DATA MINING OF MAGNETOCARDIOGRAMS FOR PREDICTION OF ISCHEMIC HEART DISEASE

YOSAWIN KANGWANARIYAKUL

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF SCIENCE
(TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2010

Thesis
entitled

DATA MINING OF MAGNETOCARDIOGRAMS FOR PREDICTION OF ISCHEMIC HEART DISEASE

……………….……….….………
Mr. Yosawin Kangwanariyakul
Candidate

……………….……….….………
Asst. Prof. Thanakorn Naenna,
Ph.D. (Engineering Science)
Major-advisor

……………….……….….………
Lect. Chanin Nantasenamat,
Ph.D. (Medical Technology)
Co-advisor

……………….……….….………
Lect. Tanawut Tantimongcolwat,
Ph.D. (Medical Technology)
Co-advisor

……………….……….….………
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……………….……….….………
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Program Director
Master of Science Program in
Technology of Information System
Management
Faculty of Engineering
Mahidol University

Thesis
entitled

DATA MINING OF MAGNETOCARDIOGRAMS FOR PREDICTION OF ISCHEMIC HEART DISEASE

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Science
(Technology of Information System Management)
on
May 14, 2010

……………….……….…..………
Mr. Yosawin Kangwanariyakul
Candidate

………………….…..………………
Lect. Kanokwan Kingpadung,
D.Eng. (Industrial Engineering)
Chair

…………………….…..…………...…
Asst. Prof. Thanakorn Naenna,
Ph.D. (Engineering Science)
Member

…………………….…..………….……
Lect. Chanin Nantasenamat,
Ph.D. (Medical Technology)
Member

…………………….…..…………...…
Assoc. Prof. Walailak Atthirawong,
Ph.D. (Manufacturing Engineering
And Operations Management)
Member

…………………….…..………………..
Lect. Tanawut Tantimongcolwat,
Ph.D.(Medical Technology)
Member

…………………….…..…………...…
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

…………………….…..…………...…
Asst. Prof. Rawin Raviwongse,
Ph.D. (Engineering Management)
Dean
Faculty of Engineering
Mahidol University

ACKNOWLEDGEMENTS

The success of this thesis would not have been succeeded without the support, hard work and efforts of a number of people. First and foremost, I would like to express my sincere gratitude to my major advisor, Asst. Prof. Dr. Thanakorn Naenna for his supervision and valuable guidance to this research.

I also would like to express my gratitude to Dr. Chanin Nantasenamat and Dr. Tanawut Tantimongcolwat, my co-advisor for much help with the useful suggestions and methodology. I sincerely thank to Assoc. Prof. Dr. Walailak Atthirawong, the external examiner of the thesis defense and Dr. Kanokwan Kingpadung, the chair of thesis defense for their kindness in valuable suggestion and many helpful comments.

Besides, thank to my friends and staffs in Master of Science Program in Technology of Information System Management, Mahidol University. Special tanks are due to Mr. Graham Pogers for partial revision of this thesis.

Finally, this thesis would not have been completed without understanding and support of my family, especially my parents for full of support, entirely care, love and encouragement through my life. I hope that this thesis will usefully for the processes of data mining for classification. The use fullness of this thesis. I dedicate to my father, my mother and all the teachers of me.

Yosawin Kangwanariyakul

DATA MINING OF MAGNETOCARDIOGRAMS FOR PREDICTION OF ISCHEMIC HEART DISEASE.

YOSAWIN KANGWANARIYAKUL 4837224 EGTI/M

M.Sc. (TECHNOLOGY OF INFORMATION SYSTEM MANAGEMENT)

THESIS ADVISORY COMMITTEE: THANAKORN NAENNA, Ph.D. (ENGINEERING SCIENCE), CHANIN NANTASENAMART, Ph.D. (MEDICAL TECHNOLOGY), TANAWUT TANTIMONGCOLWAT, Ph.D. (MEDICAL TECHNOLOGY)

ABSTRACT

Ischemic Heart Disease (IHD) is a major cause of death. Early detection of IHD with an accurate and quick diagnosis is important for reducing the mortality rate. Magnetocardiogram (MCG) is a tool for detecting electro-physiological activity of the myocardium. MCG is a fully non-contact method, which avoids the problems of skin-electrode contact in the Electrocardiogram (ECG) method. However, the interpretation of MCG recordings is time-consuming and requires an expert. Therefore, we propose the use of machine learning for identification of IHD patients. In an automatic classification system, the back-propagation neural network (BPNN), the Bayesian neural network (BNN), the probabilistic neural network (PNN) and the support vector machine (SVM) were applied to develop classification models for identifying IHD patients.

MCG data are acquired by sequential measurement, above the torso, of the magnetic field emitted by the myocardium using a J-T interval of 125 cases. The training and validation data of 74 cases used 10-fold cross-validation methods to optimize support vector machine and neural network parameters. The predictive performance was obtained by prediction on testing data of 51 cases and performance comparison was decided on three parameters (accuracy, sensitivity, and specificity), and area under the receiver operating characteristic (ROC) curve.

The results demonstrated that the BPNN and BNN have the same highest accuracy of 78.43%, but the decision threshold of BPNN is -0.2774, and the area of ROC curve is 0.90596. While the decision threshold and the area of ROC curve of BNN are 0.0470 and 0.84953, respectively. Therefore, the BPNN model is the best classification model. The BNN is the best with 96.65% sensitivity, and the radial basis function (RBF) kernel SVM came out to be the best with 86.36% specificity.

KEY WORDS: ISCHEMIA/ MAGNETOCARDIOGRAPHY/ DATA MINING/ BACK-PROPAGATION NEURAL NETWORK/ BAYESIAN NEURAL NETWORK/ PROBABILISTIC NEURAL NETWORK/ SUPPORT VECTOR MACHINE

72 pages

การทำเหมืองข้อมูลคลื่นแม่เหล็กหัวใจเพื่อพยากรณ์โรคกล้ามเนื้อหัวใจขาดเลือด
DATA MINING OF MAGNETOCARDIOGRAMS FOR PREDICTION OF
ISCHEMIC HEART DISEASE.

ยศวิน กังวานอริยะกุล 4837224 EGTI/M

วท.ม. (เทคโนโลยีการจัดการระบบสารสนเทศ)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : ธนกรณ์ แน่นหนา, Ph.D. (ENGINEERING SCIENCE), ชนินทร์ นันทเสนามาตร์, Ph.D. (MEDICAL TECHNOLOGY), ธนาวุฒิ ตันติมงคลวัฒน์, Ph.D. (MEDICAL TECHNOLOGY)

บทคัดย่อ

โรคกล้ามเนื้อหัวใจขาดเลือดเป็นหนึ่งในสาเหตุหลักที่ทำให้เกิดการเสียชีวิต ซึ่งถ้าผู้ป่วยได้รับการตรวจและรักษาตั้งแต่ในระยะเริ่มแรกของโรคก็จะช่วยลดอัตราการเสียชีวิตของผู้ป่วย การตรวจวัดคลื่นแม่เหล็กจากกล้ามเนื้อหัวใจมีข้อดีกว่าการตรวจวัดคลื่นไฟฟ้าหัวใจ กล่าวคือการตรวจวัดคลื่นแม่เหล็กจากกล้ามเนื้อหัวใจไม่จำเป็นต้องติดตั้งตัวรับสัญญาณบนร่างกายผู้ป่วย ซึ่งจะช่วยลดปัญหา skin-electrode contact แต่เนื่องจากการแปรผลคลื่นแม่เหล็กหัวใจจำเป็นต้องอาศัยผู้เชี่ยวชาญ และยังใช้เวลานาน ดังนั้นงานวิจัยนี้จึงได้พัฒนาระบบการเรียนรู้อัตโนมัติเพื่อพยากรณ์โรคกล้ามเนื้อหัวใจขาดเลือด โดยใช้เทคนิคการเรียนรู้แบบต่างๆ คือ back-propagation neural network (BPNN), Bayesian neural network (BNN), probabilistic neural network (PNN) และ support vector machine (SVM) ซึ่งข้อมูลคลื่นแม่เหล็กหัวใจในช่วงสัญญาณ J-T ของผู้ป่วยทั้ง 125 คน จะถูกแบ่งออกเป็นชุดข้อมูลในการเรียนรู้และชุดข้อมูลในการตรวจสอบ 74 คน เพื่อใช้ในการหาค่าพารามิเตอร์ที่เหมาะสมใน SVM และ neural network โดยใช้วิธี 10-fold cross-validation ส่วนข้อมูลของผู้ป่วยที่เหลือ 51 คน จะถูกใช้เป็นชุดข้อมูลทดสอบประสิทธิภาพของระบบที่ได้ โดยวัดจาก ค่าความแม่นยำ ค่าความไว ค่าความจำเพาะ และค่าพื้นที่ใต้กราฟ ROC

ผลการวิจัยพบว่า BPNN และ BNN มีค่าความแม่นยำ 78.43% เท่ากัน แต่ BPNN มีค่า decision threshold เท่ากับ -0.2774 และค่าพื้นที่ใต้กราฟ ROC เท่ากับ 0.90596 ในขณะที่ค่า decision threshold และ ค่าพื้นที่ใต้กราฟ ROC มีค่าเท่ากับ 0.0470 และ 0.84953 ตามลำดับ ดังนั้น BPNN จะมีประสิทธิภาพในการพยากรณ์ดีที่สุด ส่วน BNN จะมีค่าความไวสูงที่สุดคือ 96.65% และ RBF kernel SVM จะมีค่าความจำเพาะสูงสุดคือ 86.36%

72 หน้า

CONTENTS

	Page
ACKNOWLEDGEMENTS	**iii**
ABSTRACT (ENGLISH)	**iv**
ABSTRACT (THAI)	**v**
LIST OF TABLES	**viii**
LIST OF FIGURES	**ix**
CHAPTER I INTRODUCTION	**1**
1.1 Background and Statement of Problems	1
1.2 Objective	3
1.3 Scope of Work	3
1.4 Expected Result	3
CHAPTER II LITERATURE REVIEW	**4**
2.1 Magnetocardiography	4
2.2 Data Mining	6
2.3 Neural Networks	10
2.4 Binary Classification	21
2.5 Back-propagation Neural Networks	22
2.6 Bayesian Neural Networks	23
2.7 Probabilistic Neural Networks	25
2.8 Support Vector Machines	27
2.9 Ischemic Heart Disease	29
2.10 Data Mining Tools	30
2.11 Related Researches	33
CHAPTER III MATERIALS AND METHODS	**36**
3.1 Research Methodology	36
3.2 Research Tools	41
3.3 Research Schedule	42

CONTENTS (cont.)

Page

CHAPTER IV RESULTS **43**

4.1 Optimization of Neural Networks 43

4.2 Optimization of Support Vector Machines 48

4.3 Prediction of Ischemic Heart Disease 50

CHAPTER V DISCUSSION **54**

5.1 Data Collection 54

5.2 Data Mining 54

5.3 Classification models 56

5.4 Tools in the research 56

5.5 Limitations 56

CHAPTER VI CONCLUSION AND RECOMMENDATION **57**

6.1 Conclusion 57

6.2 Recommendation 58

REFERENCES **59**

APPENDIX **64**

BIOGRAPHY **72**

LIST OF TABLES

Table **Page**

3.1 Representation of confusion matrix 40

4.1 The confusion matrix of classification models. 52

4.2 The prediction performance of classifier 53

LIST OF FIGURES

Figures **Page**

2.1 Data mining as the core of knowledge discovery process 6

2.2 Structure of typical neuron 10

2.3 The neuron model 11

2.4 The processing elements 13

2.5 Threshold Function 13

2.6 Piecewise-Linear Function 14

2.7 Sigmoid Function 15

2.8 Gaussian Function 15

2.9 The interconnection schemes 16

2.10 Block diagram of Supervised learning 18

2.11 Block diagram of Unsupervised learning 20

2.12 Block diagram of Reinforcement learning 20

2.13 Configuration of Bayesian Neural Network Classifier 23

2.14 Network Architecture of Probabilistic Neural Networks 26

2.15 The objects of a linear separable support vector machine 28

2.16 The mapping of non-linear classification into feature space 29

2.17 The cause of Ischemic Heart Disease 30

3.1 Phases of the CRISP-DM Process Model 36

3.2 Flow diagram of the design and development process of neural network 38

4.1 The plotting of average RMSE versus hidden nodes of BPNN 43

4.2 The plotting a graph RMSE versus the number of learning epoch of BPNN 44

4.3 The error surface is the optimal learning rate and momentum of BPNN 45

4.4 The plotting of average RMSE versus hidden nodes of BNN 46

4.5 The plotting a graph RMSE versus the number of learning epoch of BNN 46

4.6 The error surface is the optimal learning rate and momentum of BNN 47

4.7 The Plot of correct classification with varying spread values 48

LIST OF FIGURES (cont.)

Figures **Page**

4.8 The plotting of average RMSE versus C of linear kernel SVM 49

4.9 The plotting of average RMSE versus C of polynomial kernel SVM 49

4.10 The error surface of the optimal C and gamma of RBF kernel SVM 50

4.11 ROC curve showing possible trade-offs between true positives (sensitivity) and false positives (1-specificity) 51

CHAPTER I
INTRODUCTION

1.1 Background and Statement of Problems

Ischemic heart disease (IHD) is often found to be a major cause of death in people. By the year 1990 to 2020, World Health Organization estimates that the global number of death, caused by cardiovascular disease, especially in developing countries have increased from 14 million to 25 million people. Furthermore, mortality rate of cardiovascular disease in Thailand is found that top three of cause of death from year 2002 number of death people have been 32,903 people and increase to 40,092 peoples in year 2003. Therefore, early detection of IHD with accuracy and quickly diagnosis is important for reduce mortality rate [1].

Ischemic heart disease is a problem due to inadequate circulation of blood supply to the heart muscle. Blood flow to the heart is blocked by occlusion of heart arteries by cholesterol deposits. This plaque leads to narrowing of blood supply to the heart muscle causes to lack of oxygen thus heart muscle may be damaged or die. Ischemic heart disease is also called coronary artery disease or coronary heart disease.

The ischemia condition is caused by occluded coronary arteries. It causes a change in electro-physiological activity in myocardium. Electrocardiogram (ECG) is a convenient tool for this diagnosis. It measures electro-physiological of heart by use electrodes attached on skin of body. Electro-physiological phenomena in the heart can be detected not only by ECG but also magnetocardiography (MCG).

Magnetocardiography (MCG) has been developed for detect and diagnose IHD, MCG is sensitive technique that use for measurement of magnetic fields emitted by the electrophysiological activity of the human heart. The magnetic field is passively recorded and is conducted as a completely noninvasive procedure without any contact to the body, this is advantage of MCG above ECG. The magnetic field is recorded using superconducting quantum interference devices (SQUIDs). The MCG is more sensitive to tangential currents in the heart than the ECG, and it is also sensitive to vortex current, which cannot be detected by the ECG. In the normal heart, the main direction of the activation waveform is radial, from endocardium to epicardium. For these reasons, MCG may show ischemia-induced deviations from the normal direction of depolarization and repolarization with better accuracy than the ECG. MCG is affected less by conductivity variations in the body (lungs, muscles, and skin) than ECG. In addition, because MCG is a fully non-contact method, the problems in the skin-electrode contact encountered in ECG are avoided.

Nowadays, medical informatics has increasingly been stored in large databases as a result of the increased use of powerful hardware and software computers with automated tools. but the interpretation of MCG recordings remains a challenge since they are no databases available from which precise rules could be deduced. And the analysis of MCG data by expert people spends a lot of time and has a little people who have skill about analysis of MCG data. Thus the methods to automate interpretation of MCG recordings to minimize human input for analysis are important for diagnosis in IHD patients.

Data mining is the process of automatically discovering useful information in the large volume of data that known as Knowledge Discovery in Databases (KDD). The KDD is the process of converting raw data into useful information which process consists of a series of transformation step, data selecting, pre-processing, transforming and mining of data [2, 3].

In this study, we used the three different types of neural network and three different types of support vector machine. The neural network classification models are back-propagation neural network, Bayesian neural network, and probabilistic neural network. The support vector machine classification models are linear kernel SVM, polynomial kernel SVM, and radial basis function kernel SVM. All models were training with a 10-fold cross-validation technique to optimize neural network parameter for optimal neural network model to compare the efficiency and accuracy of these classification models.

1.2 Objectives

1.2.1 To study the data mining techniques to extract knowledge from Magnetocardiograms data for development of automatic detection method for ischemic heart disease using three models of neural network (back-propagation, Bayesian and probabilistic algorithms) and three models of support vector machine (linear kernel, polynomial kernel and radial basis function kernel).

1.2.2 To compare a performance of neural network classification models and support vector machine classification models.

1.3 Scopes of Work

1.3.1 The 125 samples (training sample 74 cases, testing sample 51 cases) MCG data were acquired at 36 locations above the torso by making four sequential measurements in mutually adjacent position.

1.3.2 The data mining task was implemented by Mathworks Matlab and the Weka.

1.3.3 The accuracy and performance of classify is evaluated by compare in three algorithms of neural network (Back-propagation, Bayesian and Probabilistic algorithms) and three kernels of support vector machine (linear kernel, polynomial kernel and radial basis function kernel) by using testing sample 51 cases.

1.4 Expected Results

1.4.1 Development of method for MCG classification of ischemic heart disease using automated machine learning.

1.4.2 Selection of suitable neural network algorithm and support vector machine for MCG classification.

CHAPTER II
LITERATURE REVIEW

2.1 Magnetocardiography

Magnetocardiography (MCG) is a method for investigating electro-physiological processes in the heart while analyzing parameters of its magnetic field recorded by non-invasive (and non-contact) way at observation points in the air (usually above a patient's chest) [4]. History of the magnetocardiography takes its origin in 1963, when Baule and McFee made first measurements of parameters of human heart magnetic field in Syracuse University [5, 6].

Since that time the MCG research method has covered a significant way in its evolution and still continues its dynamic development nowadays. The MCG examination is aimed at analysis of electrical activity in the myocardium. The same purpose is inherent in electrocardiography (ECG). Since magnetocardiography and electrocardiography are based on the same electrophysiological sources, the data they bear complement each other and are interrelated. A new, more informative content of MCG is associated with its extreme sensitivity to tangential components of the heart excitation wave and less (compared to ECG) dependence of magnetic field measured parameters at an observation point upon effect of multi-layered conducting medium containing the field source. Currently many experimental and clinical works have proved much higher spatial resolution of magnetocardiography as compared to electrocardiography, i.e. its much higher sensitivity to local currents. These currents, usually low in magnitude, emerge at interfaces of myocardium portions having different electro-physical properties, in other words, with different duration of action potentials of their cells. Such currents are much more legible in magnetic field than in electric one. This advantage becomes especially obvious when there are two excitation fronts, close in magnitude and opposite in direction. In this case they compensate for each other in electric field, but in magnetic field they will be explicitly displayed. Keeping this in mind, Baule's and McFee's comment on MCG method becomes clear, that "magnetocardiography allows to reveal such components of heart electro-motive force which otherwise would stay silent" [5, 6]. Sensitivity to local currents allows to assess homogeneity of myocardium in electrical terms.

A conclusion can be made that magnetocardiography possesses easiness and safety for patients inherent in non-invasive methods, along with carefulness and accuracy of direct methods of myocardium electrophysiology. It should be recognized that such an advantageous combination has to be "reimbursed for" by some technical complexities. Magnetic field of the Earth is on the order of 10^{-4} T, and magnetic field of human heart makes up tens of picoTesla ($(10\text{-}50) \times 10^{-12}$ T). It is clear that to record such a weak field an extremely sensitive sensor is needed. Such a sensor is a superconductive quantum interferometer (SQUID) [7] operating at low temperatures, for that it must be placed into liquid helium (or liquid nitrogen). Besides, special technical methods shall be used to reduce effect of external noise on MCG signal. These all makes the MCG systems a little bit more expensive and requirements to the level of maintenance personnel higher.

During few years already this information has been used by our medical colleagues and has been helping to solve real diagnostic problems of cardiological ward.

Main tasks and clinical situations resolutions of which use MCG are the following [10]:

1. MCG is applied for localization of the additional conducting pathways (ACP) and determining effectiveness of radio frequency ablation of these pathways. Preliminary out-surgery non-invasive ACP locating allows to reduce time for a surgery.

2. Determining presence and degree of myocardium ischemia, including its early stage. Clinical research shows that MCG in rest possesses high sensitivity to chronical ischemia even among patients with stable or slightly changed ECG in rest. MCG examination allows to solve the issue of whether there is an indication to medication antianginal therapy, specify indications to coronaroangiography.

3. Determining effectiveness of antianginal therapy, including period after cardiac infarction. The examination helps to address the issue of dose change or change in antianginal medicine.

4. Determining risk of paroxism of ventricular tachycardia. The examination allows to solve a question of antiarythmic therapy, specify indications for invasive EPE.

