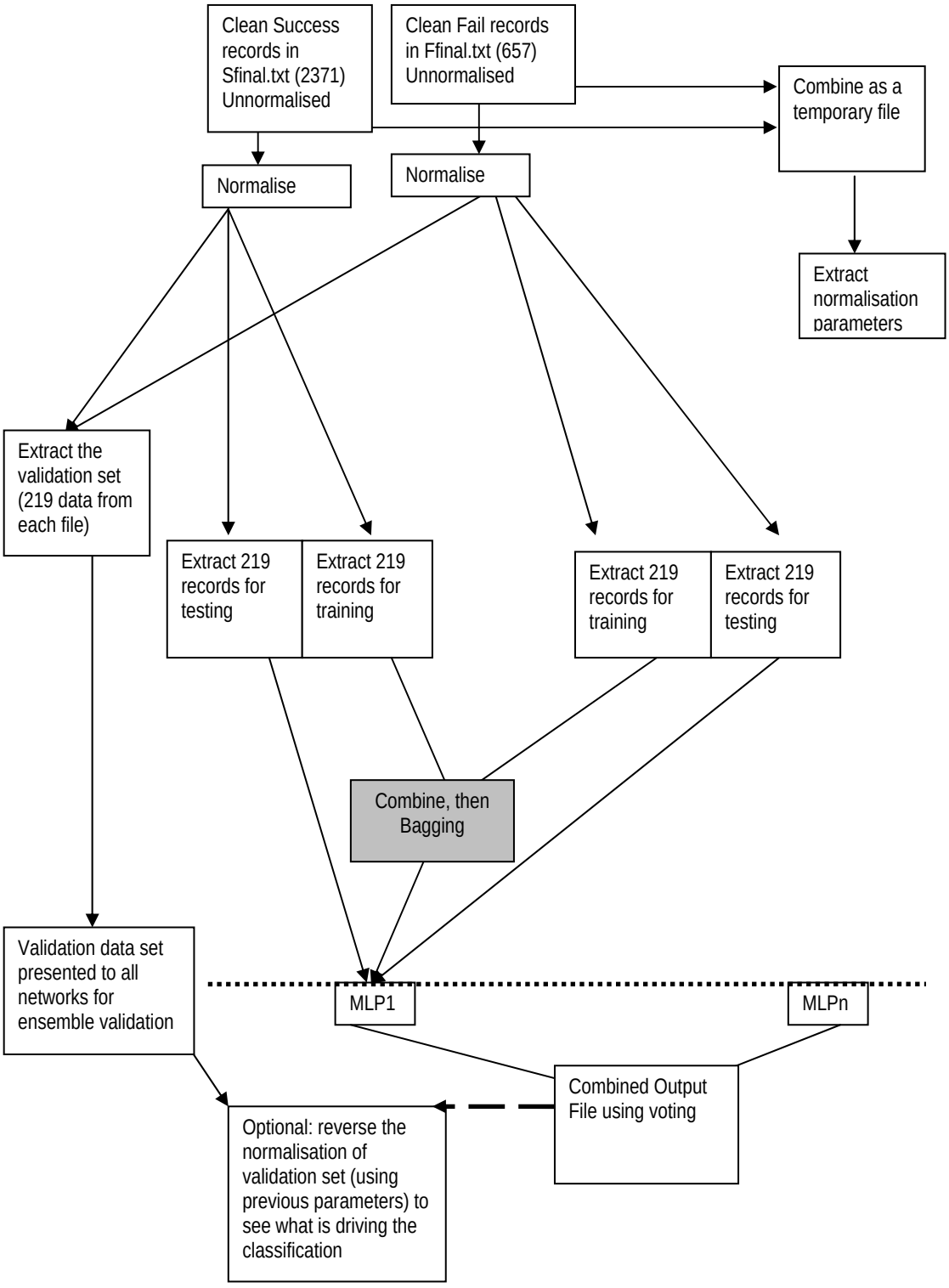

Figure D.4: 100- MLP classifiers with all success data & with bagging (Exp 4.3.4)

Appendix E

Detailed experimental results

This study describes the empirical study of a novel neural networks ensemble technology, named Rules and Information Driven by Consistency in Artificial Neural Networks Ensemble (RIDC-ANNE) for the purposes of predicting the outcome of medical events. This approach attempts to improve data quality by configuring an ensemble of bagged networks as a filter and identifying the regions in the data space that have a high impact on the system performance.

This section demonstrates the results in detail by showing the percentage of votes (agreements) between classifiers, the prediction accuracy and the number of data points that were used to produce such agreements for the kidney transplant, Pima Indian Diabetes and Wisconsin Cancer datasets.

Table E.1: The results for 100-MLP classifiers without bagging- Kidney Transplant

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	61.42	438	76	65.77	260
51	61.31	429	77	65.22	253
52	61.85	422	78	65.69	239
53	61.87	417	79	65.5	229
54	62.2	410	80	65.6	218
55	63.28	403	81	65.55	209
56	63.57	398	82	66.17	201
57	64.05	395	83	65.82	196
58	64.43	388	84	65.78	187
59	64.23	383	85	66.09	174
60	64.17	374	86	66.67	165
61	63.56	365	87	68.21	151
62	63.51	359	88	68.35	139
63	63.66	355	89	67.94	131
64	64.08	348	90	66.95	118
65	64.81	341	91	69.09	110
66	65.28	337	92	68.63	102
67	64.65	331	93	71.59	88
68	64.42	326	94	69.86	73
69	64.38	320	95	69.49	59
70	64.52	310	96	72.55	51
71	65.44	298	97	75.61	41
72	65.74	289	98	73.68	38
73	65.02	283	99	73.68	19
74	65.68	271	100	87.5	8
75	66.04	265			

Table E.2: The results for 100-MLP classifiers with bagging- Kidney Transplant

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	57.76	438	76	61.85	173
51	57.64	432	77	63.8	163
52	58.07	415	78	63.58	151
53	58.66	404	79	61.87	139
54	58.88	394	80	63.85	130
55	59.84	381	81	64.46	121
56	59.36	374	82	64.76	105
57	59.45	365	83	64.29	98
58	59.83	361	84	64.13	92
59	59.77	353	85	66.25	80
60	60.06	343	86	65.75	73
61	60.24	332	87	65.08	63
62	60.63	320	88	71.7	53
63	60.65	310	89	72.92	48
64	60.6	302	90	68.29	41
65	60.21	289	91	68.42	38
66	60.07	283	92	67.65	34
67	60.07	273	93	64	25
68	60.15	266	94	63.16	19
69	61.9	252	95	61.11	18
70	61.6	237	96	57.14	14
71	61.84	228	97	70	10
72	61.93	218	98	66.67	6
73	60.87	207	99	75	4
74	60.8	199	100	50	2
75	61.38	189			

Table E.3: The results for 100-MLP classifiers with all available success points without bagging-Kidney Transplant

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	61.42	438	76	65.77	260
51	61.31	429	77	65.22	253
52	61.85	422	78	65.69	239
53	61.87	417	79	65.5	229
54	62.2	410	80	65.6	218
55	63.28	403	81	65.55	209
56	63.57	398	82	66.17	201
57	64.05	395	83	65.82	196
58	64.43	388	84	65.78	187
59	64.23	383	85	66.09	174
60	64.17	374	86	66.67	165
61	63.56	365	87	68.21	151
62	63.51	359	88	68.35	139
63	63.66	355	89	67.94	131
64	64.08	348	90	66.95	118
65	64.81	341	91	69.09	110
66	65.28	337	92	68.63	102
67	64.65	331	93	71.59	88
68	64.42	326	94	69.86	73
69	64.38	320	95	69.49	59
70	64.52	310	96	72.55	51
71	65.44	298	97	75.61	41
72	65.74	289	98	73.68	38
73	65.02	283	99	73.68	19
74	65.68	271	100	87.5	8
75	66.04	265			

Table E.4: The results for 100-MLP classifiers with all available success points and bagging- Kidney Transplant

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	60.5	438	76	65.46	194
51	60.46	435	77	65.78	187
52	60.38	424	78	64.04	178
53	60.87	414	79	64.42	163
54	60.99	405	80	67.11	152
55	61.36	396	81	68.06	144
56	61.46	384	82	68.38	136
57	62.1	372	83	70.73	123
58	62.47	365	84	72.07	111
59	61.97	355	85	71.43	105
60	61.99	342	86	70.21	94
61	61.59	328	87	69.32	88
62	61.39	316	88	68.29	82
63	61.33	300	89	68.92	74
64	61.64	292	90	70.15	67
65	62.11	285	91	71.19	59
66	62.82	277	92	70	50
67	62.31	268	93	68.89	45
68	63.36	262	94	73.53	34
69	63.57	258	95	73.08	26
70	64.37	247	96	85	20
71	65.27	239	97	81.82	11
72	64.66	232	98	87.5	8
73	64.76	227	99	100	4
74	64.35	216	100	100	2
75	64.71	204			

Table E.5: The results for 500-MLP classifiers with all available success points and bagging- Kidney Transplant

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points	Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
250	60.5	438	314	61.99	321	378	65.45	191	442	72.22	72
251	60.64	437	315	62.54	315	379	65.26	190	443	72.86	70
252	60.55	436	316	62.5	312	380	65.08	189	444	73.91	69
253	60.37	434	317	62.58	310	381	65.08	189	445	76.19	63
254	60.7	430	318	62.21	307	382	64.89	188	446	76.19	63
255	60.51	428	319	62.09	306	383	65.76	184	447	75.41	61
256	60.33	426	320	62.58	302	384	65.38	182	448	75.41	61
257	60.33	426	321	62.54	299	385	65.75	181	449	74.14	58
258	60.14	424	322	62.63	297	386	66.29	178	450	75.44	57
259	60.1	421	323	62.84	296	387	67.05	176	451	75	56
260	60	420	324	63.57	291	388	66.67	174	452	73.08	52
261	60.14	419	325	63.45	290	389	67.05	173	453	73.08	52
262	60.05	418	326	63.07	287	390	66.86	172	454	73.08	52
263	60.1	416	327	63.07	287	391	66.47	170	455	71.43	49
264	60.24	415	328	62.94	286	392	66.07	168	456	71.43	49
265	60.24	415	329	62.99	281	393	66.46	164	457	71.74	46
266	60.05	413	330	63.08	279	394	66.87	163	458	71.74	46
267	60.05	413	331	62.82	277	395	67.08	161	459	71.74	46
268	60.19	412	332	63.14	274	396	66.67	159	460	74.42	43
269	60.29	408	333	63	273	397	66.67	159	461	73.68	38
270	60.34	406	334	62.87	272	398	66.67	159	462	72.97	37
271	60.45	402	335	63.2	269	399	66.67	156	463	72.22	36
272	60.5	400	336	63.53	266	400	66.45	155	464	72.73	33
273	60.4	399	337	63.5	263	401	65.79	152	465	72.73	33
274	60.3	398	338	63.74	262	402	64.63	147	466	71.88	32
275	60.51	395	339	64.23	260	403	64.14	145	467	76.67	30
276	61.13	391	340	64.09	259	404	63.83	141	468	75.86	29
277	61.18	389	341	64.09	259	405	64.96	137	469	74.07	27
278	61.66	386	342	64.84	256	406	64.96	137	470	76.92	26
279	61.72	384	343	65.1	255	407	64.71	136	471	79.17	24
280	61.78	382	344	65.22	253	408	64.93	134	472	81.82	22
281	61.48	379	345	65.48	252	409	65.41	133	473	80.95	21
282	61.8	377	346	65.86	249	410	65.41	133	474	80.95	21
283	61.97	376	347	66.13	248	411	65.12	129	475	80	20
284	62.3	374	348	66.26	246	412	65.08	126	476	82.35	17
285	62.2	373	349	66.8	244	413	65.6	125	477	81.25	16
286	62.1	372	350	66.67	243	414	65.04	123	478	80	15
287	62.26	371	351	66.94	242	415	66.12	121	479	76.92	13
288	62.23	368	352	67.22	241	416	66.12	121	480	75	12

289	62.4	367	353	67.22	241	417	65.55	119	481	81.82	11
290	62.47	365	354	67.23	238	418	66.1	118	482	90	10
291	62.15	362	355	67.37	236	419	66.1	118	483	90	10
292	62.33	361	356	67.23	235	420	66.1	118	484	90	10
293	62.29	358	357	67.52	234	421	65.52	116	485	88.89	9
294	62.15	354	358	67.38	233	422	66.37	113	486	87.5	8
295	62.32	353	359	67.24	232	423	67.27	110	487	87.5	8
296	62.32	353	360	67.69	229	424	67.59	108	488	83.33	6
297	62.46	349	361	67.7	226	425	66.02	103	489	80	5
298	62.14	346	362	67.26	223	426	67.33	101	490	66.67	3
299	61.92	344	363	66.97	221	427	67	100	491	66.67	3
300	61.92	344	364	66.97	218	428	67.35	98	492	100	2
301	61.95	339	365	66.82	214	429	66.67	96	493	100	2
302	61.95	339	366	66.35	211	430	66.67	96	494	100	2
303	61.61	336	367	66.03	209	431	65.96	94	495	100	1
304	61.49	335	368	65.87	208	432	68.54	89	496	100	1
305	61.68	334	369	66.18	207	433	68.97	87	497	0	0
306	61.56	333	370	66.34	205	434	69.41	85	498	0	0
307	61.45	332	371	66.01	203	435	70.24	84	499	0	0
308	61.33	331	372	65.67	201	436	70.73	82	500	0	0
309	61.52	330	373	66	200	437	70	80			
310	62.08	327	374	65.82	196	438	70	80			
311	61.96	326	375	65.8	193	439	70	80			
312	61.92	323	376	65.8	193	440	71.79	78			
313	61.99	321	377	65.8	193	441	72.97	74			

Table E.6: The results for the Pima Indian Diabetes dataset with 100 bagging

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	77.34	256	76	80.18	227
51	77.34	256	77	80.18	227
52	77.56	254	78	80.18	227
53	77.56	254	79	80	225
54	77.47	253	80	80.36	224
55	77.29	251	81	80.36	224
56	77.29	251	82	80.72	223
57	77.91	249	83	81.28	219
58	78.23	248	84	81.19	218
59	78.86	246	85	81.48	216
60	78.86	246	86	81.31	214
61	79.1	244	87	81.13	212
62	79.42	243	88	81.34	209
63	79.34	242	89	81.25	208
64	79.67	241	90	81.07	206
65	79.67	241	91	81.37	204
66	79.67	241	92	82	200
67	79.92	239	93	83.33	192
68	79.92	239	94	83.6	189
69	80.17	237	95	83.42	187
70	80.34	234	96	84.36	179
71	80.17	232	97	84.21	171
72	80.09	231	98	85.37	164
73	80.26	228	99	87.50	152
74	80.18	227	100	89.39	132
75	80.18	227			