5. Determining nature of a disease resulted in heart size increase (cardio megalium). The examination allows in controversial cases to solve the issue of main disease, and this effects tactics of treatment.

2.2 Data Mining

Data mining, also known as knowledge discovery from data (KDD), is a process that uses statistical, mathematical, artificial intelligence and machine-learning techniques to identify and extract high-level knowledge from a large amount of data. While data mining and knowledge discovery from data (KDD) are frequently treated as synonyms, data mining is actually a part of the knowledge discovery process. The Figure 2.1 shows data mining as a step in an iterative knowledge discovery process [8].

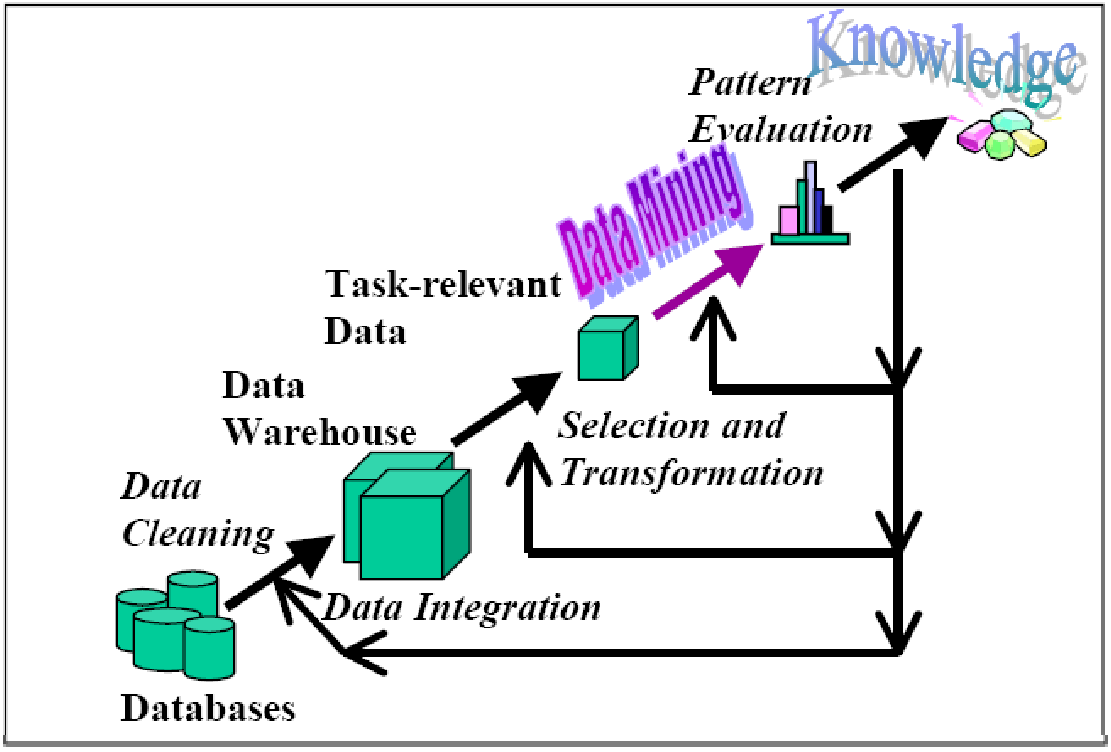

Figure 2.1 Data mining as the core of knowledge discovery process [8]

The KDD process comprises a number of steps ranging from the collection of raw data to the extraction of new knowledge. The iterative process consists of the following steps :

- **Data cleaning :** also known as data cleansing. It is a phase in which noises and irrelevant data are removed from the collection.
- **Data integration :** In this step, multiple data sources, often heterogeneous, may be combined into a single common source.
- **Data selection :** In this step, the data relevant to the analysis is identified and retrieved from the data collection.
- **Data transformation :** also known as data consolidation. It is a phase in which the selected data is transformed into the form appropriate for the mining procedure.
- **Data mining :** It is the crucial step in which a number of techniques are applied to extract potentially useful patterns.
- **Pattern evaluation :** In this step, patterns representing interesting knowledge are identified based on given measures.
- **Knowledge representation :** It is the final phase in which the discovered knowledge is visually presented to users. This step uses visualization techniques to help the users understand and interpret the results.

It is common to combine some of these steps together. For instance, data cleaning and data integration can be performed together as a pre-processing step which constructs a data warehouse. Data selection and data transformation can also be combined as a single step, or, in the case of data warehouses, the selection is done on the transformed data.

The KDD is an iterative process. Once the discovered knowledge is presented to the users, the evaluation measures can be enhanced, the mining can be further refined, the new data can be further transformed, or additional data sources can be integrated – these are performed in order to get different or more appropriate results.

Data mining derives its name from the similarities between searching for valuable information in a large database and mining rocks for a vein of valuable ore. Both imply either sifting through a large amount of material or ingeniously probing the material to exactly pinpoint where the values reside. It is, however, a misnomer, since mining for gold in rocks is usually called “gold mining” and not “rock mining”, thus by analogy, data mining should have been called “knowledge mining” instead. Nevertheless, data mining became the accepted customary term, and very rapidly a trend that even overshadowed more general terms such as knowledge discovery from data (KDD) that describe a more complete process. Other similar terms referring to data mining are: data dredging, knowledge extraction, and pattern discovery.

The term “data mining” was formerly used to describe the process through which undiscovered patterns in data were identified. However, the original definition has been modified to include most types of (automated) data analysis. It now refers to the process of engineering mathematical patterns from usually large sets of data. These patterns can be rules, trends, interrelationship, prediction models, or new and meaningful information [8, 28].

Data mining tasks include knowledge extraction, data exploration, data pattern processing, data dredging, and information harvesting. These are conducted automatically and allow quick discovery even by non-programmers.

There are three classes of data mining techniques to identify patterns and infer rules from data :

- **Simple models** (SQL-base query, OLAP, human judgment)
- **Intermediate models** (regression, decision trees, clustering)
- **Complex models** (neural networks, other rule induction)

These patterns and rules can be used to guide decision-making and forecast the effects of the decisions. Another approach to classify the data mining techniques, as described below, depends on the types of problems they are able to solve :

- **Classification :** infers the defining characteristics of a certain group. It involves seeding a set of data with a known set of classes (probably found by clustering), and mapping all other items into these sets. Decision trees and neural networks are useful techniques.

- **Clustering :** identifies groups of items that share some certain characteristics (clustering differs from classification in that no predefining characteristic is given). This approach addresses segmentation problems. Clustering algorithms can be used, for example, to identify classes of customers with certain needs to be met.
- **Association :** identifies relationships between events that occur at one time. This approach addresses a class of problems typified by market basket analysis. In retailing, there is an attempt to identify which products are sold with which other ones, and to what degree. Statistical methods are typically used.
- **Sequencing :** is similar to association, except that the relationship occurs over a period of time. For example, purchasing activities can be tracked by credit card usage of the customers, or by some other means.
- **Regression :** maps data to a prediction value. Linear and nonlinear techniques are used. This is a form of estimation. It often involves identifying metrics and evaluating an item along the metrics by assigning scores. For example, sales prediction can be made using regression techniques.
- **Forecasting :** estimates future values based on patterns within large sets of data. This is another form of estimation. Statistical time-series methods can also be used to predict future sales.
- **Other techniques :** are typically based on advanced artificial intelligence methods such as case-based reasoning, fuzzy logic, genetic algorithms, and fractal-based transforms.

2.3 Neural Networks

2.3.1 Biological Neurons

The study of artificial neural networks (ANNs) has been inspired in part by the observation that biological learning systems are built of very complex webs of interconnected neurons. Figure 2.2 shows schematic of biological neuron.

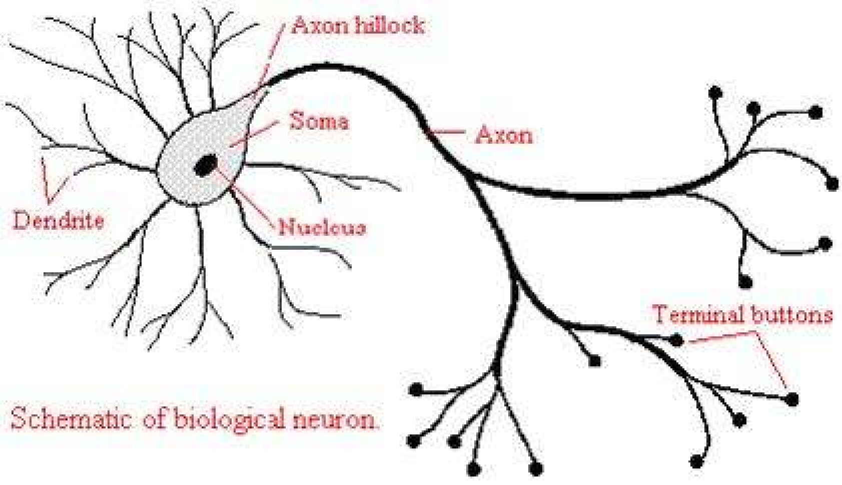

Figure 2.2 Structure of a typical neuron.

In general any typical neuron has 4 morphologically defined regions:

1) Soma is the cell body which includes the nucleus perikaryon.

2) Dendrites which serve an input role. A typical neuron has several dendrites, normally arranged in an extremely branched fashion in order to establish contacts with many other neurons.

3) Axon, representing the conduction component of the neuron. Every neuron has only a single axon.

4) The synaptic terminals, these are distinct structures located at the every end of an axon.

By deduction the essential features of neurons and their interconnections can simulate simple models for biological neurons, shows in Figure 2.3

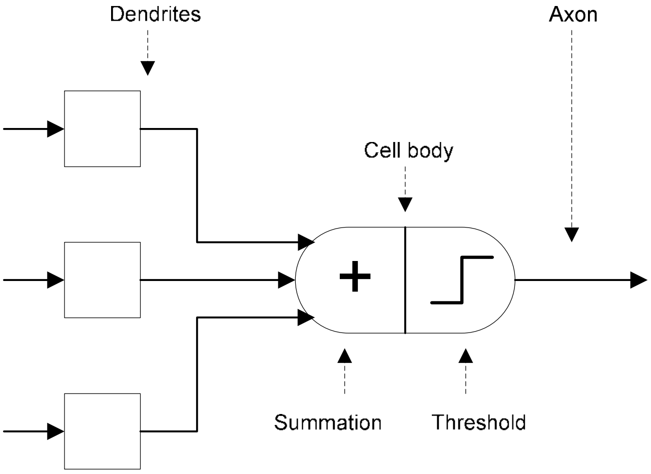

Figure 2.3 The neuron model.

2.3.2 Artificial Neural Networks

The branch of artificial intelligence neural networks dates back to the 1940's, when McCulloch and Pitts [1943] developed the first neural model. This was followed in 1962 by the perceptron model, devised by Rosenblatt, which generated much interest because of its ability to solve some simple pattern classification problems. This interest started to fade in 1969 when Minsky and Papert [1969] provided mathematical proofs of the limitations of the perceptron and pointed out its weakness in computation. In particular, it is incapable of solving the classic exclusive-or (XOR) problem, which will be discussed later. Such drawbacks led to the temporary decline of the field of neural networks.

Artificial Neural Network is a system loosely modeled based on the human brain. The field goes by many names, such as connectionism, parallel distributed processing, neuro-computing, natural intelligent systems, machine learning algorithms, and artificial neural networks. It is an inherently multiprocessor-friendly architecture and without much modification, it goes beyond one or even two processors of the von Neumann architecture. It has ability to account for any functional dependency. The network discovers (learns, models) the nature of the dependency without needing to be prompted. No need to postulate a model, to amend it, etc.

Neural networks are a powerful technique to solve many real world problems. They have the ability to learn from experience in order to improve their performance and to adapt themselves to changes in the environment. In addition to that they are able to deal with incomplete information or noisy data and can be very effective especially in situations where it is not possible to define the rules or steps that lead to the solution of a problem.

An ANN can transform and represent complex and highly non-linear relationship automatically. It can also automatically detect different state of phenomena through independently variable data patterns and switch on/off model components as appropriate. Hence, the ANNs are appropriate for complex phenomena. Moreover, ANNs are ideally suited for prediction and forecasting because of their self-adaptive, automatic modeling properties [9].

2.3.3 Components of an Artificial Neuron

1) Processing elements (PEs)

Each unit performs a relatively basic function such as receive input from neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherently parallel in the sense that many units can carry out their computations at the same time. Within neural systems processing elements are useful to distinguish three types of units: input units which receive data from outside the neural network, output units which send data out of the neural network, and hidden units whose input and output signals remain within the neural network. During operation, units can be updated either synchronously or asynchronously. The three types of PEs are shown in Figure 2.4.

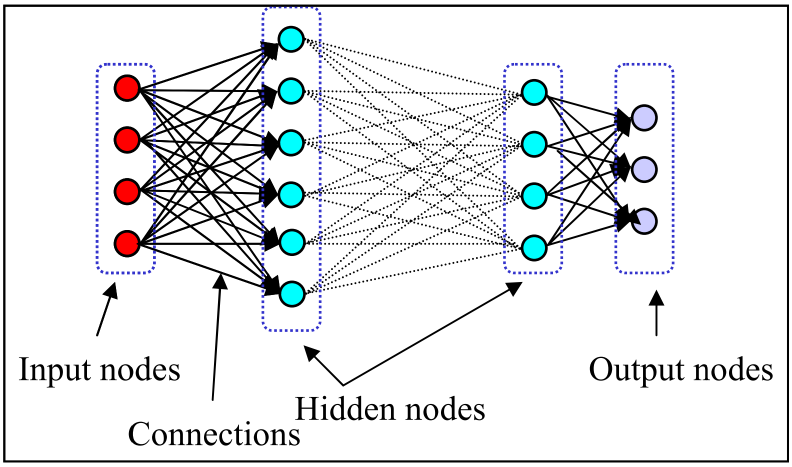

Figure 2.4 The Processing Elements.

2) Activation function

This is the mechanism of translating input signals to an output signal for each PE. The activation function can be identified four types:

2.1) Threshold Function

For this type of activation function describe in equation 2.1

$$\varphi(v) = \begin{cases} 1 & if \quad v \geq 0 \\ 0 & if \quad v < 0 \end{cases} \tag{2.1}$$

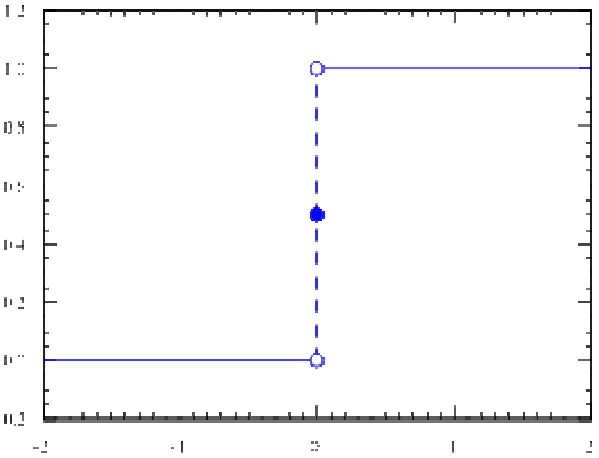

Figure 2.5 Threshold Function.

2.2) Piecewise-Linear Function

For the piecewise-linear function described in equation 2.2

$$\varphi(v)=\begin{cases}1, & v\geq+\tfrac{1}{2}\\ v, & +\tfrac{1}{2}>v>-\tfrac{1}{2}\\ 0, & v\leq-\tfrac{1}{2}\end{cases}\qquad(2.2)$$

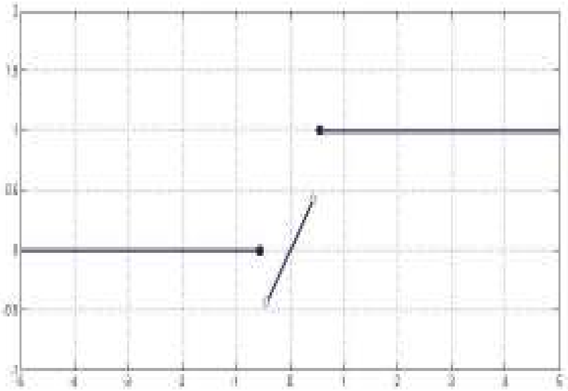

Figure 2.6 Piecewise-Linear Function.

2.3) Sigmoid Function

The sigmoid function, whose graph is s-shaped, is by far the most common form of activation function used in the artificial neural networks. It is defined as a strictly increasing function that exhibits a graceful balance between linear and nonlinear behavior. An example of the sigmoid function is the logistic function, defined by equation 2.3.

$$\varphi(v)=\frac{1}{1+e^{-av}}\qquad(2.3)$$

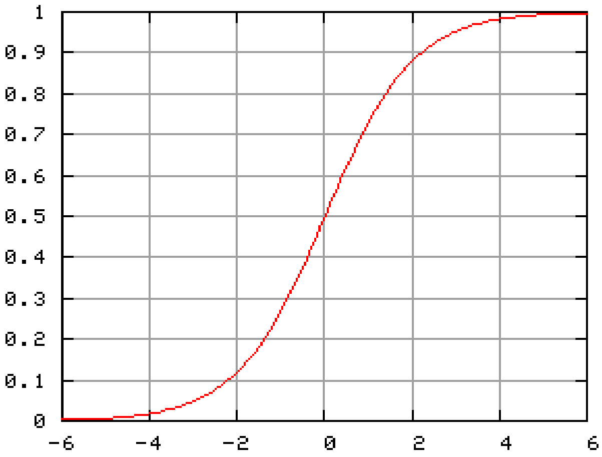

Figure 2.7 Sigmoid Function.

Where a is the slope parameter of the sigmoid function

2.4) Gaussian Function

For the Gaussian function described in equation 2.4

$$\varphi(v) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(v-\mu)^2}{2\sigma^2}} \qquad (2.4)$$

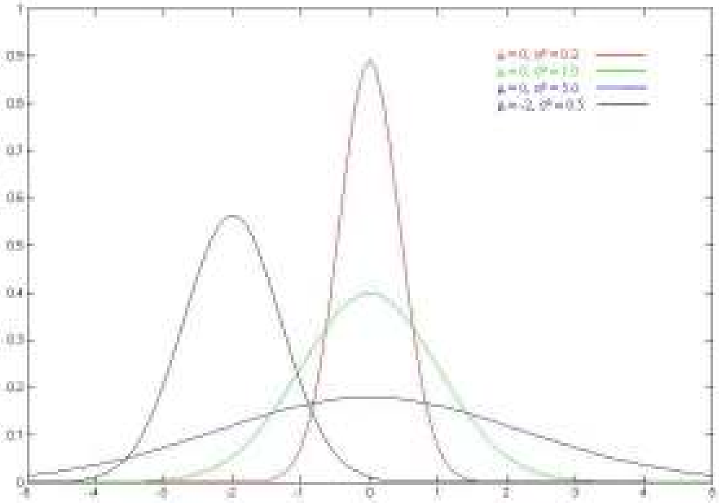

Figure 2.8 Gaussian Function.

Where μ is the mean

σ is the standard deviation

e is the base of the natural logarithm, sometime called Euler's e (2.71…)

π is the constant Pi (3.14…)

3) Interconnection

The interconnection scheme is a part of the network architecture definition, which propagates a signal from one PE to another, or even to a PE itself. It is unidirectional and there is a weight value assigned to each connection, these weight values forming the memory of the network [10].

There are three different connection schemes that connect PEs in a network.

- Intrafield connections connect PEs in the same layer.
- Interfield connections connect between PEs in different layers.
- Recurrent connections loop and connect back to the PE itself.

If the information flows in one direction, the interconnections are called feed forward. Feedback lets the information flow among PEs in either direction and/or recursively. The interconnection schemes are shown in figure 2.9.

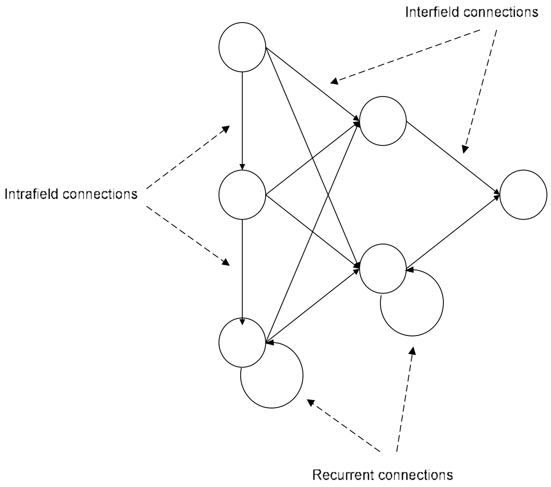

Figure 2.9 The interconnection schemes.