Table E. 7: The results for the Wisconsin Cancer dataset with 100 bagging

Nets agree	Accuracy	Points	Nets agree	Accuracy	Points
50	97.81	228	76	98.24	227
51	97.81	228	77	98.24	227
52	97.81	228	78	98.24	227
53	97.81	228	79	98.24	227
54	97.81	228	80	98.24	227
55	97.81	228	81	98.24	227
56	97.81	228	82	98.24	227
57	97.81	228	83	98.23	226
58	97.81	228	84	98.23	226
59	97.81	228	85	98.23	226
60	97.81	228	86	98.23	226
61	97.81	228	87	98.22	225
62	97.81	228	88	98.22	225
63	97.81	228	89	98.22	225
64	97.81	228	90	98.22	225
65	98.24	227	91	98.22	225
66	98.24	227	92	98.21	224
67	98.24	227	93	98.21	224
68	98.24	227	94	98.21	224
69	98.24	227	95	98.21	223
70	98.24	227	96	98.2	222
71	98.24	227	97	98.2	222
72	98.24	227	98	98.19	221
73	98.24	227	99	98.19	221
74	98.24	227	100	98.64	220
75	98.24	227			

.

Bibliography

[1] Aikens, J. S., Kunz, J. C., Shortliffe, E. H., and Fallat, R. J. (1983), "PUFF: An expert system for interpretation of pulmonary function data." *Computers and Biomedical Research* 16: 199-208.

[2] Andrews, R., Diederich, J., and Tickle, A. B. (1995), "A Survey and Critique Of Techniques For Extracting Rules From Trained Artificial Neural Networks." *Knowledge Based Systems* 8: 373-89.

[3] Andrews, R. and Geva, S. (1994), "Rule extraction from a constrained error back propagation MLP." *Proceedings of the 5th Australian Conference on Neural Networks*: 9-12.

[4] Andrews, R. and Geva, S. (1995), "Inserting and extracting knowledge from constrained error back propagation networks." *Proceedings of the 6th Australian Conference on Neural Networks*: 29-32.

[5] ANZDATA (2000), "Data Dictionary: ANZDATA Registry Database." URL: www.anzdata.org.au/.

[6] Ashutosh, K., Lee, H., Mohan, C. K., and et al. (1992), "Prediction criteria for successful weaning from respiratory support: statistical and connectionist ``analyses." *Critical Care Medicine* 20: 1295-301.

[7] Asker, L. (1997), "Ensembles as a sequence of classifiers." *15th International Joint Conference on Artificial Intelligence (IJCAI97)*: 860-65.

[8] Bagley, S., White, H., and Golomb, B. (2001), "Logistic regression in the medical literature: standards for use and reporting, with particular attention to one medical domain." *The Journal of Clinical Epidemiology* 54 (10): 979-85.

[9] Barnett, G. O., Cimino, J. J., Hupp, J. A., and Hoffer, E. P. (1987), "DXplain: an evolving diagnostic decision-support system." *The Journal of the American Medical Association* 258 (1): 67–74.

[10] Bates, D. W., Cohen, M., L., L. L., Overhage, J. M., Shabot, M. M., and Sheridan, T. (2001), "Reducing the frequency of errors in medicine using information technology." *The Journal of the American Medical Association* 8 (4): 299-308.

[11] Bauer, E. and Kohavi, R. (1999), "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants." *Machine Learning* 36: 105-139.

[12] Baxt, W. G. (1991), "Use of an artificial neural network for the diagnosis of myocardial infarction." *Annals of Internal Medicine* 115: 843-8.

[13] Baxt, W. G. (1995), "Application of artificial neural networks to clinical medicine." *Lancet* 346: 1135-7.

[14] Baxt, W. G. and White, H. (1995), "Bootstrapping confidence intervals for clinical input variable effects in a network trained to identify the presence of acute myocardial infarction." *Neural Computation* 7: 624-638.

[15] Bellman, R. E. and Zadeh, L. A. (1970), "Decision-making in a fuzzy environment." *Management Sci.* 17: B141-B-164.

[16] Berrar, D. P., Sturgeon, B., Bradbury, I., Downes, C. S., and Dubitzky, W. (2003), "Microarray Data Integration and Machine Learning Techniques for Lung Cancer Survival Prediction." *Proceedings of the in Critical Assessment of Microarray Data Analysis (CAMDA 2003)*: 43-54, URL:http://www.camda.duke.edu/camda03/papers/days/friday/berrar/paper, Viewed 12 Oct 2006.

[17] Bertsekas, D. P. and Tsitsiklis, J. N. (1996), *Neuro-Dynamic Programming.*, First ed. (Athena Scientific, Belmont, MA).

[18] Blackmore, K. and Bossomaier, T. (2003), "Using a Neural Network and Genetic Algorithm to Extract Decision Rules." *Proceeding of the 8th Australian and New Zealand Intelligent Information Systems Conference (ANZIIS)* 1: 187-91.

[19] Boz, O. (2002), "Extracting Decision Trees From Trained Neural Networks." *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*: 456-61.

[20] Breiman, L. (1996), "Heuristics of instability and stabilization in model selection." *Annals of Statistics* 24: 2350-83.

[21] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984), *Classification and Regression Tree.*, (Wadsworth, Inc., NY).

[22] Breslow, L. A. and Aha, D. W. (1997), "Simplifying Decision trees:A survey." *Knowlege Engineering Review* 12 (1): 1-40.

[23] Brieman, L. (1996), "Bagging predictors." *Machine Learning* 24 (2): 123-140.

[24] Brownbridge, G., Evans, A., Fitter, M., and Platts, M. (1986), " An interactive computerized protocol for the management of hypertension: effects on the general practitioner's clinical behaviour." *Royal College of General Practitioners* 36: 198-202.

[25] Buchanan, B. G. and Feigenbaum, E. A. (1978), "DENDRAL and META-DENDRAL: their applications dimensions." *Artificial Intelligence* 11: 5-24.

[26] Castro, J. L., Mantas, C. J., and Benitez, J. M. (2003), " Interpretation of artificial neural networks by means of fuzzy rules." *IEEE Transactions on Neural Networks* 13 (1): 101-116.

[27] Chandrasekaran, B. (1986), "Generic tasks for knowledge-based reasoning:High-level building blocks for expert system design." *IEEE Expert*, 1 (3): 23-30.

[28] Chase, C. R., Vacek, P. M., Shinozaki, T., Giard, A. M., and Ashikaga, T. (1983), "Medical information management: improving the transfer of research results to presurgical evaluation." *Medical Care* 21: 410-24.

[29] Chen, C. H. (1999),"On the relationship between statistical pattern recognition and artificial neural networks." New York, World Scientific.

[30] Cherkauer, K. J. (1996), "Human expert level performance on a scientific image analysis task by a system using combined artificial neural networks." *Proceedings of the 13th AAAI Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*: 15-21.

[31] Cheung, N. (2001), "Machine Learning Techniques for Medical Analysis" Honours Thesis, School of Information Technology and Electrical Engineering, University of Queensland.

[32] Clancey, W. J. and Letsinger, R. (1981), "NEOMYCIN: reconfiguring a rule-based expert system for application to teaching." *Proceedings of the Seventh Intl Joint Conf on AI*, : 829-835.

[33] Clancey, W. J. and Shortliffe, E. H. (1984), "Readings in Medical Artificial Intelligence - The First Decade.", (Addison-Wesley, MA).

[34] Cox, R., Clark, D., and Richardson, A. (1999), "An investigation into the effect of ensemble size and voting threshold on the accuracy of neural network ensembles." *Proceedings of the 12th Australian Joint Conference of Artificial Intelligence*: 268-77.

[35] Craven, M. (1996), "Extracting Comprehensible Models from Trained Neural Networks." Department of Computer Sciences, University of Wisconsin, PhD thesis, Madison.

[36] Craven, M. and Shavlik, J. (1999), "Rule Extraction: Where Do We Go from Here?" *University of Wisconsin Machine Learning Research Group Working Paper, 99-1*.

[37] Craven, M. W. and Shavlik, J. W. (1994), "Using sampling and queries to extract rules from trained neural networks." *Proceedings of the 11th International Conference on Machine Learning*: 37-45.

[38] Craven, M. W. and Shavlik, J. W. (1996a), "Extracting tree-structured representations from trained networks." *Advances in Neural Information Processing Systems* 8: 24-30.

[39] Craven, M. W. and Shavlik, J. W. (1996b), "Extracting tree-structured representations of trained networks." *Proceedings of the Advances in Neural Information Processing Systems* 8: 24-30.

[40] Cross, S. S., Harrison, R. F., and Kennedy, R. L. (1995), " Introduction to neural networks." *Proceedings of the Lancet* 346: 1075-9.

[41] Das, G., Lin, K., H., M., Renganathan, G., and P., S. (1998), "Rule discovery from time series", *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*.

[42] d'Avila Garcez, A. S., Broda, K., and Gabbay, D. M. (2001), "Symbolic knowledge extraction from trained neural networks: a sound approach." *Artificial Intelligence* 125 (1-2): 155-207.

[43] Dayhoff, J. E. and DeLeo, J. M. (2001), "Artificial neural networks: opening the black box." *Cancer* 91 (8): 1615-35.

[44] Deco, G. and Obradovic, D. (1996), *An Information-Theoretic Approach to Neural Computing.*, (Springer-Verlag, Ny).

[45] Diederich, J., Hild, H., and Bakiri, G. (1995), "A comparison of ID3 and backpropagation for English test-to-speech mapping." *Machine Learning* 18: 51-80.

[46] Dietterich, T. G. (1997), "Machine learning research: four current directions." *AI Magazine* 16: 97-136.

[47] Domingos, P. (1998), "Knowledge Discovery Via Multiple Models." *Intelligent Data Analysis* 2: 187-202.

[48] Dorsey, S. G., Waltz, C. F., Brosch, L., Connerney, I., Schweitzer, E. J., and Barlett, S. T. (1997), "A neural network model for predicting pancreas transplant graft outcome." *Diabetes Care* 20: 1128-33.

[49] Doyle, H., Dvorchik, I., Mitchell, S., Marino, I., Ebert, F., McMichael , J., and Fung, J. (1994), "Predicting outcomes after liver transplantation. A connectionist approach." *Annals of Surgery* 219 (4): 408-415.

[50] Duch, W., Adamczak, R., and Grabczewski, K. (2001), "A new methodology of extraction, optimization and application of crisp and fuzzy logical rules", *IEEE Transactions on Neural Networks* 12: 277-306.

[51] Efron, B. and Tibshirani, R. (1993), *An introduction to the bootstrap.*, (Chapman & Hall, New York).

[52] Eftekhar, B., Mohammad, K., Eftekhar, A. H., Ghodsi, M., and Ketabchi, E. (2005), "Comparison of artificial neural network and logistic regression models for prediction of mortality in head trauma based on initial clinical data", *Biomed Central, Medical Informatics and Decision Making* 5 (3): Published online 15 February 05.

[53] Ennett, C. M., Frize, M., and Walker, C. R. (2001), "Influence of missing values on artificial neural network performance." *Medinfo* 10 (part 1): 449-53.

[54] Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P. (1996), *Data Mining to Knowledge Discovery: An Overview, in Advances in Knowledge Discovery in Databases.*, (MIT Press,, Cambridge, MA).

[55] File, P. E., P. I. Dugard, and A. S. Houston, (1994), "Evaluation of the use of induction in the develeopment of a medical expert system." *Comp. and Biomedical Research* 27 (5): 383-395.

[56] Freeman, R. V., Eagle, K. A., Bates, E. R., and Werns, S. W. e. a. (2000), "Comparison of artificial neural networks with logistic regression in prediction of in-hospital death after percutaneous transluminal coronary angioplasty." *American Heart Journal* 140 (3): 511-20.

[57] Freund, Y. (1995), "Boosting a weak algorithm by majority." *Information and Computation* 121: 256-85.

[58] Freund, Y. and Schapire, R. (1996), "Experiments with a New Boosting Algorithm." *Proceedings of the International Machine Learning Conference*: 148-156.

[59] Fu, L. (1991), "Rule learning by searching on adapted nets." *Proceedings of the 9th National Conference on Artificial Intelligence*: 590-95.

[60] Fu, L. (1994), "Rule generation from neural networks." *IEEE Transactions on Systems, Man, and Cybernetics* 28 (8): 1114-24.

[61] Fuchs, J., Heller, I., Topilsky, M., and Inbar, M. (1999), "CaDet, A Computer-Based Clinical Decision Support System for Early Cancer Detection." *Cancer Detection and Prevention 1999; 23(1):78-87* 23 (1): 78-87.

[62] Gabutti, L., Lötscher, N., Bianda, J., Marone, C., Mombelli, G., and Burnier, M. (2006), "Would artificial neural networks implemented in clinical wards help nephrologists in predicting epoetin responsiveness?" *BMC Nephrology* 7 (13).

[63] Gallant, S. (1988a), "Connectionist expert systems." *Communications of the ACM*, 31 (2): 152-169.

[64] Gallant, S. I. (1988b), "Connectionist expert systems." *Communications of the ACM*, 31 (2): 152-169.

[65] Gallant, S. I. and Hayashi, Y. (1991), "A Neural Network Expert System with Confidence Measurements.," in *Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU'90*, B. Bouchon-Meunier, R. R. Yager, and L. A. Zadeh, Eds., (Springer, Berlin, Heidelberg), 562-567.

[66] Gaudart, J., Poudiougou, B., Ranque, S., and Doumbo, O. (2005), "Oblique decision trees for spatial pattern detection: optimal algorithm and application to malaria risk." *Medical Research Methodology (BMC)* 5 (1): 22.

[67] Gericke, B., Koebnick, C., Reimann, M., Forterre, S., Franz Zunft, H. J., and Schweigert, F. J. (2005), "Influence of hormone replacement therapy on proteomic pattern in serum of postmenopausal women." *Maturitas* 51 (4): 334-42.

[68] Giarratano, J. C. and Riley, G. D. (1998), "Expert Systems: Principles and Programming." (Third Edition, Course Technology Publishing Company), 624 pages.

[69] Giles, C. L., Miller, C. B., Chen, D., Chen, H., Sun, G. Z., and Lee, Y. C. (1992), "Learning and extracting finite state automata with second-order recurrent neural networks." *Neural Computation* 4 (3): 393-405.

[70] Gill, P. E., Murray, W., and Wright, M. H. (1981), *Practical Optimization.*, (Academic Press, London).