The neural network is typically composed of several layers of nodes. The first or the lowest layer is an input layer where the set of neuron whose input signals are from the environment only. The number of neurons in the input layer depends on the characteristics of the input data. There is only one input layer in the neural networks. The last or the highest layer is an output layer where the set of neurons whose output signals are not transferred to other neurons but only to the environment. The number of neurons in the output layer depends on the required output set, but usually less than those of the input layer. The input layer and output layer are separated by one or more intermediate layers, called that the hidden layers. The set of neurons in the hidden layer receive and pass signals to other neurons, but no interface with the environment. The number of neurons in the hidden layer differs from problem to problem. There can be more than one hidden layer in the neural network, through there is no rule to determine the number of appropriate hidden layer in the networks.

4) Learning rules

The output of a network, given its input, depends on the connection structure and weights. In general, the connection structure is held constant and the weights are modified to allow the network to implement different functions. How the weights are modified depends on the objective of the network and the information available to the learning rule. There are two main learning paradigms [11]:

4.1) Learning with a teacher

- Supervised Learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until an acceptable network accuracy is reached.

With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. In most applications, actual data must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches an user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application. Some network types allow continual training, at a much slower rate, while in operation. This helps a network to adapt to gradually changing conditions.

Training sets need to be fairly large to contain all the needed information if the network is to learn the features and relationships that are important. Not only do the sets have to be large but the training sessions must include a wide variety of data. If the network is trained just one example at a time, all the weights set so meticulously for one fact could be drastically altered in learning the next fact. The previous facts could be forgotten in learning something new. As a result, the system has to learn everything together, finding the best weight settings for the total set of facts. For example, in teaching a system to recognize pixel patterns for the ten digits, if there were twenty examples of each digit, all the examples of the digit seven should not be presented at the same time.

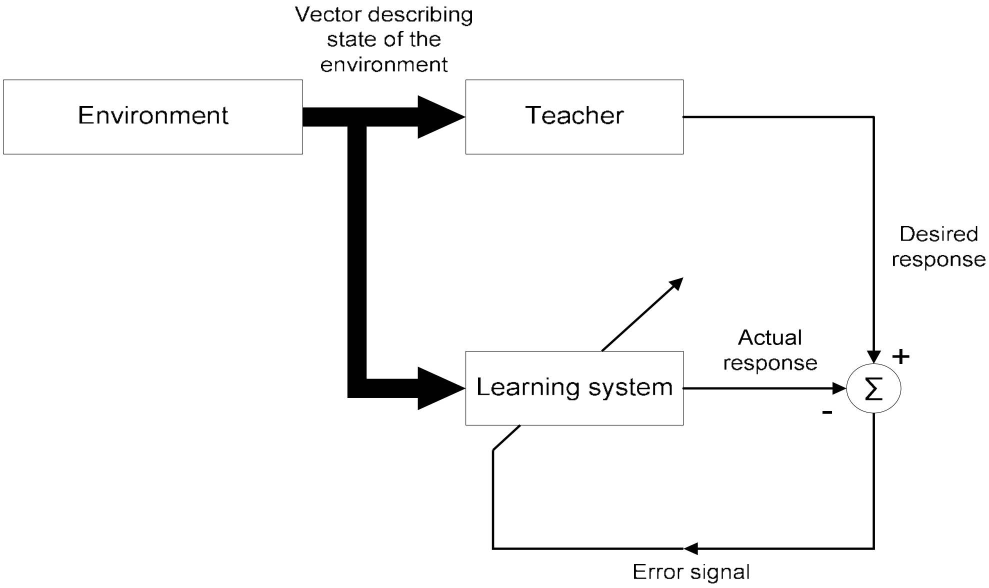

Figure 2.10 Block diagram of supervised learning [11].

How the input and output data is represented, or encoded, is a major component to successfully instructing a network. Artificial networks only deal with numeric input data. Therefore, the raw data must often be converted from the external environment. Additionally, it is usually necessary to scale the data, or normalize it to the network's paradigm. This pre-processing of real-world stimuli, be they cameras or sensors, into machine readable format is already common for standard computers. Many conditioning techniques which directly apply to artificial neural network implementations are readily available. It is then up to the network designer to find the best data format and matching network architecture for a given application.

After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application.

4.2) Learning without a teacher

In supervised learning, the learning process takes place under the tutelage of a teacher. However, in the paradigm known as learning without teacher, as the name implies, there is no teacher to oversee the learning process. That is to say, there are no labeled examples of the function to be learned by the network. Under this second paradigm, two subdivisions are identified:

- Unsupervised Learning

In unsupervised or self-organized learning there is no external teacher or critic to oversee the learning process, as indicated in Figure 2.11. Rather, provision is made for a task independent measure of the quality of representation that the network is required to learn, and the free parameters of the network are optimized with respect to that measure. Once the network has become tuned to the statistical regularities of the input data, it develops the ability to form internal representations for encoding features of the input and thereby to create new classes automatically.

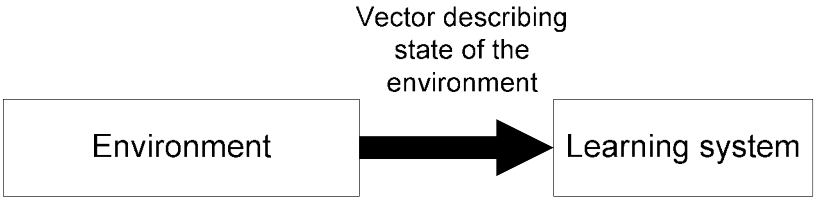

Figure 2.11 Block diagram of unsupervised learning [11].

- Reinforcement Learning

In reinforcement learning, the learning of an input-output mapping is performed through continued interaction with the environment in order to minimize a scalar index of performance. Figure 2.12 shows the block diagram of one form of a reinforcement learning system built around a critic that converts a primary reinforcement signal received from the environment into a higher quality reinforcement signal called the heuristic reinforcement signal, both of which are scalar input.

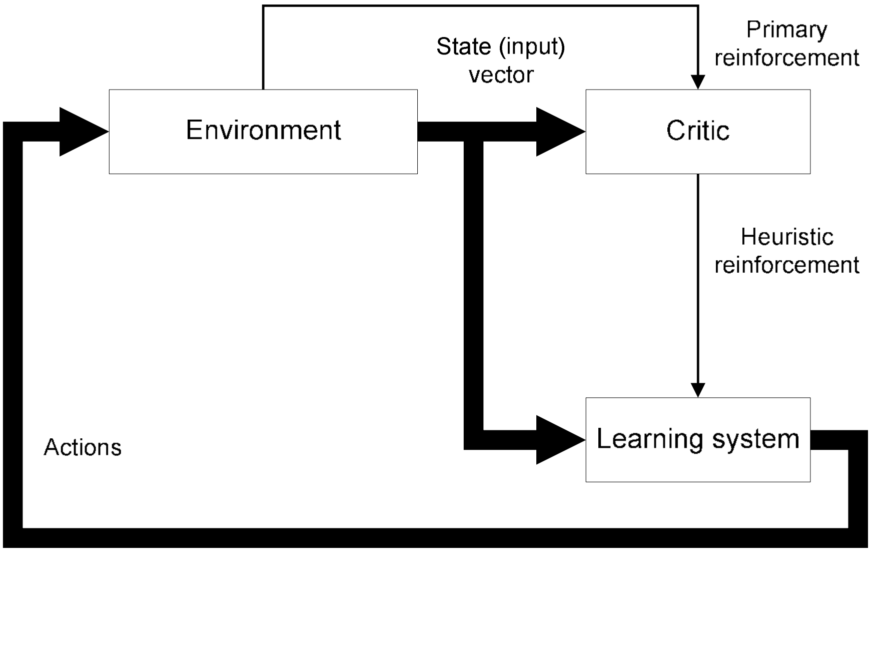

Figure 2.12 Block diagram of reinforcement learning [11].

2.4 Binary Classification

Classification is one of the most data mining tasks. There are many problems can be classified as classification problems such as face recognition, handwritten recognition, speed recognition, weather forecast, medical diagnosis, and quality control. A binary classification is the task of classifying the objects into two groups.

Data mining is an important phase in the knowledge discovery process that has a number of analytical techniques. There are many analytical techniques that can be applied to classification tasks. Examples such as decision tree, artificial neural network (ANN), logistic regression, k-nearest neighbours (kNN), Naïve Bayes, support vector machine (SVM), K-means, fuzzy logic, genetic algorithm [49].

ANN and SVM have been applied for many biomedical prediction and classification tasks. Mobley et al. [44] were developed neural network models for automatic detection of coronary artery stenosis using 14 risk factors for coronary artery disease. Chou et al. [50] used integrating artificial neural networks with the multivariate adaptive regression splines approach to classification breast cancer. Liew et al compared the predictive accuracy of artificial neural network and conventional logistic regression for prediction of gallbladder disease in obese patients. In addition, neural network was applied to the prediction of acute coronary syndrome compare with multiple logistic regression [51]. Wang et al. [52] had developed three models (ANN, random forests, and SVM) for prediction virological response to therapy from HIV genotype and other clinical information and comparing accuracy of prediction. Amendolia et al. [53] had applied three data mining techniques, K-nearest neighbour, support vector machine, and multi-layer perceptron to compare the performance for Thalassemia screening.

From above listed research, the predictive accuracy of ANN and SVM illustrates good performance for classification tasks and ranking the top ten data mining algorithms identified by the IEEE International Conference on Data Mining (ICDM) [54]. Therefore, the aim of this study was to compare the predictive accuracy of three algorithms of neural network for prediction of ischemic heart disease from magnetocardiograms.

2.5 Back-propagation Neural Networks

One of the most commonly used supervised ANN model is back-propagation network that uses back-propagation learning algorithm. Back-propagation algorithm is one of the well-known algorithms in neural networks. The introduction of back-propagation algorithm has overcome the drawback of previous neural network algorithm in 1970's where single layer perceptron fail to solve a simple XOR problem [11]. The back-propagation neural network is essentially a network of simple processing elements working together to produce a complex output. These elements or nodes are arranged into different layer that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation nodes. The input signal propagates through the network in a forward direction.

Basically, error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass.

In the forward pass, an input vector is applied to the sensory nodes of the network, and its effect propagates through the network layer by layer. Finally, a set of outputs is produced as the actual response of the network. During the forward pass the synaptic weights of the networks are all fixed.

During the backward pass, the synaptic weights are all adjusted in accordance with an error-correction rule. Specifically, the actual response of the network is subtracted from a target response and multiplying the difference by the derivative of the activation function, as in equation 2.5 [12].

$$\delta_j = (T_j - A_j)\varphi'_j(s) \qquad (2.5)$$

Where δ_j = error signal

T_j = target response

A_j = actual response

$\varphi'_j(s)$ = derivative of activation function respectively.

After the error has been calculated, the synaptic weight of the network are adjusted by using the least mean squared (LMS) learning rule. The new synaptic weight can be calculated using equation 2.6 [12].

$$\Delta w_{ji}(n) = \eta \delta_j x_{ji} + \alpha \Delta w_{ji}(n-1) \qquad (2.6)$$

Where η = learning-rate

α = momentum constant

δ_j = error signal

n = iteration or cycle number

ω = weight value

2.6 Bayesian Neural Networks

Bayesian Artificial Neural Network (ANN) Classifier was implemented on ECG Arrhythmia diagnostic system by Dayong Gao for detection of ECG arrhythmias [13]. ANN Classifier was built by the use of a logistic regression model and the back propagation algorithm based on a Bayesian framework. The Bayesian ANN's configuration was shown schematically in Figure. 2.13.

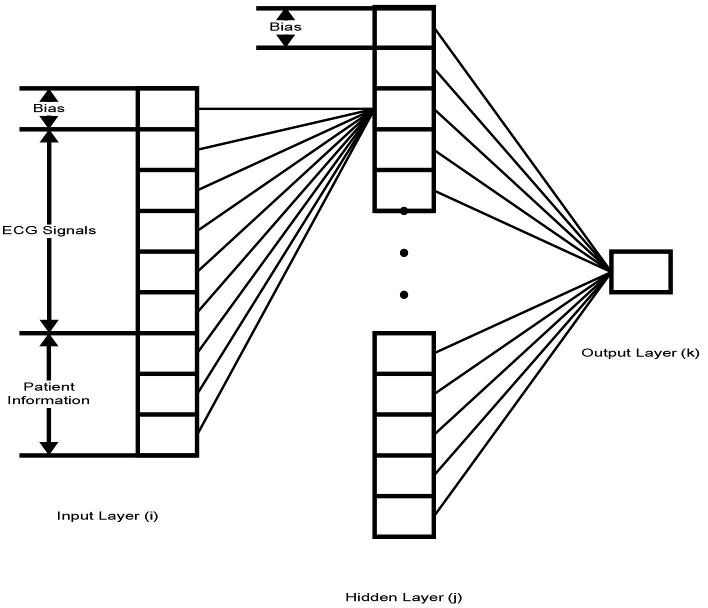

Figure 2.13 Configuration of Bayesian Neural Network Classifier [13].

Where $i = 1, 2, \ldots, I$ is the number of input nodes.

$j = 1, 2, \ldots, J$ is the hidden nodes.

$k = 1, 2, \ldots, K$ is the class labels of the output node.

Assume that a training set D, consisting of N input-output pairs:

$$D = [(\boldsymbol{X}^n, y^n) \mid n = 1, 2, \ldots, N] \qquad (2.7)$$

where $\boldsymbol{X}$ is an input vector consisting of I elements and y is the corresponding class label consisting of K classes. The objective is to use an ANN to model the input-output relation ($y = k \mid \boldsymbol{X}$). Here, the class label is binary-valued $y = (1,0)$, corresponding to Normal (no arrhythmia) and Abnormal (arrhythmia), respectively.

An alternative target for our task is to use a logistic regression model based on a Bayesian method, and estimate the class probability for a given input by:

$$P(y = k \mid \boldsymbol{X}) \qquad k = 1, 0 \qquad (2.8)$$

The outputs for the summation operation and sigmoid activation function in the hidden and output neurons, denoted by S_j and S_k, respectively, can be written as follows:

Hidden layer:

$$S_j = \tanh\left(\sum_i \omega_{ji} X_i + \omega_{j0}\right) \qquad (2.9)$$

Output layer:

$$S_k = \sum_k \omega_{kj} S_j + \omega_{k0} \qquad (2.10)$$

where tanh is the tangent hyperbolic function, a conventional sigmoid function. Here, ω_{ji} denotes the weight matrix in the input layer and ω_{kj} denotes the weight matrix in the output layer.

To allow the outputs to be interpreted as probabilities, logistic regression is used to model the risk (or probability) of occurrence of arrhythmia. Let P ($y = k \mid \boldsymbol{X}$) be the probability of the event $y = 1$, given the input vector $\boldsymbol{X}$. This is modeled as a function of network output y by:

$$P(y = k \mid X) = \frac{1}{1 + \exp(-S_k)} \qquad (2.11)$$

The logistic regression model is simply a non-linear transformation of the linear regression. The logistic distribution is an S-shaped distribution function which is similar to the standard-normal distribution but easier to work with in most applications. It constrains the estimated probabilities to lie between 0 and 1.

The system is a multi-layer perceptron neural network, trained using the back-propagation algorithm. The network is optimized using a log-likelihood cost function, given by

$$C(w) = -\frac{1}{k}\sum_k \sum_i \; y_i(k)\ln\left[P(y = k|X)\right] \qquad (2.12)$$

To minimize the cost function between the actual and desired outputs of the network, the back-propagation algorithm passes information from the output neuron backwards to all hidden units to form error terms which are used to update the weights of the multi-layer network.

2.7 Probabilistic Neural Networks

Probabilistic neural networks can be used for classification problems. When an input is presented, the first layer computes distances from the input vector to the training input vectors and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities [15].

Finally, a complete transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes. The architecture for this system is shown in Figure 2.14.

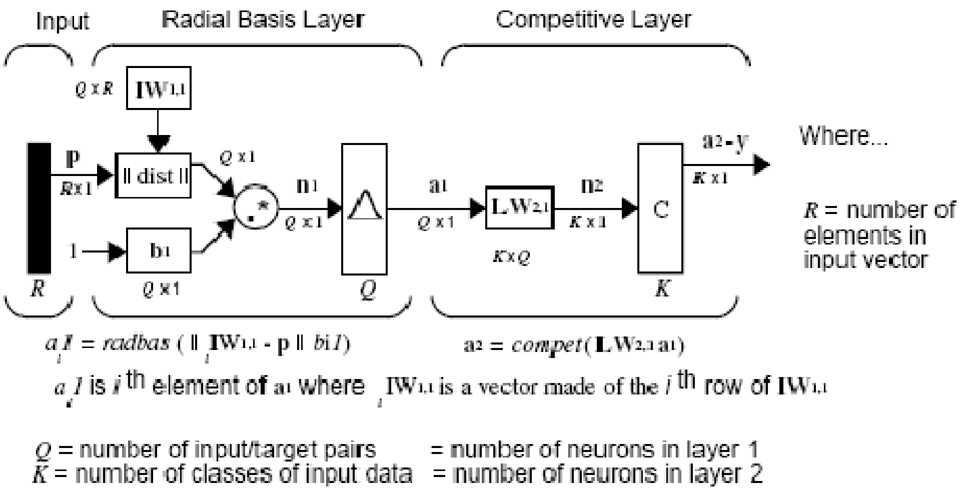
Input
Radial Basis Layer
Competitive Layer
Q x R
IW1,1
p
R x 1
|| dist ||
Q x 1
1
b1
Q x 1
n1
Q x 1
.*
a1
Q x 1
LW2,1
K x Q
n2
K x 1
C
a2 = y
K x 1
R
Q
K
Where...
R = number of elements in input vector
a_i1 = radbas (|| iIW1,1 - p || bi1)
a2 = compet(LW2,1 a1)
a_i1 is i th element of a1 where iIW1,1 is a vector made of the i th row of IW1,1
Q = number of input/target pairs = number of neurons in layer 1
K = number of classes of input data = number of neurons in layer 2

Figure 2.14 Network Architecture of Probabilistic Neural Networks [14]

The PNN has three layers: the Input Layer, Radial Basis Layer and the Competitive Layer. Radial Basis Layer evaluates vectors distances between input vector and row weight vectors in weight matrix. These distances are scaled by Radial Basis Function nonlinearly. Then the Competitive Layer finds the shortest distance among them, and thus finds the training pattern closest to the input pattern based on their distance.

The network structure scheme is illustrated in Figure 2.14. Dimensions of arrays are marked under their name [14].

1) Input Layer: The input vector, denote as **p**, is presented as the black vertical bar in Figure 2.14. Its dimension is Rx1.
2) Radial Basis Layer: In Radial Basis Layer, the vector distances between input vector **p** and the weight vector made of each row of weight matrix **IW** are calculated. Here, the vector distance is defined as the dot product between two vectors. Assume the dimension of **IW** is QxR. The dot product between **p** and i-th row of **IW** produces the i-th element of the distance vector || **dist** ||, whose dimension is Qx1, as shown in Figure 2.14. The minus symbol, "-", indicates that it is the distance between vectors.

Then, the bias vector **b** is combined with || **dist** || by an element-by element multiplication, represented as ".* " in Figure 2.14. The result is denoted as **n1** = || **dist** ||.* **p**.

The transfers function in PNN has built into a distance criterion with respect to a center. In this paper is defined as

$$radbas\ (n) = e^{-n^2} \qquad (2.13)$$

Each element of **n1** is substituted into equation 2.13 and produces corresponding element of **a1**, the output vector of Radial Basis Layer can represent the i-th element of **a1** as

$$a1_i = radbas(\|\ dis\ \|\ .*\ b_i) \qquad (2.14)$$

3) Some characteristics of Radial Basis Layer: The i-th element of **a1** equals to 1 if the input **p** is identical to i-th row of input weight matrix **W**. A radial basis neuron with a weight vector close to input vector **p** produces a value near 1 and then its output weight in the competitive layer will pass their values to the competitive function. It is also possible that several elements of **a1** are close to 1 since the input pattern is close to several training pattern.

4) Competitive Layer: There is no bias in Competitive Layer. In Competitive Layer, the vector **a1** is firstly multiplied with layer weight matrix **LW**, producing an output vector **n2**. The Competitive function, denoted as **C** in Figure 2.14, produces a 1 corresponding to the largest element of **n2**, and 0's elsewhere. The output vector of competitive function is denoted as **a2**. The index of K in **a2** is the number of output that can classify.

2.8 Support Vector Machines

SVMs are a kind of supervised learning based on statistical learning theory that are introduced by Vapnik and co-workers [16, 17]. The techniques have been developed by a number of other researchers [18, 19] and they are also showing high performances in biological, chemical and medical applications [20-22].

Support vector machines are a useful technique that can be applied to data classification or regression. SVM models were defined for finding a hyper-plane that linearly separable classes of objects. Figure 2.15 illustrates the objects of a linear separable SVM. Let vectors $x_i \in \mathbf{R}^n$, (i = 1, 2, …, m) and class labels $y_i \in \{-1, 1\}$. The term $f(x_i)$ can be represented by a linear function of the form by $y_i = f(x_i)$

$$f(x_i) = \langle w_i, x_i \rangle + b \qquad (2.15)$$

where $\langle w_i, x_i \rangle$ represents the inner product of **w** and **x**, **w** is weight vector and *b* is bias. The support vector machine approximates the set of data with linear function with the following function

$$y = \sum_{i=1}^{m} w_i \Phi(x_i) + b \qquad (2.16)$$

where $\{\Phi(x_i)\}_{i=1}^{m}$ represents the feature of input variables subjected to kernel transformation while $\{w_i\}_{i=1}^{m}$and *b* are coefficients.

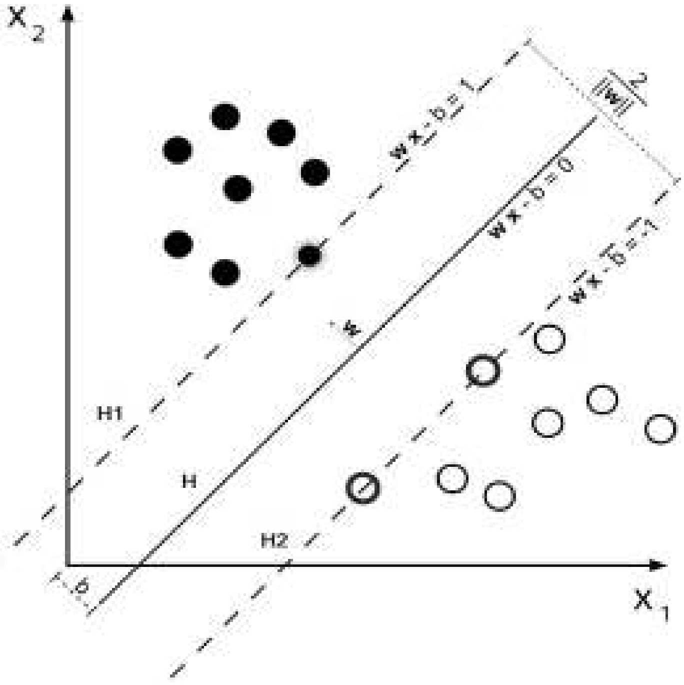

Figure 2.15 The objects of a linear separable support vector machine.

SVM can apply to non-linear classification by using the non-linear kernel functions to map the input data into a higher dimensional feature space in which the input data can be separated with a linear classifier (Figure 2.16). Kernel function *K*(**x**,**y**) can represents the inner product ⟨ϕ(**x**), ϕ(**y**)⟩ in feature space.

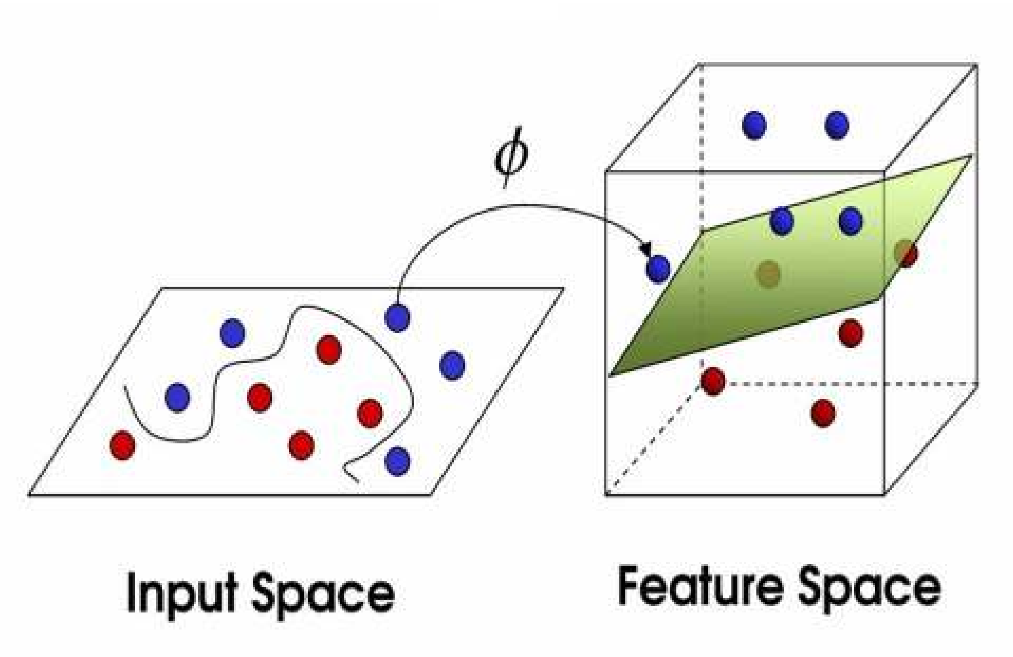

Figure 2.16 The mapping of non-linear classification into feature space.

The kernel functions are used in this study have three type of kernel functions

- Linear kernel

$$K(x, y) = x * y \qquad (2.17)$$

- Polynomial kernel

$$K(x, y) = (x * y)^d \qquad (2.18)$$

where *d* is the degree of the polynomial (*d* = 0 for linear kernel).

- Radial basis function kernel

$$K(x, y) = \exp\left(-\gamma \|x - y\|^2\right) \qquad (2.19)$$

where $\gamma > 0$

2.9 Ischemic Heart Disease

Ischemia heart disease is a condition in which the blood flow is blockage of the arteries that feed the heart muscle. Blocking causes cholesterol plaques deposits accumulate in the cells lining the wall of the coronary arteries [1]. These cholesterol plaques deposits build up gradually and irregularly in the large branches of the two main coronary arteries, show in Figure 2.17.

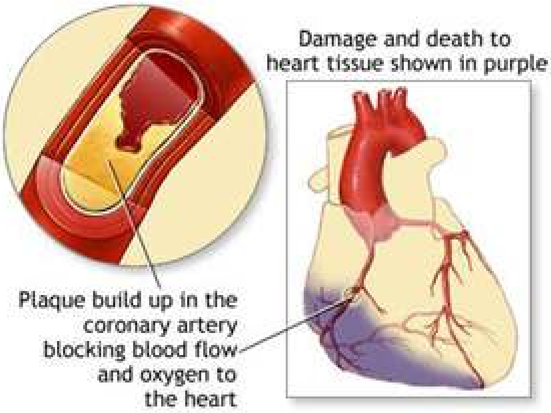
Damage and death to heart tissue shown in purple
Plaque build up in the coronary artery blocking blood flow and oxygen to the heart

Figure 2.17 The cause of ischemia heart disease [23].

This process leads to narrowing or hardening of blood vessels supplying blood to the heart muscle. Thus areas of the heart muscle may be damaged or die from lack of oxygen. Complete occlusion of the blood vessel can lead to a heart attack.

2.10 Data Mining Tools

2.10.1 MATLAB Software

Matlab is both a powerful computational environment and a programming language that easily handles matrix and complex arithmetic. It is a large software package that has many advanced features built-in, and it has become a standard tool for many working in science or engineering disciplines [10, 14]. Among other things, it allows easy plotting in both two and three dimensions. Typical uses include:

- Developing Algorithms and Applications
- Analyzing and Accessing Data
- Visualizing Data
- Performing Numeric Computation
- Publishing Results and Deploying Application

MATLAB is a high-level technical computing language and interactive environment. It is used in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

The Neural Network Toolbox extends MATLAB with tools for designing, implementing, visualizing, and simulating many kind of neural networks. Neural networks are uniquely powerful tools in applications where formal analysis would be difficult or impossible, such as pattern recognition and nonlinear system identification and control. The Neural Network Toolbox provides comprehensive support for many proven network paradigms, as well as graphical user interface that allows designing and managing networks.

2.10.2 WEKA Software

Weka is a collection of machine learning algorithms for data mining tasks. Weka software was developed by the University of Waikato in New Zealand in 1993. The original of Weka was non-Java version modeling algorithms implemented in other programming languages [24], but the more recent fully Java-based version (Weka3), was developed in 1997. Weka contains tools for data preprocessing, classification, regression, clustering, association rules, visualization, and feature selection.

Since the Weka system is open source software issued under the GNU General Public License (GNU GPL), people can modify the Weka system for using in many researches [25-27].

Weka provides implementations of learning algorithms that can easily apply to data mining tasks. The package of Weka has three different interfaces consist of a command line interface, an Explorer GUI interface, and an Experimenter GUI interface. The input file of Weka is formed relational tables or ARFF format, which can be access from text file. The complete functionalities of Weka is given in the on-line documentation. A short overview of the Weka functionalities [28]:

- **SVM:** only polynomial kernels are supported. Also, support vector regression is not supported.

- Decision trees: ID3 and C4.5 are implemented, and M5, a model tree induction algorithm for predicting numeric values (each leaf node has a regression model). PART is a rule-learner that makes rules by building different decision trees and each time keeping the leaf with the largest coverage.

- Memory-based methods: kNN and locally weighted regression.

- Neural Networks: only back-propagation with momentum is supported.

- Simpler methods: naive Bayes (for numeric values, a normal distribution is used, but also 'kernel density estimation' can be used to avoid assuming a normal distribution) and linear regression are useful simple methods. Two-class logistic regression is also supported. The algorithm uses a 'ridge estimator'.

- Other simple methods: decision tables, 1R (make a rule based only on one attribute) and decision stump (one-level decision trees). Although methods this simple might seem useless, they can be combined via boosting or bagging, and form a strong classifier through combining several weak ones.

Also meta-learning schemes are supported:

- Bagging

- Stacking: using a range of base classifiers and a meta classifier which classifies their output.

- Adaboost: a boosting method based on Freund & Schapire's Adaboost M1 method

- MultiClassClassifier: to solve multiclass problems using two-class classifiers: one classifier is built per class (it is also possible to ask for one classifier per pair of classes).

- CVParameterSelection: will try out a defined range over a set of parameters for a certain classifier using cross-validation, and use the best combination to build the final model.

Weka also includes a package that contains clustering algorithms. It supports:

- The EM algorithm: working with numeric as well as nominal values, but assuming that all attributes are independent.

- Incremental clustering: a clustering technique that builds a tree using the category utility measure.

kMeans is also provided.

- Association rules: the APriori algorithm is supported.

- Data preprocessing support: user can add new attributes (based on calculations of existing ones), transform attribute values, manually select certain attributes, discrete numeric values, remove attributes with only one distinct value, select records on the basis of attribute values, transform nominal values into binary ones, merge two nominal values into one, normalize numeric values, randomize the order of the dataset, replace missing values by the mean or the mode, create random subsamples, etc.

- Attribute selection: Both filter methods and wrapper methods are supported. Among the provided filter methods are the chi-squared method, the information gain and gain ratio measures. In wrapper methods the feature subset is evaluated using the actual classifier that is going to be used for classification. When searching for an optimal feature subset, a search strategy has to be selected, best-first, exhaustive search, forward selection, ranking, genetic search, random search will be applied.

- Visualization: Weka provides limited visualization possibilities. There are maximum three dimensions: 2 axis and one overlay color.

2.11 Related Researches

Embrechts et al. [29] using machine learning method for pattern recognition in magnetocardiography (MCG). This research has purpose for automate interpretation of MCG measurements to minimize human input for analysis, result on detecting ischemia, a condition arising in many common heart diseases that may result in heart attack, the leading cause of death. For unsupervised learning used Direct Kernel based Self-Organizing Maps. Direct Kernel Partial Least Squares and (Direct) Kernel Ridge Regression for supervised learning. Input data are acquired at 36 locations by making four sequential in mutually adjacent positions. For diagnosis of ischemia, a bandwidth of 0.5 Hz to 20 Hz is needed, the data from a time window between the J point and T peak of the cardiac cycle in which value for 32 evenly space point were used for automatic classification.

The analysis used the Analyze/StripMiner software package, but made use of SVMLib for SVM model. Using training set for optimized the parameters in DK-SOM, SVM, DK-PLS and LS-SVM before testing. The results show that kernel PLS (K-PLS), direct kernel PLS (DK-PLS), support vector machine (SVMLib), and least square SVM (LS_SVM) is generally excellent. In this case, DK-PLS gave a superior performance, but the differences between kernel-based methods are not significant.

Another paper that involves magnetocardiography classification is using Self Organizing Maps (SOMs) for classify abnormal and normal magnetocardiography (MCG). This research was presented by Naenna and co-worker [30], MCG data are acquired at 36 locations above the torso by making sequential time series. The complete time series is averaged using the maximum of the R peak of the cardiac cycle as trigger point. For automatic classification used data from a time window between the J point and T peak of the cardiac cycle in which values for 32 evenly spaced points. The training data consist of 74 cases. The testing was done on a set of 51 cases.

A typical 9x18 self-organizing map on hexagonal grid based, the dark hexagons indicate diseased cases and the light hexagons indicate healthy cases. Fully colored hexagons indicate the positions for the training data, while the white and dark-shaded numbers are the pattern identifiers for healthy and diseased test cases. Prediction results for the magnetocardiogram data that the predictions miss 15/51 test cases (5 healthy or negative case and 10 ischemia cases).

Comes and Kelemen have explored classifying microarray data patterns in gene expression [31]. They were used Probabilistic Neural Networks (PNN) compared with Bayesian Neural Networks to classify microarray data patterns in gene expressions. In this experiment that use gene expression data that identifies seven expression patterns (1= Metabolic, 2 = Early I, 3 = Early II, 4 = Early-Mid, 5 = Middle, 6 = Mid-Late and 7 = Late) during yeast sporulation. The acquired dataset contained 6118 rows of which only 477 rows possessed temporal pattern profiles. The 477 samples were ordered by temporal class and the temporal text class value was modified to a numerical value 1 through 7. The Training data consisted of input/target pairs. Each input row consisted of the seven time point measurements normalized. A

PNN offered by Mathworks Matlab and the Neural Network Toolbox is used to conduct the experiment.

The PNN showed that fairy good results with the testing conducted. A correct classification rate as high as 80.86%, with an average over 80%. The PNN model achieved only 10% lower classification rate than BNN, but PNN used lower time and model complexity than BNN.

Another Probabilistic Neural Network for classification was presented in A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network by Wu SG et al. [15] They employed Probabilistic Neural Network (PNN) to automate leaf recognition for plant classification. They had used 5 basic geometric features define 12 digital morphological features. To reduce the dimension of input vector of neural network, PCA was used to orthogonalize 12 features. They found that the contribution of first 5 principal components is 93.6%. To balance the computational complexity and accuracy. The PNN was trained by 1800 leaves to classify 32 kinds of plants. The accuracy of PNN was evaluated 10 pieces of leaves from each kind of plant from testing sets, the average accuracy is 90.312%.

Gao et al. [13] had developed a system for detection of cardiac arrhythmias within ECG signals. ANN Classifier was built by the use of a logistic regression model and back propagation algorithm based on a Bayesian framework. Performance of this model was evaluated by comparison with other classifiers including Naive Bayes, Decision Trees, Logistic Regression, and RBF Networks. The results Naive Bayes, Logistic Regression, Decision Trees, RBF Networks and Bayesian ANN show that the scores of the Bayesian ANN Classifier for sensitivity, Specificity and False Discovery Rate are higher or equivalent to those of the other methods.

CHAPTER III
MATERIALS AND METHODS

3.1 Research Methodology

The process model for data mining will be applied to this research for developing methodology. The Cross Industry Standard Process for Data Mining (CRISP-DM) was established by three veterans consist of Daimler Chrysler (then Dainler-Benz), SPSS (then ISL), and NCR in 1996 [32]. The life cycle of a CRISP-DM consists of six phases. Figure 3.1 shows the phases of a data mining process.

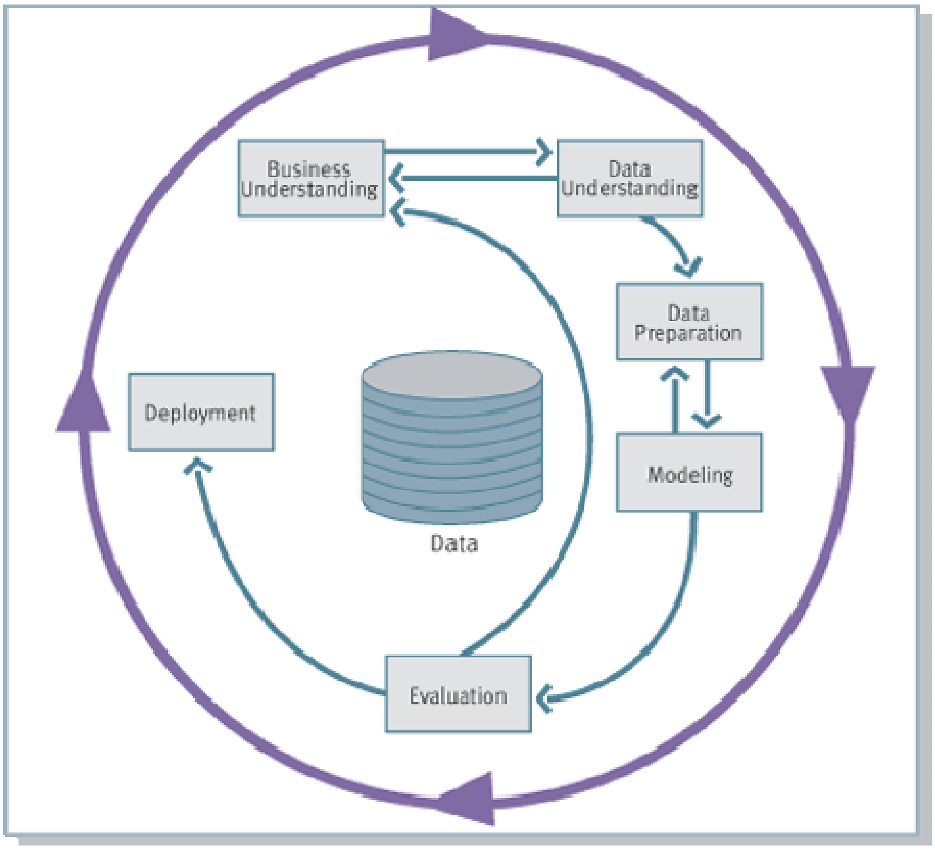

Figure 3.1 Phases of the CRISP-DM Process Model [32].

3.1.1 Business/ Project Understanding

This research proposes the use of machine learning for identification of IHD patients by using MCG data that high sensitivity and avoid skin-electrode contact in traditional ECG method. The data mining techniques apply to MCG data will reduce spending time for interpretation of MCG.

3.1.2 Data Understanding

MCG data were acquired at 36 locations in 6x6 matrixes above the torso by making four sequential measurements in mutually adjacent positions. In each position the nine sensors measure the cardiac magnetic field for 90 seconds using a sampling rate of 1000 Hz leading to 36 individual time series.

3.1.3 Data Preparation

For diagnosis of ischemia, a bandwidth of 0.5 Hz to 20 Hz is needed, so hardware low pass filter at 100 Hz using 6th order Bessel filter characteristics is applied, followed by an additional digital low pass filter at 20 Hz using the same characteristics, but a high order. To eliminate remaining stochastic noise components, the complete time series is averaged using the maximum of the R peak of the cardiac cycle as a trigger point. Data from a time window between the J point and T peak of the cardiac cycle were used for automatic classification [33], in which value for 32 evenly spaced point were interpolated from the measured data.

3.1.4 Modeling

The training data consist of 74 cases were randomly selected from 125 cases. In this study, to optimize parameters of neural network 10-fold cross-validation approach is used. The previous study illustrated that 10 folds seem to be an optimal number of folds [34]. In this approach, the dataset is divided into 10 folds with equal-sized. During each run, one of folds is chosen to test the optimal parameter of the neural network, while the remaining nine folds are used for training. For each parameter of neural network is selected using the average of root mean square error.

1) Neural network architecture design and development

After the data representation, the next step is to design and develop the neural networks. The flow diagram of the design and development process of neural network is shown in figure 3.2.

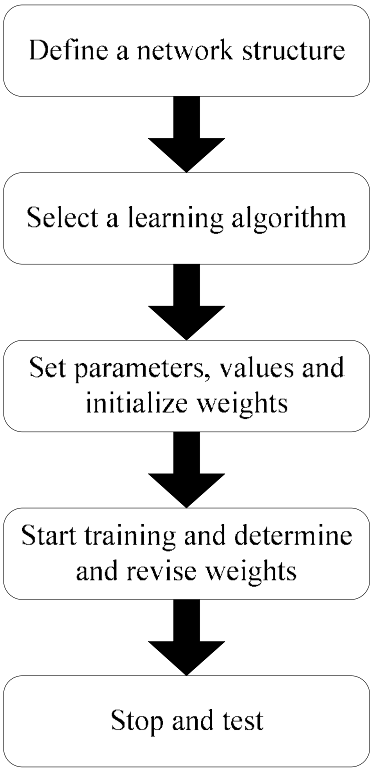
Define a network structure
Select a learning algorithm
Set parameters, values and initialize weights
Start training and determine and revise weights
Stop and test

Figure 3.2 Flow diagram of the design and development process of neural network

In this step, neural network architecture and a learning method are selected to create neural network system that suitable for input and output data characteristics.