[71] Goldberg, D. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning.*, (Addison-Wesley, Reading,MA).

[72] Golea, M. (1996), "On the complexity of rule extraction from neural networks and network querying." *Proceedings of the Rule Extraction From Trained Artificial Neural Networks Work shop. Society for the study of Artificial Intelligence and Simulation of Behavior Workshop Series(AISB'96)*: 51-59.

[73] Goutte, C. (1997), "Note on free lunches and cross-validation." *Neural Computation* 9: 1211-15.

[74] Guerriere, M. R. J. and Detsky, A. S. (1991), "Neural networks: what are they?" *Ann Intern Med* 115: 906-7.

[75] Gutta, S. and Wechsler, H. (1996), "Face recognition using hybrid classifier systems." *Proceedings of the IEEE International Conference on Neural Networks*: 1017-22.

[76] Hansen, L. K. and Salamon, P. (1990), "Neural network ensembles." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12: 993-1001.

[77] Heckerling, P. S., Gerber, B. S., Tape, T. G., and Wigton, R. S. (2003), "Prediction of community-acquired pneumonia using artificial neural networks." *Med. Decis. Making* 23: 112-21.

[78] Holland, J. (1986), "Escaping brittleness.," in *Machine Learning*, vol. 2, R. Michalski, C. J., and S. Mitchell, Eds., (Morgan Kaufmann), 593-623.

[79] Hosmer, D. W. and Lemeshow, S. (1989), "Applied logistic regression.", (John Wiley, New York).

[80] Huang, F. J., Zhou, Z. H., Zhang, H. J., and Chen, T. (2000), "Pose invariant face recognition." *Proceedings of the 4th IEEE International Conference on Automatic Face and Gesture Recognition*: 245-50.

[81] Itchhaporia, D., Snow, P. B., and Almassy, R. J. e. a. (1996), "Artificial neural networks: current status in cardiovascular medicine." *Journal of the American College of Cardiology* 28: 2515-21.

[82] Jaimes, F., Farbiarz, J., Alvarez, D., and Martínez, C. (2005), "Comparison between logistic regression and neural networks to predict death in patients with suspected sepsis in the emergency room." *PubMed Central (PMC) , The National Institutes of Health (NIH) Journal* 9 (2): 150-156.

[83] Jayadeva, Khemchandani, R., and Chandra, S. (2007), "Twin Support Vector Machines for Pattern Classification", *Pattern Analysis and Machine Intelligence, IEEE Transactions* 29 (5): 905-10.

[84] Johansson, U., König, R., and Niklasson, L. (2003), "Rule Extraction from Trained Neural Networks using Genetic Programming." *13th International Conference on Artificial Neural Networks*: supplementary proceedings 13-16.

[85] Johansson, U. and Niklasson, L. (2001), "Predicting the Impact of Advertising - a Neural Network Approach." *Proceedings of the International Joint Conference on Neural Networks*: 1799-1804.

[86] Johansson, U. and Niklasson, L. (2002), "Neural Networks - from Prediction to Explanation." *Proceedings of the International Conference Artificial Intelligence and Applications*: 93-98.

[87] Katz, S., A.S., K., Lowe, N., and Quijano, R. C. (1994), "Neural net-bootstrap hybrid methods for prediction of complications in patients implanted with artificial heart valves." *Heart Valve Dis.* 3: 49-52.

[88] Keedwell, E., Narayanan, A., and Savic, D. (2000), "Creating rules from trained neural networks using genetic algorithms", *International Journal of Computers, Systems and Signals(IJCSS)* 1: 30-42.

[89] Khum, T., S. (2001), "What Determines The Outcome of Kidney Transplants." MIT Alliance, National University of Singapore, Master thesis, Singapore.

[90] King, R. D., Muggleton, S., Lewis, R. A., and Sternberg, M. J. E. (1992), "Drug design by machine learning: the use of inductive logic programming to model the structure-activity relationship of trimethoprim analogues binding to dihydrofolate reductase." *Proceedings of the National Academy of Sciences* 89: 11322-26.

[91] Koutroumbas, K., Paliouras, G., Karkaletsis, V., and Spyropoulos, C. D. (2001), "Comparison of Computational Learning Methods on a Diagnostic Cytological Application." *Proceedings of the European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems (EUNITE)*: 500-508.

[92] Krishnan, R. (1997), "A systematic method for decompositional rule extraction from neural networks." *Proceedings of the NIPS'96 Workshop on Rule Extraction from Trained Artificial Neural Networks*: 38-45.

[93] Kukar, M. (1997), "Cost-sensitive learning in medical diagnostics." *Proceedings of the Computer-Aided Data Analysis in Medicine.*

[94] Lange, N. (2003), "What can modern statistics offer imaging neuroscience?" *Statistical Methods in Medical Research* 12 (5): 447-69.

[95] LaPuerta, P., L'Italien, G. J., and Paul, S. e. a. (1998), "Neural network assessment of perioperative cardiac risk in vascular surgery patients." *Med Decis Making* 18: 70-5.

[96] Ledley, R. S. and Lusted, L. B. (1959), "Reasoning foundations of medical diagnosis." *Science* 130: 9-21.

[97] Lette, J., Colletti, B. W., Cerino, M., and et al. (1994), "Artificial intelligence versus logistic regression statistical modeling to predict cardiac complications after noncardiac surgery." *Clinical Cardiology* 17: 609-14.

[98] Levenberg, K. (1994), "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics* 2: 164-68.

[99] Li, L., Huang, J., Sun, S., Shen, J., Unverzagt, F., Gao, S., Hendrie, H., Hall, K., and Hui, S. (2004), "Selecting pre-screening items for early intervention trials of dementia-a case study." *Statistical Medicine* 23 (2): 271-83.

[100] Liberati, D. and Setti, E. (1994), "Pappaletterrra M., The application of neural networks in predicting the success of kidney transplants," *Automazione Energia Informazione* 3 (7): 67-70.

[101] Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, v. (1980), "Application of Artificial Intelligence for Chemistry : The DENDRAL Project.", (New York: McGraw-Hill).

[102] Lisboa , P. J. (2002), "A review of evidence of health benefit from artificial neural networks in medical intervention." *Neural Net* 15 (1): 11-39.

[103] Long, W. J., Griffith, J. L., Selker, H. P., and D'Agostino, R. B. (1993), "A comparison of logistic regression to decision-tree induction in a medical domain." *Computer Biomedical Research.* 26 (1): 74-97.

[104] Mangasarian, L. and Wolberg, H. W. (1990), "Cancer diagnosis via linear programming." *Society for Industrial and Applied Mathematics (SIAM)* 23 (5): 1-18.

[105] Masuoka, R., Watanabe, N., Kawamura, A., Owada, Y., and Asakawa, K. (1990), "Neurofuzzy systems - fuzzy inference using a structured neural network." *Proceedings of the International Conference on Fuzzy Logic and Neural Networks*: 173-77.

[106] Matis, S., Doyle, H., Marino, I., Murad, R., and Uberbacher, E. (1995), "Use of Neural Networks for Prediction of Graft Failure following Liver Transplantation." *Proceedings of the 8th Annual IEEE Symposium on Computer-Based Medical Systems*: 0133.

[107] McCance, D. R., Dyer, D. G., Dunn, J. A., Bailie, K. E., Thorpe, S. R., Baynes, J. W., and Lyons, T. J. (1993), "Maillard reaction products and their relation to complications in insulin-dependent diabetes mellitus." *Journal of Clinical Invest* 91 (6): 2470-78.