And learning step, there are parameter for tuning the network by adjust the weights. This step can be iterate by change amount of input nodes, hidden nodes, output nodes or network architecture or learning algorithm for the desire outcome.

2) Neural network architecture and parameter tuning

The Back-propagation and Bayesian neural network used in this study consists of a fully connected three-layer network (input layer, hidden layer, and output layer) with the following structural characteristics. The neural network has one input layer, one hidden layer and one output layer. The Probabilistic neural network consists of three-layer same Back-propagation and Bayesian neural network but Probabilistic neural network have input layer, radius basis layer, and competitive layer.

The most basic method of training a neural network is trial and error by adjusting various parameters.

Root mean square (RMS) error was used a measure of the prediction error by the trained model that is calculated with following equation.

$$RMS = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \tag{3.1}$$

Where p_i is the predicted output, a_i is the actual output, and n is the number of compounds in the dataset. The RMS of various parameters was calculated and the optimal value to use for each parameter was those processing low RMS value.

3) Support vector machine parameter optimization

The support vector machine was implemented with WEKA [24] using the sequential minimal optimization (SMO) algorithm. Linear kernel support vector machine was optimized C parameter in the range of 2^{-17} to 2^{17} by 2 step increasing.

Polynomial kernel support vector machine has two parameters (exponent parameter and C parameter). The exponent parameter was adjusted value from 2 until 10 and C parameter was adjusted value from 2^{-17} until 2^{17} by 2 step increasing.

Finally, the radial basis function kernel support vector machine was optimized by setting of C parameter and gamma (γ) parameter. The optimal value of parameter was determined by making a surface plot of RMSE as a function of C parameter and γ parameter.

3.1.5 Evaluation

The neural network performances and support vector machine performances were evaluated on the testing data that were done on a set of 51 cases that included patients whose magnetocardiograms.

1) Confusion matrix

A confusion matrix contains information about actual and predicted classifications done by neural network. Performance of neural network is commonly evaluated using the data in the matrix. Table 3.1 shows the confusion matrix for a two class classifier.

Table 3.1 Representation of confusion matrix

Predicted / Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

- TP is the number of correct predictions that an instance is positive.
- FN is the number of incorrect predictions that an instance is negative.
- FP is the number of incorrect predictions that an instance is positive.
- TN is the number of correct predictions that an instance is negative.

2) Classification Performances

Clinical research often investigates the statistical relationship between symptoms (or test results) and the presence of disease. When significant associations are found, it is useful to express the data in ways which are clinically relevant. Therefore, sensitivity and specificity are used to measure the accuracy of the model and verify that classifier acquires the underlying dynamics of the system from the data.

Let: FP = false positive.
FN = false negative.
TP = true positive.
TN = true negative.

Then, sensitivity is defined as the probability that a symptom is present (or a test predicts that the person has the disease) given that the person has the disease.

$$Sensitivity = \frac{TP}{FN+TP} \qquad (3.2)$$

Specificity is defined as the probability that a condition is not present (or a test predict that the person does not have the disease) given that the person does not have the disease.

$$Specificity = \frac{TN}{FP+TN} \tag{3.3}$$

As well as sensitivity and specificity, another widely used measurement, the accuracy is used to examine how well the neural network classifier performs in recognizing ischemia heart disease and to compare each neural network model. Accuracy is defined as the expected proportion of true predictions in the set of prediction, rather than quantifying the chance of any true positives.

$$Accuracy = \frac{TP+TN}{FP+TP+FN+TN} \tag{3.4}$$

3.1.6 Deployment

In this process, the extracted knowledge is fed back to the business environment. In order words, the knowledge obtained will need to be organized and presented that often involves applying models within an organization's decision making processes.

3.2 Research Tools

3.2.1 Hardware

Personal Computer

- CPU	:	Intel Pentium4 2.4GHz or higher
- RAM	:	DDR 2048 MB
- Hard Disk	:	250 GB
- Monitor	:	VGA Monitor
- Peripherals	:	Keyboard, Mouse, Printer, Diskette and CD-ROM Drive

3.2.2 Software

- Operating System	:	Microsoft Windows XP
- Programming Language	:	Matlab 7.0.4 R14 SP2, WEKA

3.3 Research Schedule

Activities	Time (Months)											
	1	2	3	4	5	6	7	8	9	10	11	12
1. Literature Review	■											
2. Project Understanding		■										
3. Data Understanding and Data Preparation		■	■									
4. Modeling				■	■	■	■	■				
5. Evaluation								■	■			
6. Conclude the results									■	■	■	
7. Final thesis document								■	■	■	■	■

CHAPTER IV
RESULTS

This chapter was about result of neural network optimization and the performance of classification from each algorithm of neural network.

4.1 Optimization of Neural Networks

The neural network architectures of each algorithm were selected by tuning various parameters by trial-and-error. The results of tuning parameters were illustrated in topic 4.1.1 to 4.1.3.

4.1.1 Back-propagation neural network

- Hidden node

The optimal number of nodes in the hidden layer was determined by varying the number of hidden nodes from 1 to 50. Others parameter was set as default. The plotting of average RMSE in figure 4.1, the minimum of average RMSE appeared at 24 hidden nodes therefore the optimal number of nodes was 24 hidden nodes.

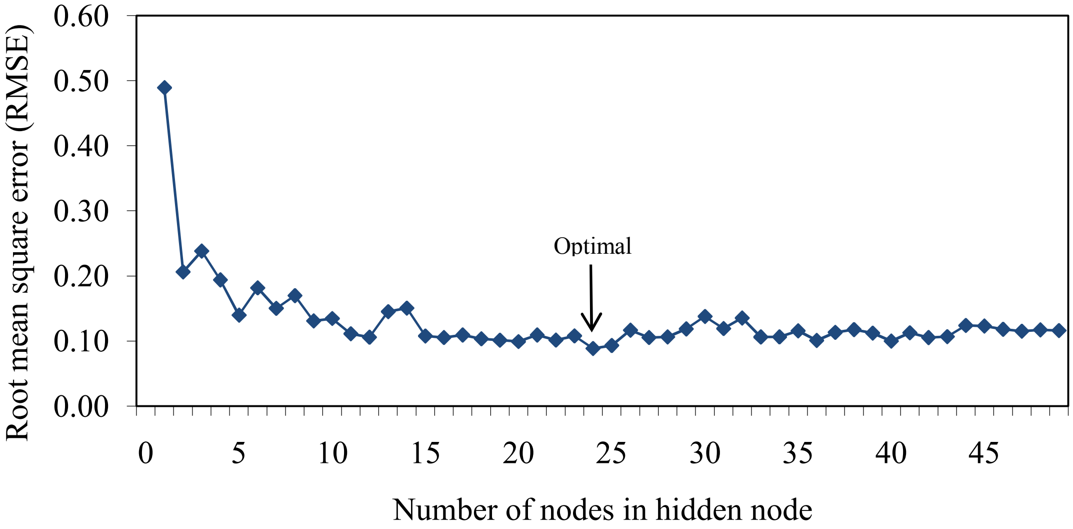

Figure 4.1 The plotting of average RMSE versus hidden nodes of BPNN

- Learning epoch

After hidden node parameter tuning, next step is learning epoch parameter tuning. The hidden node parameter was selected as 24 and both learning rate and momentum were set as 0.1. The number of learning epoch was started from 50 to 1000 and RMSE were measured at every 50 learning epoch.

The graph compares the rate of average RMSE between validation set and training set. It can be seen that trend of average RMSE of training set decreased from 1 epoch to 1000 epochs. The optimal learning epoch to stop training is 100 epochs for the BPNN.

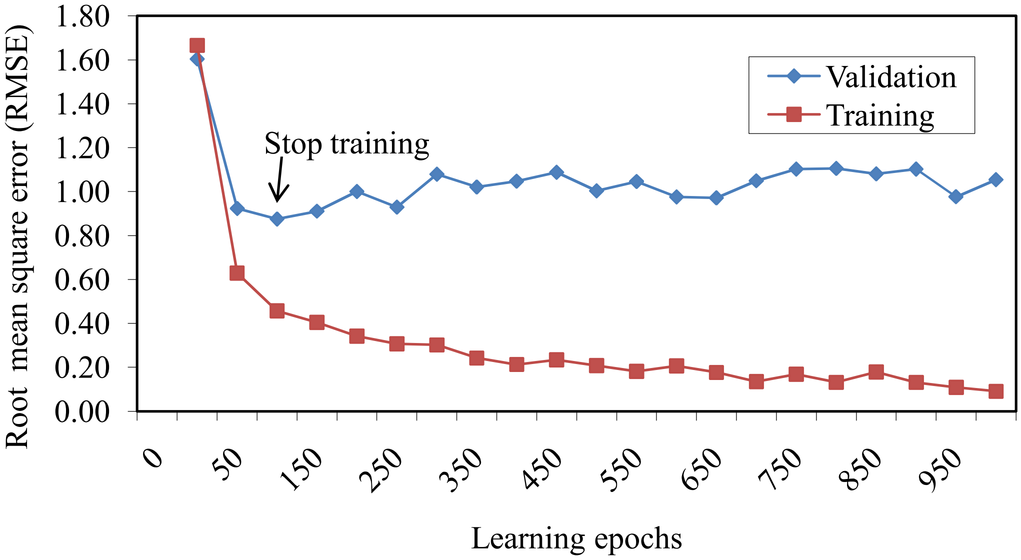

Figure 4.2 The plotting a graph RMSE versus the number of learning epoch of BPNN

- Momentum and Learning rate

The optimal learning rate and momentum were determined by varying the number of learning rate and momentum in the range of 0 to 1 then making a surface plot of RMSE as a function of learning rate and momentum. The global minimum of error surface is the optimal learning rate and momentum. We found that the BPNN has the optimal learning rate and momentum is 0.1 and 0.7 (Figure 4.3)

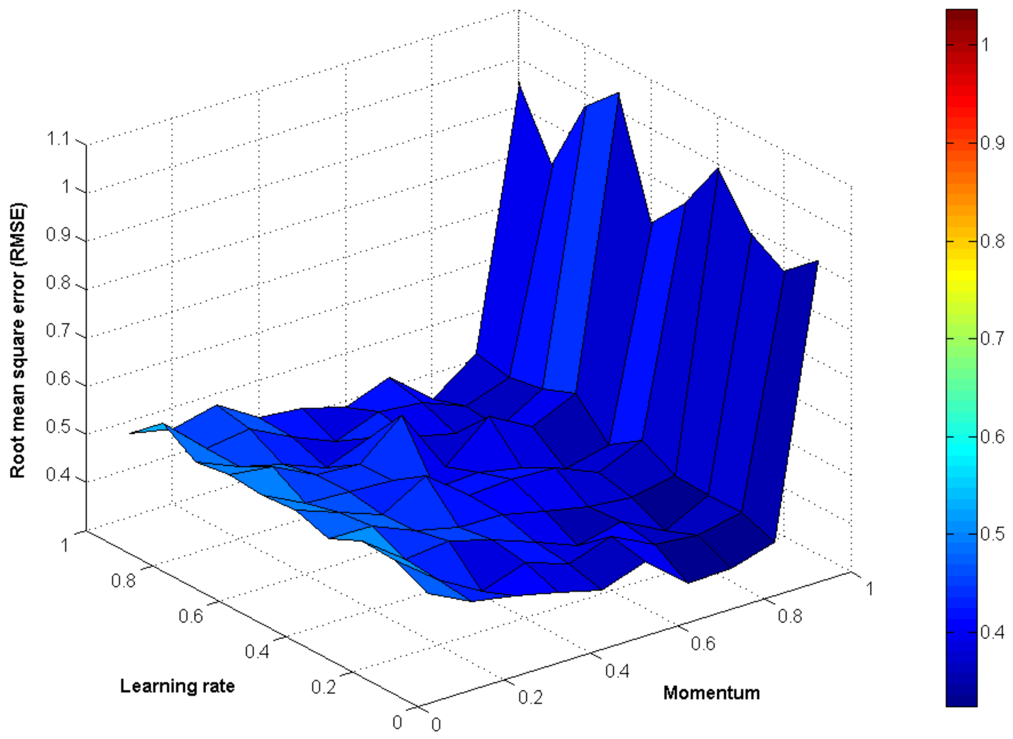

Figure 4.3 The error surface is the optimal learning rate and momentum of BPNN

Finally, the back propagation neural network architecture consists of the number of node in hidden layer is 24 nodes, learning epoch is 100 epochs, learning rate and momentum is 0.1 and 0.7, respectively.

4.1.2 Bayesian neural network

In the same way, bayesian neural network was built by use of a logistic regression model and the back propagation algorithm based on a bayesian frameworks.

- Hidden node

Optimization of node in hidden layer of BNN similar the optimization of BPNN, making a plot of average RMSE versus number of hidden node by varying the number of hidden nodes in the range from 1 to 50. Figure 4.4 shows that the optimal number of nodes in hidden layer was 11 nodes.

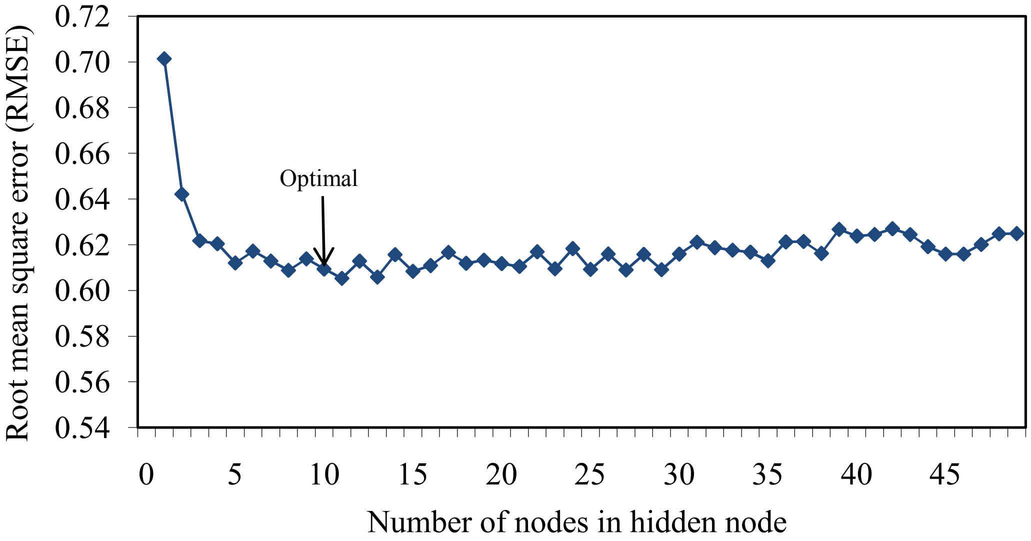

Figure 4.4 The plotting of average RMSE versus hidden nodes of BNN

- Learning epoch

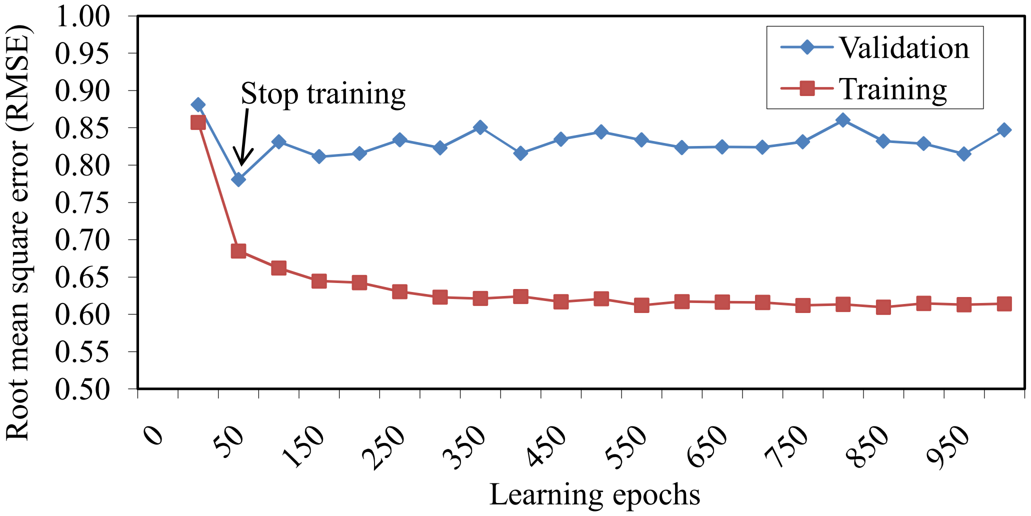

Figure 4.5 The plotting a graph RMSE versus the number of learning epoch of BNN

In the same method, the size of learning epoch was defined by varying the learning epoch between 50 and 1000 at interval of 50. The graph reveals that the minimum average RMSE of validation set at 50 epochs, therefore 50 epochs was chosen for the optimum of learning epoch.

- Momentum and Learning rate

The momentum and learning rate were chosen by concurrently varying the momentum and learning rate between 0 and 1 at interval of 0.1 then making surface plot of RMSE as a function of momentum and learning rate. From the figure 4.6, it is observable that the optimal value exist in global minimum of error surface that momentum is 0.1 and learning rate is 0.6.

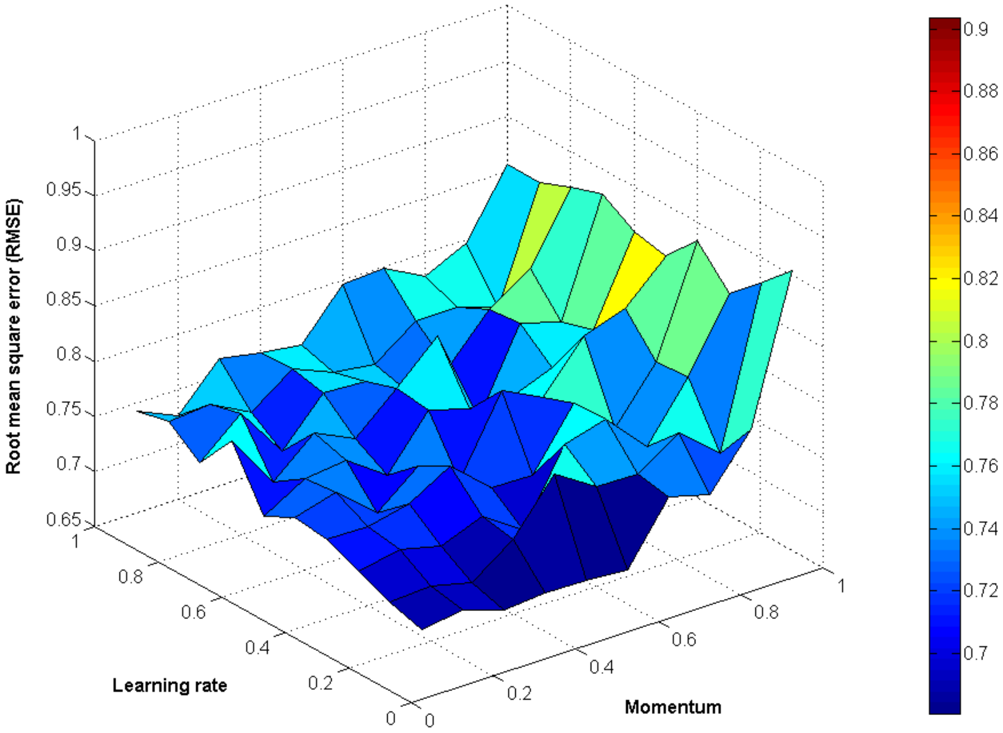

Figure 4.6 The error surface is the optimal learning rate and momentum of BNN

BNN are number of node in hidden layer is 11 nodes, learning epoch is 50 epochs, learning rate and momentum is 0.1 and 0.6, respectively.

4.1.3 Probabilistic neural network

- Spread value

The PNN trained and validated 10 times which the PNN models were trained each with varied spread parameter from 0.1 to 5 at interval of 0.1. In each case, one of the folds is taken as validation data and remaining folds are added to training data. Thus, 10 different validation results exist for each training data.

The spread parameter value for PNN model is chosen at the maximum average of accuracy. The highest accuracy was observed at spread value is 0.9 (Figure 4.7). Thus, it was chosen as the optimal of spread value.