[108] McDonald, C. J. (1976a), "Use of a computer to detect and respond to clinical events: its effect on clinician behavior." *Annals of Internal Medicine* 84: 162-7.

[109] McDonald, C. J. (1976b), " Protocol-based computer reminders, the quality of care and the non-perfectability of man." *N Engl J Med* 295: 1351-5.

[110] McDonald, C. J., Hui, S. L., Smith, D. M., Tierney, W. M., Cohen, S. J., and Weinberger, M. e. a. (1984), "Reminders to physicians from an introspective computer medical record. A two-year randomized trial." *Ann Intern Med* 100: 130-8.

[111] McDonald, C. J., Wilson, G. A., and McCabe, G. P. (1980), "Physician response to computer reminders." *The Journal of the American Medical Association (JAMA)* 244: 1579-81.

[112] McDowell, I., Newell, C., and Rosser, W. (1986), "Comparison of three methods of recalling patients for influenza vaccination." *Canadian Medical Association* 135: 991-7.

[113] McDowell, I., Newell, C., and Rosser, W. (1989), "Computerized reminders to encourage cervical screening in family practice." *The Journal of Family Practice* 28: 420-4.

[114] Medsker, L. R. (1994), *Hybrid Neural Network and Expert Systems.*, (Kluwer Academic Publishers, Boston).

[115] Michael, E. B., Prasun, C., Klein, R. B., and Klein, J. B. (2003), "Prediction of delayed renal allograft function using an artificial neural network." *Nephrol Dial Transplant* 18: 2655-2659.

[116] Michie, D., Spiegelhalter, D. J., and Taylor, C. (1994), *Machine learning neural nets and statistical classification.*. (Ellis-Horwood, Chichester, Chichester).

[117] Miller, A., Blott, B., and Hames, T. (1992), "Review of Neural Network Applications in Medical Imaging and Signal Processing. Medical and Biological Engineering and Computing." 30 (5): 449-464.

[118] Miller, R. A., Masarie, F. E., and Myers, J. D. (1986), "Quick medical reference (QMR) for diagnostic assistance." *MD Computing* 3 (5): 34-48.

[119] Mingers, J. (1989), "An empircial comparison of pruning methods for decision-tree induction." *Machine Learning* 4: 227-43.

[120] Mitchell, T. (1996), *An introduction to Genetic Algorithms.*, (MIT Press, Cambridge, MA).

[121] Mitchell, T. (1997), *Machine Learning.*, (McGraw-Hill, New York).

[122] Mitra, S. (1994), "Fuzzy MLP based expert system for medical diagnosis." *Fuzzy Sets and Systems* 65 (2-3): 285-96.

[123] Mobley, B. A., Leasure, R., and Davidson, L. (1995), "Artificial neural network predictions of length of stay on a post-coronary care unit." *Heart Lung* 24: 251-6.

[124] Morio, S. (1989), "An expert system for early detection of cancer of the breast." *Computers in Biology and Medicine* 9 (5): 295-306.

[125] Mueller, M., Wagner, C. L., Annibale, D. J., Knapp, R. G., Hulsey, T. C., and Almeida, J. S. (2006), "Parameter selection for and implementation of a web-based decision-support tool to predict extubation outcome in premature infants." *BMC Medical Informatics* 6 (11).

[126] Murthy, S. K. (1998), "Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey." *Data Mining and Knowledge Discovery* 2 (4): 345-89.

[127] Musen, M. A. (1999), "Stanford Medical Informatics: uncommon research, common goals." *MD Computer* 16 (1): 47-50.

[128] Omlin, C. W., Giles, C. L., and Miller, C. B. (1992), "Heuristics for the extraction of rules from discrete time recurrent neural networks." *Proceedings of the International Joint Conference on Neural Networks* 1: 33-38.

[129] Opelz, G., Gustafsson, L. A., and Terasaki, P. I. (1976), "Influence of interval between first graft removal and retransplantation on outcome of second cadaver kidney grafts." *Transplantation* 19 (226).

[130] Opitz, D. W. and Shavlik, J. W. (1996), "Actively searching for an efective neural network ensemble." *Connection Science* 8: 337-53.

[131] Ortiz, J., Ghefter, C. G. M., and Silva, C. E. S. (1995), "One-year mortality prognosis in heart failure: a neural network approach based on echocardiographic data." *Journal of the American College of Cardiology* 26: 1586-93.

[132] Pantazopoulos, d., Karakitsos, p., Iokim-liossi, a., Pouliakis, a., Botsoli-stergiou, e., and Dimopoulos, c. (1998), "Back propagation neural network in the discrimination of benign from malignant lower urinary tract lesions." *Journal of Urology* 159 (5): 1619-23.

[133] Pesonen, E. (1997), *Is neural network better than statistical methods in diagnosis of acute appendicitis?* Medical Informatics Europe, (IOS Press, Amsterdam, Netherlands).

[134] Petrovsky, N., Khum, T. S., Brusic, V., Russ, G., Socha, L., and Bajic, V. B. (2002), "Use of artificial neural networks in improving renal transplantation outcomes." *Graft*: 6-13.

[135] Petrucci, K., Petrucci, P., Canfield, K., McCormick, K. A., Kjerulff, K., and Parks, P. (1991), "Evaluation of UNIS: Urological Nursing Information Systems", *Proceedings of the Fifteenth Annual Symposium on Computer Applications in Medical Care*: 43-7.

[136] Piccolo, D., Ferrari, A., Peris, K., Diadone, R., Ruggeri, B., and Chimenti, S. (2002), "Dermoscopic diagnosis by a trained clinician vs. a clinician with minimal dermoscopy training vs. computer-aided diagnosis of 341 pigmented skin lesions: a comparative study." *British Journal of Dermatology* 147 (3): 481-6.

[137] Provost, F., Fawcett, T., and Kohavi, R. (1998), "The case against accuracy estimation for comparing induction algorithms." *Proceedings of the Fifteenth International Conference on Machinen Learning*.

[138] Quinlan, J. (1996), "Bagging, boosting, and C4.5." *Proceedings of the Thirteenth National Conference on Artificial Intelligence*: 725-30.

[139] Quinlan, J. R. (1986), "Induction of decision trees." *Machine Learning* 1: 81-106.

[140] Quinlan, J. R. (1987a), "Simplifying decision trees." *Int. J. of Man-Machine Studies*, 27: 221-34.

[141] Quinlan, J. R. (1987b), "Generating production rules from decision trees." *In Proceedings of the Tenth International Joint Conference on Artificial Intelligence*: 301-07.

[142] Quinlan, J. R. (1993), *C4.5: programs for machine learning.*, (Morgan Kaufmann Publishers Inc., San Francisco, CA,).

[143] Quinlan, J. R. (2000), "Data mining tools See5 and C5.0." available online from http://www.rulequest.com/see5-info.html.

[144] Rajimehr, R., Farsiu, S., Kouhsari, L. M., Bidari, A., Lucas, C., Yousefian, S., and Bahrami, F. (2002), "Prediction of lupus nephritis in patients with systemic lupus erythematosus using artificial neural networks", *Lupus* 11 (8): 485-492.

[145] Ramesh, A. N., Kambhampati, C., Monson, J. R., and Drew, P. J. (2004), "Artificial intelligence in medicine." *Annals of The Royal College of Surgeons of England* 86 (5): 334-8.

[146] Rapaport, F. T. (1995), "The current status of the HLA controversy in clinical transplantation," *Transplant. Proc* 27 (1).

[147] Reed, R. D. and Marks, R. J. (1999), *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks.*, (The MIT Press, Cambridge, MA).

[148] Ribeiro, B. (2003), "Support Vector Machines For Medical Data Bases Classification. " in *EUNITE Summer Course on Soft Computing in Medicine, Coimbra*.).

[149] Ribeiro, B. (2005), "Selective Classification and Regression Models Based on Support Vector Clustering." *Neural, Parallel & Scientific Computations, Dynamic Publishers*, 13: pp. 327-36.

[150] Riffenburgh, R. H. (2006), *Statistics in Medicine*. (Elsevier academic Press, MA, USA).

[151] Ripley, B. D. (1996), *Pattern Recognition and Neural Networks.*, (Cambridge University Press, Cambridge).

[152] Roger, R. (1985), "Hastie, Trevor, Tibshirani, R., Friedman, J. The Elements of Statistical Learning Data Mining,Inference, and Prediction.", (Also see: New York Times, New York), http://dpls.dacc.wisc.edu/pubs/Newsletters/feb04news.html.

[153] Rogers, J. L., Haring, O. M., and Goetz, J. P. (1984), "Changes in patient attitudes following the implementation of a medical information system." *QRB* 10: 65-74.

[154] Rosser, W. W., Hutchison, B. G., McDowell, I., and Newell, C. (1992), "Use of reminders to increase compliance with tetanus booster vaccination." *Canadian Medical Association* 146: 911-7.

[155] Rotolo LS, Wilson, J. B., Ginsberg, M. D., and Ledley, R. S. (1963), "Digital computer picture processor." *Proceedings of the Sixteen Annual Conference on Engineering in Medicine and Biology (ACEMB)*: 14-15.

[156] Rudolfer, S. M., Paliouras, G., and Peers, I. (1999), "A Comparison of Logistic Regression to Decision Tree Induction in the Diagnosis of Carpal Tunnel Syndrome." *Computers and Biomedical Research*, 32 (391-414).

[157] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986), "Learning internal representations by error propagation." *Proceedings of the Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 13: 318-62.

[158] Safavin, S. R. and Landgrebe, D. (1991), "A survey of decision tree classifier methodology." *IEEE Trans. on Systems, Man and Cybernetics*, 21 (3): 660-674.

[159] Saito, K. and Nakano, R. (1988), "Medical diagnostic expert system based on PDP model." *Proceedings of the IEEE Int. Conf. Neural Networks* 1: 255-62.

[160] Santos-Garcia, G., Varela, G., Novoa, N., and Jimenez, M. F. (2004), "Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble." *Artificial Intelligence in Medicine* 30 (1): 61-69.

[161] Santos-García, G., Varela, G., Novoa, N., and Jiménez, M. F. (2004), "Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble." *Artif Intell Med.* 30 (1): 61-9.

[162] Schapire, R. E. (1990), "The strength of weak learnability." *Machine Learning* 5: 197-227.

[163] Schwartz, M. H., Ward, R. E., MacWilliams, C., and Verner, J. J. (1997), "Using neural networks to identify patients unlikely to achieve a reduction in body pain after total hip replacement surgery." *Medical Care* 35: 1020-30.

[164] Scott, A. C., Clancey, W. J., Davis, R. W., and Shortliffe, E. H. (1984), "Rule-Based Expert Systems: the MYCIN experiments of the Stanford Heuristic Programming Project," in *Addison-Wesley, Reading MA (The book is available online from AAAI's Classic Books in AI collection,* B. G. Buchanan and E. H. Shortliffe, Eds.).

[165] Scott, R. (1993), "Artificial intelligence: its use in medical diagnosis." *The Journal of Nuclear Medicine* 34: 510-4.

[166] Sestito, S. and Dillon, T. S. (1991), "The use of sub-symbolic methods for the automation of knowledge acquisition for expert systems." *Proceedings of the Eleventh International Conference Expert Systems and their Applications. Avignon* 1: 317-28.

[167] Sestito, S. and Dillon, T. S. (1994), *Automated Knowledge Acquisition.*, (Prentice Hall, Australia).

[168] Setiono, R. (1997), "Extracting rules from neural networks by pruning and hidden-unit splitting." *Neural Computation* 9 (1): 205-25.

[169] Shadabi, F., Cox, R., Sharma, D., and Petrovsky, N. (2004), "Use of Artificial Neural Networks in the Prediction of Kidney Transplant Outcomes." *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems(KES 04)* 3: 566-572.

[170] Shadabi, F. and Khodai-Joopari, M. (2003), "Case Based Reasoning Models in Management Application", *Proceedings of the 8th Australian and New Zealand Intelligent Information Systems Conference, ANZIIS03, Sydney, Australia*: 457-62.

[171] Shadabi, F. and Khodai-Joopari, M. (2004), "An Investigation of Prediction Techniques in Management Application." *Proceedings of the 2nd International Conference on Artificial Intelligence in Science and Technology, Hobart, Australia*: 47-50.

[172] Shao, J. (1993), "Linear model selection by cross-validation." *Journal of the American Statistical Association,* 88: 486-494.

[173] Shavlik, J. W., Mooney, R., and Towell, G. G. (1991), "Symbolic and neural learning algorithms: an experimental comparison." *Machine Learning* 6: 111-43.

[174] Sheppard, D., McPhee, D., Darke, C., Shrethra, B., Moore, R., Jurewitz, A., and Gray, A. (1999), "Predicting Cytomegalovirus disease after renal transplantation: an artificial neural network approach." *International Journal of Medical Informatics* 54: 55-76.

[175] Shortliffe, E. H. (1976), *Computer-Based Medical Consultation: MYCIN.*, (American Elsevier, New York. NY).

[176] Shortliffe, E. H., Axline, S. G., Buchanan, B. G., Merigan, T. C., and Cohen, S. N. (1973), "An artificial intelligence program to advise physicians regarding antimicrobial therapy." *Computer Biomedical Research.* 6 (6): 544-560.

[177] Sima, J. (1995), "Neural expert systems." *Journal of the International Neural Network* 8 (2): 261-71.

[178] Simpson, P. K. (1992), "Artificial Neural Systems : Foundations, Paradigms, Applications, and Implementations.", (Pergamon Press, New York).

[179] Sotos, J. (1990), "MYCIN and NEOMYCIN: two approaches to generating explanations in rule based expert systems." *Aviation, Space, and Environmental Medicine* 61 (950-4).

[180] Stone, M. (1974), "Cross-validatory choice and assessment of statistical predictions." *Journal of the Royal Statistical Society* 36 (1): 111-147.

[181] Szolovits, P. (1982), "Artificial Intelligence in Medicine. " in *AAAS Selected Symposia Series,* Westview Press, Ed., Colorado).

[182] Szukalski, S. K., Cox, R., and Crowther, P. S. (2005), "Using Artificial Neural Network Ensembles to Extract Data Content from Noisy Data", *9th International Conference on Knowledge-Based Intelligent Information & Engineering Systems(KES 05)* 3: 974-980.

[183] T. K Sen, Oliver, R., and N Sen (1995), *Predicting Corporate mergers*. In Neural Networks and the Capital Markets Refenes,A.P. (ed.), (John Wiley and Sons, New York.).

[184] Taha, I. and Ghosh, J. (1996), *Symbolic Interpretation of Artificial Neural Networks. Technical Report TR-9701-106, University of Texas, 1996.* Technical Report TR-9701-106, University of Texas, Texas).