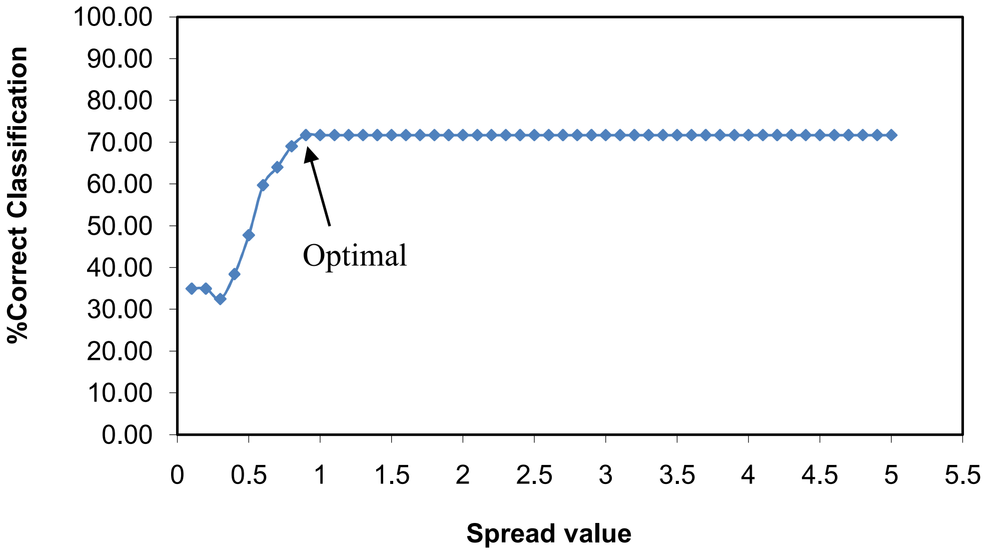

Figure 4.7 The Plot of correct classification with varying spread values

4.2 Optimization of Support Vector Machines

The support vector machine was optimized C parameter in linear kernel SVM in the range of 2^{-17} to 2^{17} by 2 step increasing. The figure 4.8 shows the plotting of RMSE versus C parameter. The minimum value of RMSE is optimal point, therefore the optimal of C parameter is 2^{-3} for linear kernel SVM.

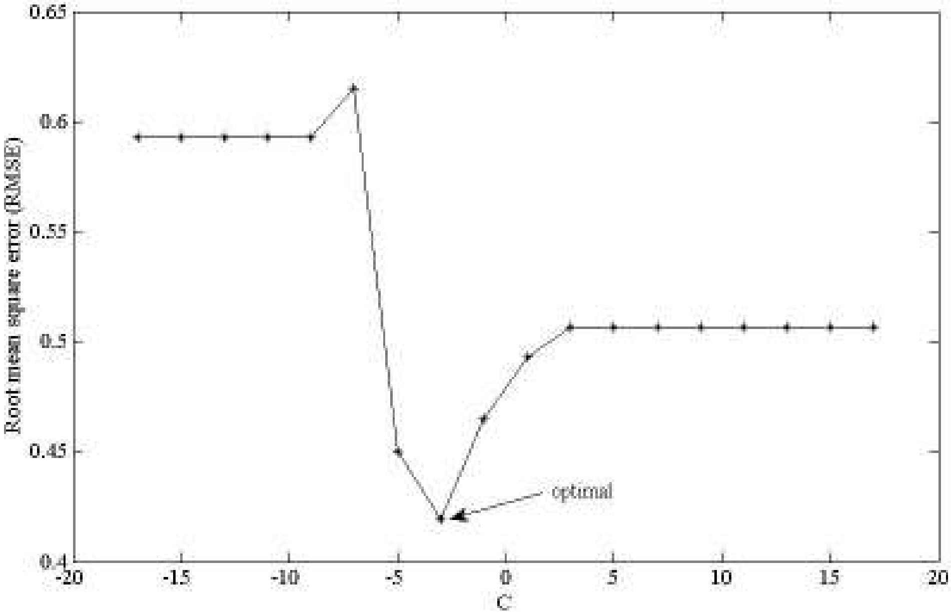

Figure 4.8 The plotting of average RMSE versus C of linear kernel SVM

The optimal parameters of polynomial kernel SVM has two parameters (exponent parameter and C parameter). The exponent parameter was adjusted value from 2 until 10 and C parameter was adjusted value from 2^{-17} until 2^{17} by 2 step increasing. We found that optimal exponent parameter is 2^2 and C parameter is 2^{-13} as shown in figure 4.9.

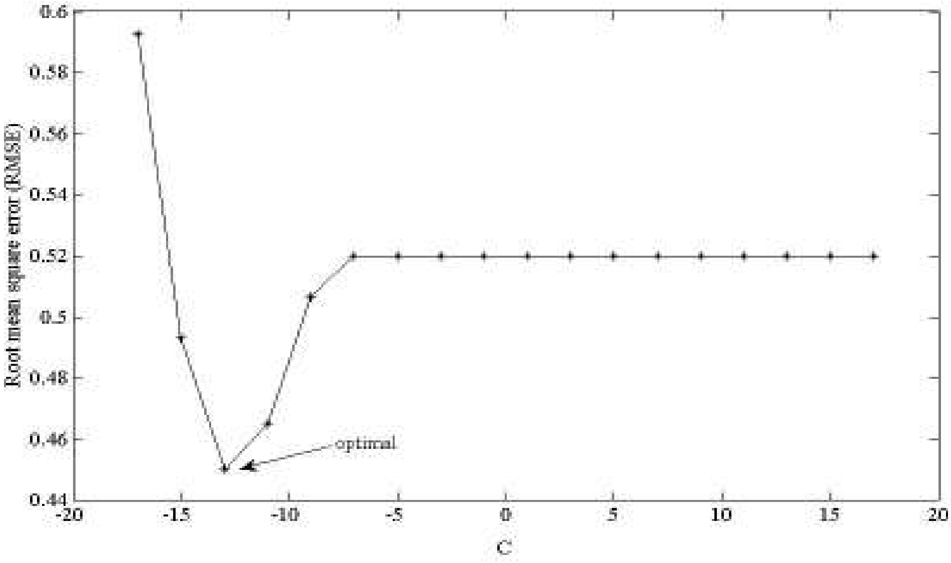

Figure 4.9 The plotting of average RMSE versus C of polynomial kernel SVM

Finally, the RBF kernel SVM was optimized by setting of C parameter and gamma (γ) parameter. The optimal value of parameter was determined by making a surface plot of RMSE as a function of C parameter and G parameter. The optimal C parameter and γ parameter are 2^{13} and 2^{-17} (Figure 4.10) respectively.

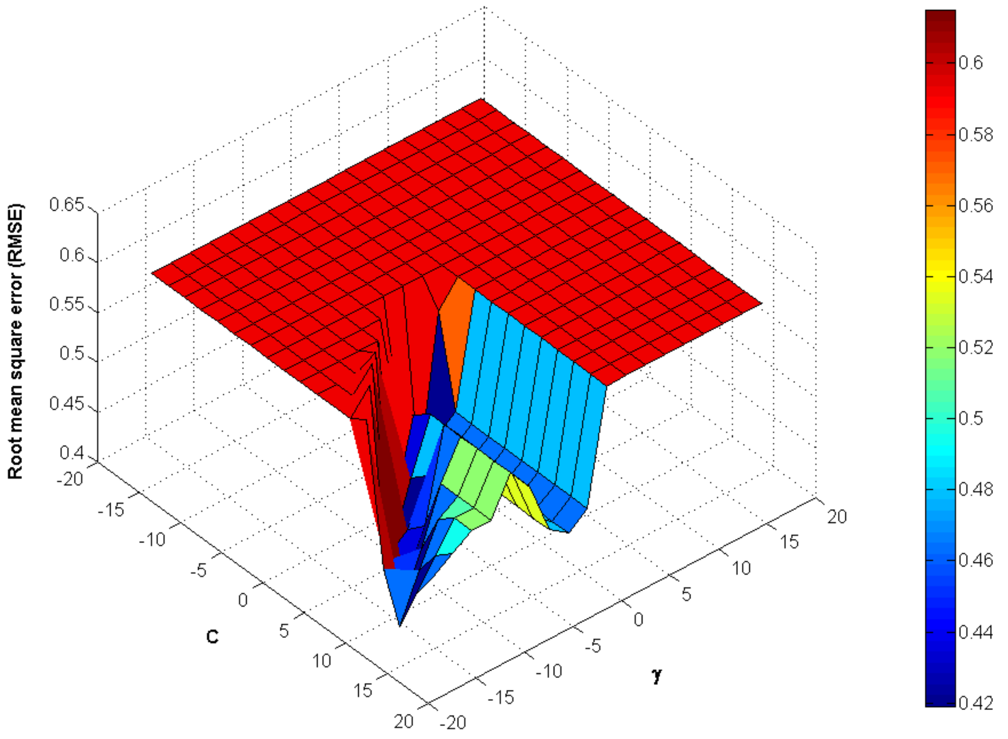

Figure 4.10 The error surface of the optimal C and gamma of RBF kernel SVM

4.3 Prediction of Ischemic Heart Disease

The models were evaluated based on MCG testing set 51 cases which consists of 29 IHD patients and 22 healthy. The prediction results are represented in the confusion matrix and three parameters (accuracy, sensitivity and specificity).

4.3.1The receiver operation characteristic (ROC)

In order to design the prediction performance of output classification, ROC curve is calculated by analyzing the output data obtained from the testing set by changing the decision threshold [36, 37]. The ROC curve is a plot of the true positive rate (sensitivity) against the false positive rate (1-specificity) for each decision threshold.

A good decision threshold should be located as close as possible to the upper left corner of ROC curve. Furthermore, the performance of the model can be measured by calculating the area under the curve [2, 35].

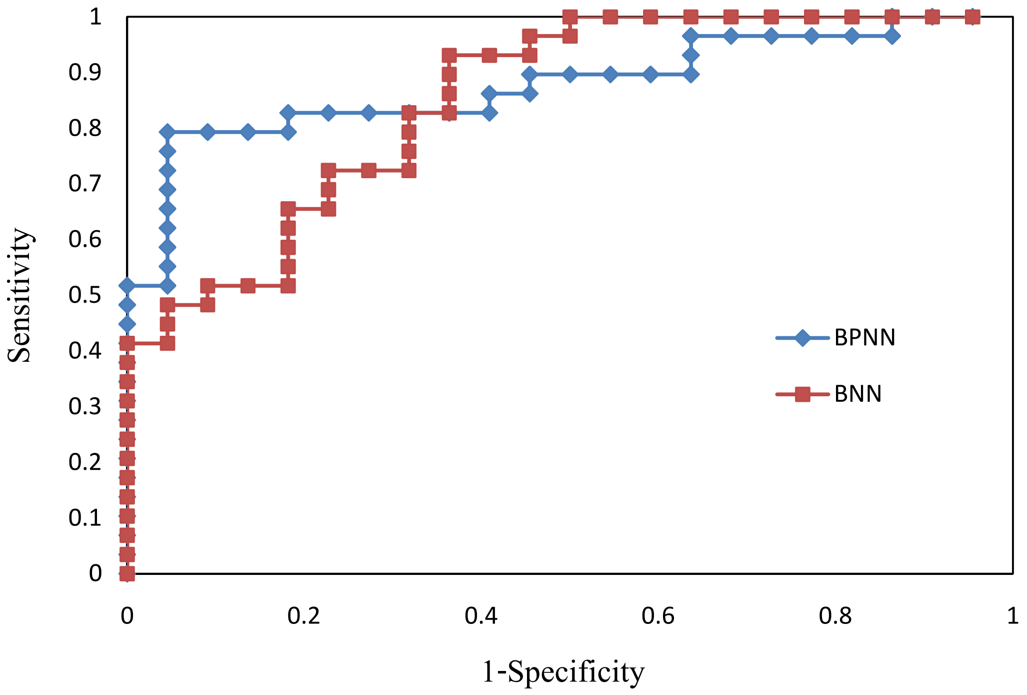

Figure 4.11 ROC curve showing possible trade-offs between true positives (sensitivity) and false positives (1-specificity)

From the figure 4.11, the decision threshold of BPNN is -0.2774 and the area of ROC curve is 0.90596. While the decision threshold and the area of ROC curve of BNN are 0.0470 and 0.84953 respectively. The area of ROC curve of BPNN more than the area of ROC curve of BNN. Therefore, the BPNN model has performance more than BNN model.

4.3.2 Classification results

Table 4.1 The confusion matrix of classification models.

Predicted / Actual	BPNN		BNN		PNN	
	IHD	**Normal**	**IHD**	**Normal**	**IHD**	**Normal**
IHD	25	4	28	1	25	4
Normal	7	15	10	12	11	11

Table 4.1 The confusion matrix of classification models. (cont.)

Predicted / Actual	Linear kernel SVM		Polynomial kernel SVM		RBF kernel SVM	
	IHD	**Normal**	**IHD**	**Normal**	**IHD**	**Normal**
IHD	20	9	26	3	12	17
Normal	4	18	12	10	3	19

Herein, machine learning models for identification of ischemic heart disease were based on three learning algorithms that include back propagation neural network (BPNN), Bayesian neural network (BNN) and probabilistic neural network (PNN). Each of predictive models was constructed by training with 74 MCG patterns (26 IHD and 48 normal cases) under optimal parameter settings. The performance of IHD classification was then tested on 51 MCG patterns consisting of 29 IHD cases and 22 normal controls. To investigate the prediction performance and to find the optimum decision threshold, ROC curve of each model was generated and shown in figure 4.11. The decision threshold of BPNN and BNN was varied in the range of -1 to 1 and 0 to 1, respectively.

The resulting prediction sensitivities were plotted as a function of one minus relevant specificities. The model giving greater area under ROC curve provides better prediction performance than the model offering the lesser area. It was observed that area under ROC curve given by BPNN model 0.90596 exhibited slightly higher than that given by BNN model 0.84953. The most advantageous decision thresholds were 0.1 and 0.5 for BPNN and BNN models, respectively. As represented in Table 4.1, 25 of 29 IHD cases could correctly be identified by BPNN and PNN, while only one IHD was failed to classify by BNN (see Table 4.1). In the classification of normal individuals, BPNN outperformed both BNN and PNN. Fifteen normal MCGs were correctly classified by BPNN, while 10 and 11 normal patterns were made by BNN and PNN.

Table 4.2 The prediction performance of classifier

Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)
BPNN	78.43	86.21	68.18
BNN	78.43	96.65	54.55
PNN	70.59	86.21	50.00
Linear kernel SVM	74.51	68.97	81.82
Polynomial kernel SVM	70.59	89.66	45.45
RBF kernel SVM	60.78	41.38	86.36

For the SVM models, the linear kernel SVM could be the highest correctly classified by 18 of 22 normal cases and 20 of 29 IHD cases (see Table 4.1) but the highest sensitivity is polynomial kernel SVM and the highest specificity is RBF kernel SVM (see Table 4.2).

In addition, prediction performance of those six machine learning models could be statistically demonstrated in the function of prediction accuracy, sensitivity and specificity as shown in Table 4.2.

The gold standard for clinical diagnosis need to obtain the maximum sensitivity, specificity, and accuracy. The maximum accuracy was used to present the prediction performance for both healthy people and unhealthy people. The BPNN and BNN have the same highest accuracy of 78.43%. Comparable accuracy of prediction (78.43%) was observed for BPNN and BNN learning models. However, BNN exhibited relatively higher in sensitivity but lesser specificity of predictions than those made by BPNN.

The high sensitivity model will present the good performance for prediction of unhealthy people when used on unhealthy people. The BNN has the highest sensitivity of 96.65%. While the high specificity model will present the good performance for prediction of healthy people when used on healthy people. The RBF kernel SVM came out to be the best with 86.36% specificity.

CHAPTER V
DISCUSSION

The purpose of this research is to mining data of magnetocardiograms and to compare performance for prediction of ischemic heart disease by applying neural network algorithm. In performance comparison, we evaluate the performance of each algorithm in the 51 testing data.

5.1 Data Collection

The raw medical data may consist noise and missing value, which can affect the predictive accuracy. As a result, raw medical data must be require data cleaning and data preprocessing that make the data more suitable for data mining [38, 39].

By improving the collection of the data can reducing the problem of noise, missing data, redundant or inconsistent values data. As mentioned above, the data cleaning and data preprocessing are directly affected by the quantity and quality of the data [40]. For noise reduction in this study, a low pass filter is applied and the complete time series is averaged by using the maximum of the R peak of the cardiac cycle as a trigger point to eliminate stochastic noise.

5.2 Data Mining

Machine learning plays an important role in knowledge discovery of medical data. It is widely applied for identification or classification of many diseases. For example, predicting of cancer diseases and heart diseases [41-45]. Including our previous works, we have employed various machine algorithms in attempt to automatically distinguish between normal MCG and ischemic patterns. Under utilization of direct kernel-self organizing map (DK-SOM), direct kernel-partial least square (DK-PLS) and back propagation neural network (BPNN) on wavelet and time domains of MCG,

Herein, we employ six kinds of machine learning model to identification of ischemic heart disease using J-T interval of abnormally MCG pattern can be useful tool for high sensitivity screening for IHD [46, 47]. On the other hand MCG had higher sensitivity for detecting myocardial ischemia then conventional 12-lead ECG [48].

As seen in the previous diagnosis heart patterns for IHD with direct kernel methods. Embrechts and Szymanski reported classification accuracy between 71% and 83% using direct kernel based self-organizing maps (DK-SOM), direct kernel partial least square (DK-PLS) and least-squares support vector machines (LS-SVM) also known as kernel ridge regression [29]. Moreover, the back-propagation neural network (BPNN) and direct kernel self-organizing map (DK-SOM) were applied to identification of IHD. The BPNN obtained 89.7% sensitivity, 54.5% specificity and 74.5% accuracy, while the DK-SOM obtained higher sensitivity, specificity and accuracy than BPNN. The DK-SOM obtained 86.2% sensitivity, 72.7% specificity and 80.4% accuracy [36].

In this study, we used neural network for classification task but using difference algorithm for neural network. Result of BPNN and BNN show that both classifier models had the equal 78.43% classification accuracy which higher than PNN classifier, the PNN classifier had 70.59% classification accuracy. Sensitivity of BPNN classifier (86.21%) lower than BNN classifier (96.55%), while specificity of BPNN classifier (68.18%) higher than BNN classifier (54.55%). Although in this study, the PNN classifier had the lowest accuracy but it had the highest specificity (73.33%). In addition for SVM models, linear kernel SVM came out to be the best classification accuracy of 74.51% for SVM models but less then BPNN and BNN. The polynomial kernel SVM had the best sensitivity (89.66%) but less then BNN and The RBF kernel had the highest specificity (86.36%) that the best specificity of six models.

5.3 Classification models

Sensitivity and specificity are measures of a test's ability to correct classify a person as having a disease or not having disease. A high sensitivity means that fewer cases of disease are missed. A high specificity means that a fewer cases of not having disease are missed. The dataset may be sending to the highest sensitivity model (BNN) for screening on high risk people for reducing the amount of data before sending to the highest specificity model (RBF kernel SVM) or expert to interpret.

5.4 Tools in the research

This research used neural network toolbox of Matlab and support vector machine module of WEKA that are widely used in data mining task. The support documents of software can be found on many books and internet. Moreover, they have GUI (Graphic user interface) mode that can help to easy for development of neural network and support vector machine.

5.5 Limitations

5.1. The MCG dataset is necessary to pre-process before data are used for input of neural network.

5.2 The MCG dataset only collect with one time period therefore we cannot monitor the prediction result of the patient in the future.

CHAPTER VI
CONCLUSION AND RECOMMENDATION

6.1 Conclusion

This research presents a comparative of prediction model for IHD identification by using three algorithms of neural network and three kernels of support vector machine. The 125 cases were randomly separated in to 74 cases for training set and 51 cases for testing set. In order to optimize the neural network structure, we used a 10-fold cross-validation on training set. In 10-fold cross-validation, the training set was randomly divided into 10 fold, each fold was used once as validation set and remaining nine-tenth was used for training set. We repeated this procedure for 10 times so that each fold would be used as the training set and validation set. The optimal parameters of neural network are determined by averaging the 10 parameter values.

In order to optimize the neural network parameters and support vector machines parameters, researchers use 10-fold cross-validation. In BPNN and BNN have four parameters are number of node in hidden layer, learning epoch, learning rate and momentum. The results show that optimal parameters of BPNN are number of node in hidden layer is 24 nodes, learning epoch is 100 epochs, learning rate and momentum is 0.1 and 0.7, respectively. BNN are number of node in hidden layer is 11 nodes, learning epoch is 50 epochs, learning rate and momentum is 0.1 and 0.6, respectively. The PNN has only one parameter to optimize that is spread parameter, the spread parameter value for PNN model is 0.9 because the maximum average of accuracy.

The minimum value of RMSE for linear kernel SVM can be found at C parameter is 2^{-3}. Polynomial kernel SVM has two parameters are exponent parameter and C parameter. We found that optimal exponent parameter is 2^{2} and C parameter is 2^{-13} and the optimal value of RBF kernel parameters, C parameter and gamma (γ) was obtained by making a surface plot of RMSE as a function of C parameter and γ

parameter. The value of C parameter and γ parameter that gave minimum RMSE is 2^{13} and 2^{-17}, respectively.

Finally, the prediction performance to compare MCG classification for IHD identification by using three algorithm of neural network and three kernels of support vector machine were applied on 51 cases of testing sets. The results shown that the BPNN and BNN performed the highest classification accuracy of 78.43%, the RBF kernel SVM performed the lowest classification accuracy of 60.78%. The BNN presented the best sensitivity of 96.55% and the RBF kernel SVM shown the lowest sensitivity of 41.38%. Both the polynomial kernel SVM and the RBF kernel SVM presented the minimum and maximum specificity of 45.45% and 86.36% (Table 4.2).

Although data mining methods can be applied to many task in medical researches for pattern recognition, classification, clustering, and predicting for knowledge discovery. The knowledge from data mining can reduce spending time in extracting knowledge from the large data and reduce error from human, but the knowledge that found from data mining methods should be evaluated by expert people.

6.2 Recommendation

This research purposes to compare of several prediction models for IHD classification. The classification results show that each classification models has different characteristics. Therefore, to improve the prediction accuracy may be use coupling system or voting system. In order words, the models that have high sensitivity may be use initially to detect IHD then the results send to the models that have high specificity.

In addition, to increase the reliability of the classification results, more dataset of MCG should be collected for training neural network. Additional factor that influence the IHD such as weight, age, cholesterol level, and etc might also be considered to be input of neural network.

Finally, in order to improve more efficiency or adaptive system, we need to collect MCG dataset in the same patient but different time period for monitoring result of prediction.

REFERENCES

1 Faculty of Science, Thaksin University [Online].[cited 2008 Dec 19]; Available from : URL : http://www.sci.tsu.ac.th/sciptl/radio/tape043/colesterol.doc.

2 Tan P, Steinbach M, Kumar V. Introduction to Data Mining. Pearson Education; 2006.

3 Weiss S. and Indurkhya N. Predictive Data Mining-a Practical Guide. Morgan Kaufmann Publishers. San Francisco; 1998.

4 Cryoton Co., Ltd. [Online]. [cited 2008 Dec 25]; Available from: URL : http://cryoton.webzone.ru/articles.htm.

5 Baule G, McFee R. Detection of the magnetic field of the heart. American Heart Journal. 1963; 66(1):95 - 96.

6 Baule G, McFee R. Theory of magnetic detection of the heart's electrical activity. Journal of Applied Physics. 1965;36(6):2066 - 2073.

7 Weak Superconductivity. Quantum Interferometers, Their Application /Edited by. B.B. Shwarts, S.Phoner. - M.:Mir, 1980. pp.256.

8 Han J. and Kamber M. Data Mining : Concepts and Techniques. Morgan Kaufmann. New York; 2001.

9 Dagli CH. Artificial neural networks for intelligent manufacturing. London: Chapman & Hill; 1994.

10 Chuchuen J. Soil Temperature Forecasting Using Neural Network. Technology of Information System Management, Mahidol University, Thailand.

11 Haykin S. Neural Network A Comprehensive Foundation 2nd ed. Prentice Hall; 1999.

12 Watcharabusaracum W. Developed System for Identification of Pathogenic Enterobacteriaceae and Related Organisms Using Artificial Neural Network. Technology of Information System Management, Mahidol University, Thailand.

13 Gao D, Madden M, Chambers D, Lyons G. Bayesian ANN Classifier for ECG Arrhythmia Diagnostic System: A Comparison Study. Proceedings of International Joint Conference on Neural Networks 2005.

14 Neural Network Toolbox 4 for designing and simulating neural networks. [cited 2008 Dec 19]; Available from: URL : http://www.mathworks.com

15 Wu SG, Bao FS, Xu EY, Wang YX, Chang Y, Xiang Q. A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network. IEEE,2007.

16 Vapnik VN. Statistical Learning Theory. New York. Wiley; 1998.

17 Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory AMC Press; 1992; Pittsburgh, USA. pp. 144-152.

18 Cristianini N, Shawe-Taylor J. An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press. Cambridge; 2000.

19 Abe S. Support Vector Machines for Pattern Classification. New York. Springer-Verlag; 2005.

20 Yao Y, Frasconi P, Pontil M. Fingerprint classification with combinations of support vector machine. 3rd International Conference of Audio and Video-Based Person Authentication (AVBPA 2001); 2001 June; Halmstad, Sweden. pp. 253-259.

21 Mehta SS, Lingayat NS. Development of SVM based classification techniques for the delineation of wave components in 12-lead electrocardiogram. Biomedical Signal Processing and Control 2008 October;3(4):341-349.

22 Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T. Prediction of bond dissociation enthalpy of antioxidant phenols by support vector machine. Journal of Molecular Graphics and Modelling 2008 September;27(2):188-196.

23 ASK Healthcare Consultants. [cited 2009 Mar 3]; Available from: URL : http://askhealthcareconsultants.com/heart-problems.php.

24 WEKA, The University of Waikato. [cited 2009 Jan 20]; Available from: URL : http://www.cs.waikato.ac.nz/ml/weka.

25 Otukei JR, Blaschke T. Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms. International Journal of Applied Earth Observation and Geoinformation 2010;12:S27-S31.

26 Correa M, Bielza C, Pamies-Teixeira J. Comparison of Bayesian networks and artificial neural networks for quality detection in a machining process. Expert Systems with Applications 2009;36(3):7270-7279.

27 Verma B, McLeod P, Klevansky A. A novel soft cluster neural network for the classification of suspicious areas in digital mammograms. Pattern Recognition 2009;42(9):1845-1852.

28 Witten IH, Frank E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers; 2000.

29 Embrechts MJ, Szymanski B, Sternickel K, Naenna T, Bragaspathi R. Use of Machine Learning for Classification of Magnetocardiograms. IEEE Proceeding of System Man Cybernetics. 2003;2:1400-1405.

30 Naenna T, Embrechts MJ, Szymanski B. Automated Magnetocardiogram Classifications with Self-Organizing Maps (SOMs). IEEE Proceeding of TENCON, 2004.

31 Comes B, Kelemen A. Probabilistic neural network classification for microarraydata. IEEE 2003:1714-1716.

32 Cross Industry Standard Process for Data Minng [cited 2010 Apr 27]; Available from: URL : http://www.crisp-dm.org/.

33 Froelicher V, Shetler K, Ashley E. Better decisions through science: Exercise testing scores. Progress in Cardiovascular Diseases 2002 Mar-Apr;44(5):395-414.

34 Vibha L, Harshavardhan GM, Pranaw K, P Deepa Shenoy, Venugopal KR, Patnaik LM. Classification of Mammograms Using Decision Trees. Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS 2006); 2006 Dec 11-13; Delhi, India.

35 Bradley A. The use of area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition 1997;30(7):1145-1159.

36 Tantimongcolwat T, Naenna T, Isarankura-Na-Ayudhya C, Embrechts M, Prachayasittikul V. Identification of ischemic heart disease via machine learning analysis on magnetocardiograms. Computers in Biology and Medicine 2008;38:817-825.

37 Ergun U. The Classification of Obesity Disease in Logistic Regression and Neural Network Methods. Journal of Medical Systems 2009;33:67-72.

38 Lavrac N. Selected techniques for data mining in medicine. Artificial Intelligence in Medicine 1999;16:3-23.

39 Cios KJ, Moore GW. Uniqueness of medical data mining. Artificial Intelligence in Medicine 2002:1303-1311.

40 Burke HB, Goodman PH, Rosen DB, Henson DE, Weinstein JN, Harrell Jr FE, et al. Artificial neural networks improve the accuracy of cancer survival prediction. Cancer Cytopathology 1997;79:857-862.

41 Fenici R, Brisinda D, Meloni AM, Sternickel K, Fenici P. Clinical Validation of Machine Learning for Automatic Analysis of Multichannel Magnetocardiography. Functional Imaging and Modeling of the Heart 2005;3504:143-152.

42 Vibha L, Harshavardhan GM, Pranaw K, P Deepa Shenoy, Venugopal KR, Patnaik LM. Classification of Mammograms Using Decision Trees. Proceedings of the 10th International Database Engineering and Applications Symposium (IDEAS 2006); 2006 Dec 11-13; Delhi, India.

43 Polat K, Gunes S. Breast cancer diagnosis using least square support vector machine. Digital Signal Processing 2007 July;17(4):694-701.

44 Mobley BA, Schecter E, Moore WE, McKee PA, Eichner JE. Prediction of coronary artery stenosis by artificial neural network. Artificial Intelligence in Medicine 2000 Mar;18(3):187-203.

45 Temurtas H, Yumusak N, Temurtas F. A comparative study on diabetes disease diagnosis using neural networks. Expert Systems with Applications 2009 May;36(4):8610-8615.

46 Berul CI, Sweeten TL, Dubin AM, Sham MJ, Vetter VL. Use of the rate-corrected JT interval for prediction of repolarization abnormalities in children. The American journal of cardiology 1994;74(12):1254-1257.

47 On K, Watanabe S, Yamada S, Takeyasu N, Nakagawa Y, Nishina H et al. Integral value of JT interval in magnetocardiography is sensitive to coronary stenosis and improves soon after coronary revascularization. Official journal of the Japanese circulation society 2007;71(10):1586-1592.

48 Sato M, Terada Y, Mitsui T, Miyashita T, Kandori A. Tsukada K. Detection of myocardial ischemia by magnetocardiogram using 64-channel SQUID system. In Nenonen J, Imoniemi RJ, Katila T, editors. Proceeding 12th International Conference on Biomagnetism; Espoo: Helsinki University Technology; 2001. p.523-526.

49 Kotsiantis SB. Supervised Machine Learning: A Review of Classification Techniques. Informatica 2007;31:249-268.

50 Chou SM, Lee TS, Shao YE, Chen IF. Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines. Expert Systems with Applications 2004;27(1):133–42.

51 Green M, Bjork J, Forberg J, Ekelund U, Edenbrandt L, Ohlsson M. Comparison between neural network and multiple logistic regression to predict acute coronary syndrome in the emergency room. Artificial Intelligence in Medicine 2006;38(3):305-318.

52 Wang D, Larder B, Rewell A, Montaner J, Harrigan R, Wolf FD, Lange J, et al. A comparison of three computational modeling methods for the prediction of virological response to combination HIV therapy. Artificial Intelligence in Medicine 2009;47(1):63-74.

53 Amendolia SR, Cossu G, Ganadu ML, Golosio B, Masala GL, Mura GM. A comparative study of K-Nearest Neighbour, Support Vector Machine and Multi-Layer Perceptron for Thalassemia screening. Chemometrics and Intelligent Laboratory Systems 2003;69(1-2):13-20.

54 Wu X, Kumar V, Quinlan R, Ghosh J, Yang Q, Motoda H, McLachlan GJ, et al. Top 10 Algorithms in Data Mining. Knowledge and Information Systems 2008;14:1-37.

APPENDIX

The optimal parameter of back propagation neural network

-Hidden node

HN	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	Average
0	0	0	0	0	0	0	0	0	0	0	0
1	0.743	0.270	0.499	0.570	0.599	0.403	0.358	0.252	0.570	0.626	0.489
2	0.170	0.244	0.250	0.145	0.124	0.413	0.089	0.221	0.245	0.163	0.206
3	0.395	0.242	0.160	0.057	0.274	0.176	0.508	0.231	0.186	0.152	0.238
4	0.362	0.186	0.219	0.237	0.142	0.150	0.117	0.232	0.128	0.165	0.194
5	0.111	0.245	0.096	0.107	0.097	0.175	0.131	0.137	0.140	0.158	0.140
6	0.256	0.144	0.120	0.090	0.429	0.129	0.128	0.244	0.113	0.162	0.181
7	0.289	0.098	0.315	0.111	0.096	0.130	0.096	0.122	0.134	0.113	0.150
8	0.384	0.470	0.121	0.031	0.096	0.110	0.115	0.097	0.112	0.161	0.170
9	0.242	0.151	0.102	0.087	0.093	0.119	0.119	0.135	0.098	0.165	0.131
10	0.095	0.116	0.172	0.065	0.094	0.117	0.112	0.372	0.076	0.128	0.135
11	0.108	0.175	0.063	0.032	0.119	0.145	0.110	0.103	0.112	0.143	0.111
12	0.076	0.109	0.138	0.137	0.102	0.117	0.165	0.026	0.079	0.107	0.106
13	0.166	0.125	0.079	0.080	0.111	0.339	0.121	0.089	0.082	0.264	0.145
14	0.089	0.068	0.395	0.168	0.121	0.110	0.146	0.054	0.163	0.192	0.150
15	0.111	0.105	0.110	0.118	0.093	0.105	0.140	0.068	0.116	0.111	0.108
16	0.131	0.121	0.115	0.066	0.093	0.067	0.171	0.093	0.098	0.099	0.105
17	0.116	0.096	0.099	0.058	0.164	0.149	0.159	0.050	0.114	0.086	0.109
18	0.070	0.122	0.054	0.133	0.085	0.079	0.104	0.191	0.127	0.068	0.103
19	0.089	0.076	0.125	0.075	0.072	0.118	0.116	0.081	0.120	0.140	0.101
20	0.121	0.087	0.076	0.065	0.117	0.127	0.087	0.128	0.115	0.069	0.099
21	0.029	0.137	0.120	0.133	0.070	0.155	0.198	0.055	0.108	0.087	0.109
22	0.095	0.150	0.103	0.082	0.069	0.109	0.094	0.053	0.151	0.108	0.101
23	0.130	0.126	0.124	0.088	0.105	0.136	0.135	0.030	0.075	0.130	0.108
24	**0.095**	**0.117**	**0.048**	**0.065**	**0.043**	**0.096**	**0.097**	**0.092**	**0.076**	**0.154**	**0.089**
25	0.068	0.112	0.100	0.064	0.069	0.137	0.114	0.045	0.099	0.124	0.093
26	0.072	0.129	0.170	0.092	0.119	0.112	0.158	0.092	0.098	0.124	0.117
27	0.080	0.101	0.151	0.113	0.117	0.104	0.106	0.044	0.130	0.106	0.105
28	0.095	0.105	0.114	0.136	0.089	0.156	0.131	0.063	0.068	0.104	0.106
29	0.067	0.103	0.114	0.094	0.110	0.166	0.167	0.108	0.118	0.137	0.118
30	0.285	0.158	0.129	0.107	0.100	0.128	0.092	0.128	0.143	0.111	0.138
31	0.082	0.132	0.106	0.076	0.133	0.108	0.158	0.142	0.134	0.121	0.119
32	0.311	0.077	0.096	0.092	0.123	0.092	0.155	0.100	0.128	0.180	0.135
33	0.162	0.093	0.061	0.107	0.111	0.115	0.078	0.109	0.131	0.093	0.106
34	0.111	0.131	0.104	0.122	0.057	0.104	0.158	0.025	0.092	0.157	0.106
35	0.126	0.158	0.084	0.091	0.118	0.123	0.105	0.123	0.056	0.174	0.116
36	0.129	0.131	0.077	0.082	0.100	0.127	0.124	0.025	0.070	0.143	0.101
37	0.094	0.113	0.093	0.082	0.100	0.110	0.114	0.138	0.142	0.147	0.113
38	0.104	0.141	0.100	0.111	0.133	0.150	0.131	0.107	0.104	0.094	0.118
39	0.076	0.052	0.089	0.059	0.145	0.097	0.196	0.101	0.127	0.180	0.112
40	0.117	0.107	0.103	0.061	0.125	0.057	0.167	0.041	0.101	0.123	0.100
41	0.119	0.102	0.117	0.053	0.125	0.123	0.164	0.104	0.119	0.100	0.113
42	0.126	0.058	0.068	0.074	0.080	0.104	0.160	0.119	0.131	0.134	0.105
43	0.098	0.111	0.127	0.084	0.079	0.085	0.124	0.116	0.095	0.152	0.107
44	0.079	0.147	0.057	0.120	0.132	0.117	0.187	0.147	0.105	0.146	0.124
45	0.145	0.127	0.162	0.103	0.123	0.106	0.198	0.049	0.089	0.127	0.123
46	0.074	0.151	0.120	0.133	0.136	0.067	0.147	0.073	0.168	0.110	0.118
47	0.080	0.107	0.100	0.091	0.099	0.107	0.107	0.146	0.149	0.165	0.115
48	0.133	0.132	0.077	0.096	0.105	0.086	0.113	0.149	0.126	0.156	0.117
49	0.090	0.118	0.093	0.109	0.141	0.131	0.102	0.106	0.132	0.139	0.116
50	0.100	0.075	0.118	0.075	0.146	0.122	0.136	0.113	0.131	0.094	0.111

-Learning epoch (Training set)

Epoch	**Fold1**	**Fold2**	**Fold3**	**Fold4**	**Fold5**	**Fold6**	**Fold7**	**Fold8**	**Fold9**	**Fold10**	**Average**
0	0	0	0	0	0	0	0	0	0	0	0
1	3.115	1.743	1.294	1.999	1.380	1.130	1.243	1.838	1.984	0.930	1.666
50	0.709	0.605	0.409	0.611	0.431	0.709	0.635	0.937	0.659	0.585	0.629
100	**0.498**	**0.518**	**0.364**	**0.537**	**0.432**	**0.515**	**0.472**	**0.423**	**0.418**	**0.391**	**0.457**
150	0.441	0.421	0.381	0.409	0.380	0.442	0.466	0.307	0.371	0.429	0.405
200	0.396	0.337	0.267	0.356	0.375	0.325	0.370	0.281	0.386	0.329	0.342
250	0.220	0.330	0.305	0.373	0.347	0.271	0.227	0.322	0.355	0.324	0.307
300	0.227	0.352	0.240	0.381	0.291	0.334	0.365	0.227	0.268	0.334	0.302
350	0.289	0.240	0.231	0.302	0.230	0.098	0.319	0.227	0.218	0.266	0.242
400	0.262	0.244	0.147	0.275	0.223	0.115	0.189	0.248	0.164	0.261	0.213
450	0.277	0.152	0.400	0.203	0.258	0.204	0.216	0.235	0.200	0.193	0.234
500	0.215	0.171	0.207	0.275	0.150	0.231	0.190	0.217	0.139	0.288	0.208
550	0.200	0.195	0.133	0.196	0.196	0.152	0.239	0.178	0.168	0.155	0.181
600	0.216	0.195	0.151	0.242	0.183	0.242	0.231	0.195	0.186	0.223	0.206
650	0.182	0.172	0.118	0.226	0.191	0.127	0.138	0.193	0.166	0.251	0.177
700	0.178	0.125	0.094	0.198	0.091	0.129	0.146	0.172	0.117	0.103	0.135
750	0.158	0.153	0.171	0.142	0.152	0.172	0.144	0.133	0.114	0.349	0.169
800	0.097	0.112	0.096	0.192	0.160	0.111	0.146	0.130	0.064	0.205	0.131
850	0.175	0.110	0.069	0.150	0.102	0.382	0.170	0.161	0.102	0.368	0.179
900	0.190	0.143	0.072	0.148	0.116	0.133	0.161	0.081	0.115	0.154	0.131
950	0.109	0.117	0.076	0.075	0.117	0.134	0.093	0.113	0.114	0.141	0.109
1000	0.088	0.130	0.027	0.100	0.133	0.048	0.114	0.085	0.094	0.087	0.091

-Learning epoch (Validation set)

Epoch	**Fold1**	**Fold2**	**Fold3**	**Fold4**	**Fold5**	**Fold6**	**Fold7**	**Fold8**	**Fold9**	**Fold10**	**Average**
0	0	0	0	0	0	0	0	0	0	0	0
1	2.806	1.662	1.276	2.108	1.029	1.634	1.204	1.883	1.613	0.827	1.604
50	1.631	0.710	0.686	0.610	1.146	0.784	0.942	0.690	1.042	0.996	0.924
100	**0.894**	**0.866**	**0.667**	**0.627**	**0.948**	**0.694**	**1.034**	**0.955**	**0.936**	**1.129**	**0.875**
150	0.732	0.882	0.715	1.000	1.207	0.945	0.749	1.232	0.845	0.800	0.911
200	0.593	1.063	1.102	0.500	1.120	1.212	1.115	1.026	1.120	1.149	1.000
250	1.126	0.995	0.764	0.828	0.964	0.946	0.711	1.180	0.886	0.888	0.929
300	1.573	1.031	1.002	1.082	1.185	0.788	1.019	1.250	0.936	0.920	1.079
350	0.871	1.101	0.890	0.991	1.085	1.064	0.780	1.296	1.202	0.924	1.020
400	0.526	1.063	1.234	0.869	0.899	1.467	0.717	1.274	1.441	0.987	1.048
450	0.592	1.036	0.863	1.168	1.183	1.026	1.171	1.404	1.145	1.290	1.088
500	0.314	0.949	1.000	0.889	1.300	0.938	0.731	1.551	1.354	1.004	1.003
550	0.370	1.133	1.027	1.162	1.127	1.084	0.743	1.433	1.210	1.172	1.046
600	0.316	1.082	1.007	1.055	0.930	0.958	0.966	1.340	1.329	0.772	0.976
650	0.862	0.995	1.100	0.752	1.127	1.048	0.763	1.308	1.064	0.688	0.971
700	0.692	1.746	0.722	0.768	1.164	1.210	0.666	1.321	1.404	0.798	1.049
750	0.839	1.211	0.953	0.692	1.025	0.958	0.675	1.336	1.150	2.185	1.102
800	0.920	1.130	0.839	0.924	1.149	1.046	0.976	1.508	1.383	1.181	1.106
850	0.712	1.186	1.159	0.863	1.065	1.293	0.631	1.407	1.343	1.150	1.081
900	0.442	1.061	1.290	1.351	1.020	1.036	0.783	1.449	1.329	1.261	1.102
950	0.587	1.202	0.965	0.405	1.145	0.932	0.873	1.496	1.228	0.933	0.976
1000	0.528	1.164	1.145	1.171	1.272	0.870	0.700	1.270	1.193	1.232	1.054