[185] Thrun, S. B. (1994), "Extracting provably correct rules from artificial neural networks." *Technical Report IAI-TR-93-5*: University of Bonn, Germany.

[186] Thrun, S. B., Bala, J., Bloedorn, E., Bratko, I., Cestnik, B., Cheng, J., Jong, K. D., Dzeroski, S., Hamann, R., Kaufman, K., Keller, S., Kononenko, I., Kreuziger, J., Michalski, R. S., Mitchell, T., Pachowicz, P., Roger, B., Vafaie, H., de Velde, W. V., Wenzel, W., Wnek, J., and Zhang, J. (1991), "The MONK's Problems: A Performance Comparison of Different Learning Algorithms", *Technical Report CMU-CS-91-19, Computer Science Department*: Carnegie Mellon University.

[187] Tickle, A. B., 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." *IEEE Transactions on Neural Networks* 9 (6): 1057-67.

[188] Tickle, A. B., Orlowski, M., and Diederich, J. (1996), "DEDEC: a methodology for extracting rules from trained artificial neural networks." *Proceedings of the AISB'96 Workshop on Rule Extraction from Trained Neural*: 90-102.

[189] Towell, G. and Shavlik, J. (1993), "The extraction of refined rules from knowledge based neural networks." *Machine learning* 13 (1): 71-101.

[190] Towell, G. and Shavlik, J. (1994), " Knowledge-based articial neural networks." *Artificial Intelligence* 70 (1,2): 119-165.

[191] Tu, J. V. (1996), "Advantage and Disadvantage of Using Artificial Neural Networks versus Logistic Regression for Predicting Medical Outcomes." *The Journal of Clinical Epidemiology* 49 (11): 1225-31.

[192] Tu, J. V. and Guerriere, M. R. J. (1993), "Use of a neural network as a predictive instrument for length of stay in the intensive are unit following cardiac surgery." *American Medical Informatics Association*: 666-72.

[193] UCI Repository, "UCI repository of machine learning databases.." http://www.ics.uci.edu/~mlearn/MLRepository.html: Access-10th Dec 2005, 2005.

[194] Van Melle, W. (1980), "A Domain independent System that Aids in Constructing Knowledge Based Consultation Programs." Computer Science Department, Doctoral thesis, Stanford University.

[195] Van Melle, W., Shortliffe, E. H., and Buchanan, B. G. (1981), "EMYCIN: A Domain Independent System that Aids in Constructing Knowledge-Based Consultation Programs." *In Pergamon-Infotech Report on Machine Intelligence*: 249-263.

[196] Wall, R. and Cunningham, P. (2000), "Exploring the Potential for Rule Extraction from Ensembles of Neural Networks." *11th Irish Conference on Artificial Intelligence & Cognitive Science (AICS)*.

[197] Wall, R., Cunningham, P., Walsh, P., and Byrne, S. (2003), "Explaining the output of ensembles in medical decision support on a case by case basis", *Artificial Intelligence in Medicine* 28: 191-206.

[198] Walther, E., Eriksson, H., and Musen, M. A. (1992), "Plug and Play: Construction of task-specific expert-system shells using sharable context ontologies." *AAAI Workshop on Knowledge Representation Aspects of Knowledge Acquisition, San Jose, CA*: 191-198.

[199] Wang, C. H. and Tseng, S. S. (1990), "A brain tumour diagnostic system with automatic learning abilities." *Proceedings of the 3rd Annual IEEE Symposium on Computer Based Medical Systems*: 313-20.

[200] Waterman, D. A. (1985), "A Guide to Expert Systems..", (Addison-Wesley Teknowledge Series In Knowledge Engineering, Boston, USA), 419 pages.

[201] Weinstein, J., Kohn, K., and Grever, M. (1992), "Neural Computing in Cancer Drug Development: Predicting Mechanism of Action. Science." *Predicting Mechanism of Action. Science* 258: 447-51.

[202] Werbos, P. J. (1994), *The Roots of Backpropagation*. (John Wiley & Sons, NY).

[203] Wilks, P. A. D. and English, M. J. (1994), "Accurate segmentation of respiration waveforms from infants enabling identification and classification of irregular breathing patterns." *Medical Eng. and Physics* 16 (1): 19-23.

[204] William, H. W. and Mangasarian, O. L. (1990), "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", *Proceedings of the National Academy of Sciences, U.S.A.* 87: 9193-9196.

[205] Witten, I. H. and Frank, E. (2000), *Data Mining: Practical Machine Learing Tools and Techniques with Java Implementations.*, (Morgan Kaufmann, San Diego, CA).

[206] Woods, K. S., Doss, C. C., Vowyer, K. W., Solka, J. L., Prieve, C. E., and Kegelmeyer, W. P. J. (1993), "Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography." *Pattern Recognition and Artificial Intelligence*, 7 (6): 1417-36.

[207] Worsley, K. J. (2003), "Detecting activation in fMRI data." *Statistical Methods in Medical Research* 12 (5): 401-418.

[208] Worsley, K. J., Liao, C., Aston, J., Petre, V., Duncan, G. H., Morales, F., and Evans, A. C. (2002), "A general statistical analysis for fMRI data." *NeuroImage* 15: 1-15.

[209] Yu, V. L., Fagan, L. M., and Wraith, S. M., et al (1979), "Antimicrobial selection by a computer: a blinded evaluation by infectious diseases experts." *The Journal of the American Medical Association* 242 (12): 1279-82.

[210] Zadeh, L. A. (1983),"The role of fuzzy logic in the management of uncertainty in expert systems." Memorandum No. UCB/ERL M83/41, University of California, Berkeley.

[211] Zenobi, G. and C., C. (2001), "Using ambiguity in preparing ensembles of classifiers based on different feature subsets to minimize generalization error." *Proceedings of the 12th European Conference in Machine Learning(ECML), Lecture Notes in AI*.

[212] Zhou, Z. and Tang, W. (2003), "Selective Ensemble of Decision Trees." *Lecture Notes in Artificial Intelligence* 2639: 476-83.

[213] Zhou, Z. H. (2004), "Rule extraction: using neural networks or for neural networks?" *Journal of Computer Science and Technology* 19 (2): 249-53.

[214] Zhou, Z. H., Chen, S. F., and Chen, Z. Q. (2000), "A statistics based approach for extracting priority rules from trained neural networks." *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks* 3: 401-06.

[215] Zhou, Z. H. and Jiang, Y. (2003), "Medical diagnosis with C4.5 rule preceded by artificial neural network ensemble." *IEEE Transactions on Information Technology in Biomedicine* 7 (1): 37-42.

[216] Zhou, Z. H. and Jiang, Y. (2004), " NeC4.5: neural ensemble based C4.5." *IEEE Transactions on Knowledge and Data Engineering* 16 (6): 770-73.

[217] Zhou, Z. H., Jiang, Y., and Chen, S. F. (2003), "Extracting symbolic rules from trained neural network ensembles." *AI Communications* 16 (1): 3-15.

[218] Zhou, Z. H., Jiang, Y., Yang, Y. B., and Chen, S. F. (2002), "Lung cancer cell identification based on artificial neural network ensembles." *Artificial Intelligence in Medicine* 24: 25-36.

[219] Zurada, J. (1992), *An Introduction to Artificial Neural Systems.* (West Publishing Company, St. Paul, Minnesota).