-Momentum and Learning rate

Mt LR	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	0.470	0.426	0.410	0.389	0.365	0.398	**0.325**	0.327	0.353	0.908
0.2	0.504	0.435	0.383	0.416	0.385	0.413	0.369	0.358	0.378	0.851
0.3	0.507	0.441	0.451	0.426	0.392	0.357	0.380	0.331	0.377	0.894
0.4	0.476	0.474	0.436	0.414	0.413	0.392	0.372	0.365	0.380	0.992
0.5	0.497	0.464	0.433	0.486	0.416	0.422	0.417	0.379	0.417	0.881
0.6	0.493	0.448	0.443	0.439	0.393	0.400	0.381	0.350	0.373	0.804
0.7	0.498	0.487	0.442	0.435	0.510	0.375	0.451	0.401	0.442	1.037
0.8	0.488	0.451	0.426	0.390	0.426	0.400	0.378	0.361	0.415	0.972
0.9	0.531	0.452	0.463	0.419	0.382	0.414	0.405	0.370	0.402	0.815
1.0	0.475	0.454	0.477	0.422	0.413	0.388	0.421	0.349	0.415	0.948

The optimal parameter of bayesian neural network

-Hidden node

HN	**Fold 1**	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Fold 6**	**Fold 7**	**Fold 8**	**Fold 9**	**Fold10**	**Average**
0	0	0	0	0	0	0	0	0	0	0	0
1	0.728	0.685	0.680	0.717	0.739	0.696	0.685	0.646	0.723	0.712	0.701
2	0.628	0.696	0.623	0.653	0.628	0.608	0.603	0.646	0.644	0.691	0.642
3	0.640	0.616	0.599	0.596	0.628	0.605	0.603	0.646	0.646	0.638	0.622
4	0.649	0.616	0.611	0.607	0.616	0.610	0.628	0.636	0.619	0.614	0.620
5	0.619	0.604	0.599	0.619	0.628	0.603	0.603	0.633	0.611	0.600	0.612
6	0.640	0.658	0.611	0.607	0.616	0.604	0.603	0.599	0.631	0.602	0.617
7	0.634	0.616	0.599	0.606	0.620	0.611	0.603	0.623	0.613	0.604	0.613
8	0.619	0.593	0.587	0.607	0.624	0.603	0.627	0.611	0.603	0.615	0.609
9	0.603	0.637	0.587	0.606	0.627	0.604	0.615	0.635	0.600	0.623	0.614
10	0.605	0.604	0.594	0.633	0.616	0.604	0.603	0.623	0.600	0.612	0.609
11	**0.604**	**0.609**	**0.586**	**0.630**	**0.612**	**0.604**	**0.599**	**0.611**	**0.599**	**0.599**	**0.605**
12	0.640	0.606	0.596	0.607	0.615	0.591	0.628	0.624	0.599	0.623	0.613
13	0.591	0.615	0.586	0.606	0.591	0.604	0.628	0.623	0.611	0.602	0.606
14	0.631	0.628	0.597	0.606	0.628	0.603	0.603	0.635	0.611	0.614	0.616
15	0.640	0.603	0.586	0.606	0.620	0.591	0.603	0.623	0.600	0.612	0.608
16	0.616	0.615	0.587	0.618	0.604	0.603	0.615	0.623	0.611	0.615	0.611
17	0.633	0.603	0.586	0.619	0.621	0.603	0.603	0.623	0.630	0.644	0.617
18	0.617	0.614	0.623	0.606	0.591	0.621	0.603	0.623	0.611	0.611	0.612
19	0.616	0.616	0.611	0.606	0.615	0.603	0.608	0.623	0.611	0.623	0.613
20	0.624	0.615	0.592	0.606	0.616	0.603	0.603	0.612	0.611	0.635	0.612
21	0.616	0.615	0.594	0.606	0.604	0.604	0.603	0.623	0.623	0.617	0.611
22	0.658	0.615	0.586	0.606	0.628	0.603	0.603	0.635	0.611	0.624	0.617
23	0.616	0.616	0.599	0.606	0.607	0.604	0.603	0.623	0.611	0.611	0.610
24	0.616	0.608	0.591	0.606	0.637	0.603	0.603	0.680	0.615	0.623	0.618
25	0.616	0.603	0.599	0.606	0.616	0.603	0.603	0.623	0.611	0.611	0.609
26	0.640	0.616	0.606	0.606	0.628	0.603	0.628	0.611	0.611	0.611	0.616
27	0.616	0.604	0.586	0.606	0.625	0.605	0.615	0.611	0.611	0.611	0.609
28	0.616	0.615	0.623	0.618	0.621	0.603	0.615	0.623	0.611	0.612	0.616
29	0.616	0.615	0.586	0.606	0.615	0.603	0.615	0.623	0.611	0.599	0.609
30	0.603	0.603	0.599	0.618	0.620	0.603	0.603	0.669	0.629	0.611	0.616
31	0.663	0.616	0.599	0.606	0.628	0.604	0.628	0.611	0.611	0.647	0.621
32	0.628	0.651	0.623	0.606	0.616	0.603	0.603	0.623	0.611	0.623	0.619
33	0.616	0.643	0.586	0.618	0.620	0.605	0.628	0.623	0.613	0.623	0.618
34	0.606	0.592	0.611	0.618	0.652	0.603	0.628	0.623	0.611	0.624	0.617
35	0.628	0.612	0.615	0.618	0.604	0.603	0.603	0.623	0.611	0.612	0.613
36	0.628	0.652	0.609	0.618	0.616	0.604	0.615	0.623	0.611	0.635	0.621
37	0.640	0.615	0.598	0.618	0.640	0.603	0.615	0.623	0.615	0.646	0.621
38	0.606	0.616	0.609	0.607	0.626	0.603	0.628	0.623	0.621	0.623	0.616
39	0.628	0.651	0.623	0.630	0.605	0.616	0.615	0.669	0.618	0.611	0.627
40	0.616	0.640	0.623	0.606	0.618	0.605	0.615	0.680	0.611	0.623	0.624
41	0.663	0.615	0.610	0.675	0.603	0.623	0.603	0.611	0.618	0.623	0.624

HN	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	Average
42	0.616	0.651	0.592	0.606	0.680	0.603	0.603	0.680	0.614	0.623	0.627
43	0.616	0.615	0.646	0.642	0.630	0.603	0.615	0.623	0.631	0.623	0.624
44	0.616	0.615	0.599	0.618	0.646	0.603	0.603	0.669	0.611	0.611	0.619
45	0.617	0.615	0.598	0.618	0.616	0.603	0.615	0.623	0.619	0.635	0.616
46	0.616	0.603	0.617	0.606	0.628	0.603	0.628	0.623	0.611	0.623	0.616
47	0.663	0.617	0.623	0.606	0.628	0.603	0.615	0.623	0.611	0.611	0.620
48	0.603	0.663	0.646	0.630	0.640	0.603	0.603	0.623	0.612	0.623	0.625
49	0.628	0.663	0.623	0.606	0.640	0.603	0.628	0.623	0.611	0.623	0.625
50	0.603	0.615	0.623	0.606	0.628	0.619	0.615	0.623	0.611	0.611	0.616

-Learning epoch (Training set)

Epoch	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
0	0	0	0	0	0	0	0	0	0	0	0
1	0.925	0.895	0.768	0.839	0.881	0.789	0.771	0.885	0.971	0.848	0.857
50	**0.717**	**0.676**	**0.667**	**0.715**	**0.678**	**0.697**	**0.683**	**0.644**	**0.668**	**0.704**	**0.685**
100	0.724	0.634	0.664	0.655	0.650	0.642	0.683	0.637	0.687	0.645	0.662
150	0.653	0.632	0.637	0.664	0.624	0.645	0.684	0.643	0.639	0.624	0.645
200	0.649	0.624	0.620	0.650	0.644	0.638	0.644	0.634	0.664	0.659	0.642
250	0.640	0.618	0.650	0.647	0.629	0.619	0.647	0.618	0.628	0.608	0.630
300	0.634	0.625	0.618	0.643	0.602	0.617	0.648	0.619	0.610	0.613	0.623
350	0.635	0.642	0.606	0.642	0.602	0.614	0.629	0.606	0.630	0.606	0.621
400	0.647	0.612	0.616	0.642	0.651	0.617	0.630	0.613	0.606	0.604	0.624
450	0.640	0.611	0.603	0.653	0.597	0.613	0.645	0.611	0.593	0.603	0.617
500	0.637	0.607	0.604	0.616	0.604	0.612	0.647	0.628	0.652	0.602	0.621
550	0.630	0.606	0.604	0.632	0.624	0.611	0.624	0.597	0.591	0.601	0.612
600	0.624	0.616	0.604	0.638	0.629	0.611	0.615	0.606	0.628	0.600	0.617
650	0.626	0.616	0.603	0.636	0.592	0.635	0.669	0.593	0.593	0.601	0.616
700	0.613	0.605	0.616	0.642	0.618	0.623	0.612	0.612	0.616	0.603	0.616
750	0.612	0.604	0.603	0.597	0.603	0.611	0.661	0.615	0.611	0.600	0.612
800	0.625	0.605	0.603	0.631	0.592	0.611	0.647	0.605	0.616	0.599	0.613
850	0.624	0.611	0.603	0.637	0.592	0.611	0.635	0.592	0.591	0.599	0.610
900	0.642	0.592	0.603	0.631	0.591	0.658	0.647	0.592	0.591	0.600	0.615
950	0.612	0.604	0.603	0.620	0.591	0.611	0.646	0.615	0.628	0.599	0.613
1000	0.666	0.613	0.615	0.616	0.591	0.623	0.646	0.592	0.590	0.587	0.614

-Learning epoch (Validation set)

Epoch	Fold1	Fold2	Fold3	Fold4	Fold5	Fold6	Fold7	Fold8	Fold9	Fold10	Average
0	0	0	0	0	0	0	0	0	0	0	0
1	0.902	0.915	0.737	0.890	1.064	0.678	0.912	1.032	0.747	0.933	0.881
50	**0.589**	**0.723**	**0.875**	**0.788**	**0.776**	**0.720**	**0.824**	**1.010**	**0.711**	**0.792**	**0.781**
100	0.638	0.718	0.883	0.831	0.843	0.864	0.821	0.987	0.860	0.866	0.831
150	0.595	0.710	0.906	0.843	0.887	0.738	0.882	0.945	0.754	0.854	0.811
200	0.610	0.712	0.860	0.714	0.940	0.737	0.798	1.005	1.023	0.757	0.816
250	0.602	0.709	0.923	0.772	0.906	0.844	0.776	1.036	0.844	0.927	0.834
300	0.652	0.708	0.881	0.841	0.813	0.823	0.836	1.027	0.884	0.765	0.823
350	0.645	0.709	0.886	0.853	0.792	0.881	0.762	1.063	0.958	0.956	0.851
400	0.650	0.710	0.975	0.667	0.764	0.866	0.743	1.038	0.822	0.925	0.816
450	0.659	0.708	0.891	0.777	0.744	0.880	0.756	1.068	0.886	0.980	0.835
500	0.649	0.712	0.883	0.590	0.831	0.895	0.842	0.994	1.056	0.993	0.844
550	0.655	0.709	0.885	0.608	0.883	0.873	0.748	1.135	0.883	0.958	0.834
600	0.573	0.708	0.872	0.739	0.785	0.892	0.665	1.067	0.986	0.946	0.824
650	0.584	0.708	0.907	0.729	0.825	0.697	0.782	1.141	0.876	0.997	0.824
700	0.540	0.709	0.922	0.679	0.880	0.824	0.731	1.050	0.862	1.041	0.824
750	0.593	0.718	0.883	0.706	0.813	0.843	0.827	1.052	0.890	0.988	0.831
800	0.731	0.708	0.906	0.602	0.866	0.885	0.845	1.074	0.994	0.990	0.860
850	0.653	0.708	0.888	0.607	0.792	0.889	0.809	1.124	0.828	1.023	0.832
900	0.654	0.760	0.884	0.668	0.830	0.660	0.845	1.156	0.838	0.994	0.829
950	0.556	0.711	0.911	0.598	0.845	0.881	0.845	1.059	0.756	0.988	0.815
1000	0.535	0.707	0.989	0.685	0.806	0.874	0.845	1.173	0.873	0.984	0.847

-Momentum and Learning rate

Mt LR	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	0.688	0.694	0.682	0.685	0.684	**0.681**	0.735	0.725	0.772	0.904
0.2	0.697	0.699	0.691	0.718	0.765	0.741	0.740	0.765	0.734	0.876
0.3	0.710	0.717	0.707	0.722	0.698	0.763	0.773	0.737	0.788	0.851
0.4	0.722	0.722	0.730	0.752	0.720	0.719	0.773	0.737	0.785	0.883
0.5	0.727	0.736	0.706	0.736	0.710	0.781	0.762	0.805	0.818	0.853
0.6	0.711	0.733	0.737	0.712	0.758	0.747	0.746	0.760	0.782	0.859
0.7	0.764	0.714	0.747	0.729	0.762	0.798	0.711	0.784	0.773	0.878
0.8	0.729	0.760	0.714	0.755	0.753	0.732	0.740	0.781	0.806	0.868
0.9	0.750	0.752	0.733	0.760	0.745	0.740	0.767	0.767	0.755	0.856
1.0	0.743	0.726	0.766	0.759	0.754	0.796	0.799	0.779	0.797	0.856

The optimal parameter of probabilistic neural network

-Spread

Spread Value	**Correct Classification (%)**										
	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Fold 6**	**Fold 7**	**Fold 8**	**Fold 9**	**Fold10**	**Average**
0.1	42.86	28.57	37.50	33.33	28.57	37.50	37.50	28.57	37.50	37.50	34.94
0.2	42.86	28.57	37.50	33.33	28.57	37.50	37.50	28.57	37.50	37.50	34.94
0.3	42.86	28.57	12.50	33.33	28.57	37.50	37.50	28.57	37.50	37.50	32.44
0.4	57.14	57.14	12.50	50.00	28.57	37.50	37.50	28.57	25.00	50.00	38.39
0.5	42.86	57.14	12.50	66.67	42.86	62.50	50.00	42.86	37.50	62.50	47.74
0.6	42.86	71.43	50.00	66.67	57.14	62.50	62.50	71.43	50.00	62.50	59.70
0.7	57.14	71.43	50.00	66.67	71.43	50.00	75.00	85.71	50.00	62.50	63.99
0.8	57.14	71.43	50.00	66.67	71.43	75.00	75.00	85.71	62.50	75.00	68.99
0.9	**71.43**	**71.43**	**50.00**	**66.67**	**71.43**	**87.50**	**75.00**	**85.71**	**62.50**	**75.00**	**71.67**
1.0	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.1	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.2	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.3	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.4	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.5	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.6	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.7	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.8	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
1.9	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.0	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.1	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.2	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.3	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.4	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.5	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.6	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.7	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.8	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
2.9	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.0	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.1	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.2	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.3	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.4	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.5	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.6	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.7	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.8	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
3.9	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.0	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67

Spread	Correct Classification (%)										
Value	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Fold 6	Fold 7	Fold 8	Fold 9	Fold10	Average
4.1	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.2	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.3	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.4	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.5	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.6	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.7	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.8	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
4.9	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67
5.0	71.43	71.43	50.00	66.67	71.43	87.50	75.00	85.71	62.50	75.00	71.67

Prediction of Ischemic Heart Disease

Testing data	Actual	Prediction					
		BPNN	BNN	PNN	Linear kernel SVM	Polynomial kernel SVM	RBF kernel SVM
1	Normal	Normal	Normal	IHD	Normal	Normal	Normal
2	Normal	IHD	Normal	Normal	Normal	Normal	Normal
3	IHD	IHD	IHD	IHD	IHD	IHD	Normal
4	IHD	IHD	IHD	IHD	IHD	IHD	IHD
5	IHD	IHD	IHD	IHD	IHD	IHD	Normal
6	IHD	IHD	IHD	IHD	IHD	IHD	Normal
7	Normal	IHD	Normal	IHD	IHD	IHD	IHD
8	IHD	IHD	IHD	Normal	IHD	IHD	IHD
9	IHD	IHD	IHD	IHD	IHD	IHD	IHD
10	IHD	IHD	IHD	IHD	Normal	IHD	Normal
11	Normal	Normal	IHD	IHD	Normal	IHD	Normal
12	Normal	Normal	Normal	Normal	Normal	Normal	Normal
13	IHD	IHD	IHD	IHD	Normal	IHD	Normal
14	Normal	Normal	IHD	IHD	Normal	IHD	Normal
15	IHD	IHD	IHD	IHD	IHD	IHD	Normal
16	Normal	IHD	IHD	IHD	Normal	IHD	Normal
17	Normal	Normal	Normal	Normal	Normal	Normal	Normal
18	Normal	Normal	Normal	Normal	Normal	Normal	Normal
19	Normal	Normal	Normal	Normal	Normal	Normal	Normal
20	Normal	Normal	Normal	Normal	Normal	Normal	Normal
21	Normal	IHD	Normal	IHD	Normal	IHD	Normal
22	Normal	Normal	Normal	Normal	Normal	Normal	Normal
23	IHD	Normal	IHD	IHD	Normal	IHD	Normal
24	IHD	IHD	IHD	Normal	Normal	Normal	Normal
25	Normal	Normal	IHD	Normal	IHD	IHD	IHD
26	Normal	Normal	IHD	IHD	Normal	IHD	Normal
27	Normal	Normal	IHD	IHD	Normal	IHD	Normal
28	Normal	IHD	IHD	IHD	IHD	IHD	Normal
29	Normal	Normal	Normal	Normal	Normal	Normal	Normal
30	IHD	IHD	IHD	IHD	IHD	IHD	Normal
31	Normal	Normal	IHD	IHD	Normal	IHD	Normal
32	Normal	Normal	Normal	Normal	Normal	Normal	Normal
33	IHD	IHD	IHD	IHD	IHD	IHD	IHD
34	Normal	IHD	IHD	Normal	IHD	IHD	IHD
35	Normal	IHD	IHD	IHD	Normal	IHD	Normal
36	IHD	Normal	IHD	IHD	IHD	Normal	Normal
37	IHD	IHD	IHD	IHD	IHD	IHD	Normal
38	IHD	IHD	IHD	IHD	IHD	IHD	IHD
39	IHD	IHD	IHD	IHD	IHD	IHD	Normal

Testing data	Actual	Prediction					
		BPNN	BNN	PNN	Linear kernel SVM	Polynomial kernel SVM	RBF kernel SVM
40	IHD	IHD	IHD	IHD	IHD	IHD	IHD
41	IHD	IHD	IHD	IHD	IHD	IHD	IHD
42	IHD	IHD	IHD	IHD	IHD	IHD	Normal
43	IHD	IHD	IHD	IHD	IHD	IHD	Normal
44	IHD	Normal	Normal	IHD	IHD	Normal	Normal
45	IHD	IHD	IHD	IHD	IHD	IHD	IHD
46	IHD	IHD	IHD	IHD	IHD	IHD	IHD
47	IHD	IHD	IHD	IHD	IHD	IHD	IHD
48	IHD	IHD	IHD	IHD	IHD	IHD	Normal
49	IHD	IHD	IHD	Normal	IHD	IHD	IHD
50	IHD	Normal	IHD	IHD	IHD	IHD	Normal
51	IHD	IHD	IHD	Normal	IHD	IHD	IHD

BIOGRAPHY

NAME	Mr.Yosawin Kangwanariyakul
DATE OF BIRTH	28 April 1979
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	King Mongkut's Institute of Technology Ladkrabang, 1997-2000: Bachelor of Engineering (Electronic Engineering) Mahidol University, 2005-2009: Master of Science (Technology of Information System Management)
HOME ADDRESS	1328/170, Bangkae, Bangkae, 10160, Thailand. E-mail: yosawin_s@hotmail.com