A New Incremental Classification Approach: Monitoring The Risk of Heart Disease

A NEW INCREMENTAL CLASSIFICATION APPROACH: MONITORING THE RISK OF HEART DISEASE

BY

SHIMA AGHTAR, B.Sc.

A thesis

submitted to the School of Graduate Studies

at McMaster University

in partial fulfillment of the requirements

for the degree

Master of Science

Master of Science (2012) McMaster University

(Computer Science) Hamilton, Ontario, Canada

TITLE:	A New Incremental Classification Approach: Monitoring The Risk of Heart Disease
AUTHOR:	Shima Aghtar Master of Science, (Computer Science) McMaster University, Hamilton, Ontario
SUPERVISOR:	Dr. Norm Archer
NUMBER OF PAGES:	xv, 67

Dedicated to my parents, Fatemeh Hoshyaripour and Gholmreza Aghtar

and

my husband, Farshad Moin-Manavi

Abstract

Medical decision support systems are one of the main applications for data mining and machine learning techniques. Most of these support systems involve solving a classification problem. Classification models can be generated by one of two types of learning classification algorithms: batch or incremental learning algorithms.

A batch (non-incremental) learning algorithm generates a classification model trained by using the complete available data. Examples of batch learning algorithms are: decision tree C4.5, k nearest neighbor, Bayesian neural network and multilayer perceptron neural network algorithms. However, an incremental learning algorithm generates a classification model trained incrementally through batches of training data. Examples of this are Learn++ and DWMV Learn++. Incremental learning algorithms are effective in problems in the healthcare domain where the training data become available periodically over time or where the size of database is very large. Incremental classification model is also able to capture dynamic health trends that are changing over time, as opposed to batch classification model based on a static large batch of data in time.

In the health care system, we consider heart disease a major cause death, and thus,

it is a domain requiring attention. Early screening of patients for heart disease before they actually have its symptoms could therefore be an effective solution for decreasing the risk of this disease. Classification techniques can be employed to recognize patients who are at high risk of developing heart disease in order to send them for further attention or treatment by specialists.

This work proposes an incremental learning algorithm, called modified DWMV Learn++, for primary care decision support that classifies patients into high risk and low risk, based on certain risk factors. This algorithm unlike DWMV Learn++ has no pre-assumption on distribution of dataset. The system uses this incremental learning algorithm for classification. This system has been tested and proven to have good performance using real-world patient clinical records.

Acknowledgements

This thesis would not have been possible without the guidance and the valuable assistance of several individuals in the preparation and completion of this study.

First and foremost, I would like to express my sincerest gratitude to Dr. Norm Archer for his supervision and his constructive advice on this thesis. All throughout, he has given me great motivation and encouragement, as well as his wisdom and patience, in dealing with the different challenges and getting the assistance of the different individuals, in order to develop the research discussed in this thesis.

I also would like to acknowledge Dr. Pace from the Department of Family Medicine at McMaster University and Dr. Moore from the Hamilton StoneChurch Clinic for their support in providing data for my research. Their data was used to test the different algorithms used in this research.

I have benefitted from the advices of Mehrdad Roham and Anait R Gabrielyan, who have always kindly granted me their time and assistance with my questions.

Finally, I would like to thank my parents for their unconditionally emotional support throughout my degree work, as well as my husband Farshad for his personal support and great patience during this masters program.

Notation and Abbreviations

BNN : Bayesian Neural Network - a classification model.

CDSS : Clinical Decision Support System - computer software which assists health givers with decision making tasks such as diagnosis.

CPT : Conditional Probability Table - a table that contains the probability of each state of the variables associated with neurons in a Bayesian network.

CVD : CardioVascular Disease - a type of disease that involve the heart or blood vessels (arteries and veins).

DBP : Diastolic Blood Pressure - minimum blood pressure during a heartbeat cycle.

DWMV : Dynamic Weighted Majority Voting - A method used for majority voting in an ensemble of classifiers.

ECG-LVH : Electrocardiography - left ventricular hypertrophy.

EMR : Electronic Medical Record - a computerized medical record used for recording patient medical and demographic information.

KDD : Knowledge Discovery in Databases - the process of automatically searching large volumes of data with the intent of finding existing patterns.

KNN : K Nearest Neighbor - a classification model.

LR : Logistic Regression - a predictive model.

MLP-NN : Multilayer Perceptron Neural Network - a classification model.

RBF : Radial Basic Function - a real-valued function whose value depends only on the distance from the origin.

SBP : Systolic Blood Pressure - maximum blood pressure attained during the heart-beat cycle.

SVM : Support Vector Machine - a supervised learning model used in machine learning.

WEKA : Waikato Environment for Knowledge Analysis - an open-source machine learning software repository.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Problem Statement

1.1 Introduction

Healthcare organizations aim to provide high quality healthcare at affordable cost to their communities. One way of lowering costs is to have early detection of incipient diseases or conditions, that also have effective interventions/treatments to alter disease course. Cardiovascular disease is an ideal example of such a condition. Many clinical decision support systems (CDSSs) have been developed to assist clinicians to make more accurate identification of diseases and to suggest well-timed screening for preventable diseases [43][18]. Cardiovascular diseases, which include heart disease, are recognized as one of the main causes of death [60] and a number of CDSSs have been developed that can be integrated with electronic medical record systems (EMRs) to help to improve the management of these and other chronic disease [31][3].

According to a report released by Statistics Canada [1], heart disease is the second main cause of death in Canada, totalling 22% of deaths in 2007. If it is possible to diagnose and provide preventive treatments for patients at risk from heart disease before obvious symptoms begin to appear, this can be an effective way to decrease fatality rates. Although universal screening for heart disease might substantially reduce the number of deaths associated with heart problems, this would be very costly to implement due to limited health care resources [21]. Selecting high-risk patients by screening is a more efficient solution to decrease the heart disease death rate. The main objective of the research in this thesis is to propose decision support that would assist in identifying patients at high risk for heart disease. Patients thus identified could be given more detailed assessments that would verify the level of their risk and would lead to physician provided regimens that would help in preventing further progression of their diseased condition.

The inspiration for the development of the research in this thesis came mainly from the Framingham research project [73] that tracked a large number of individuals over a long period of time to determine what measures related to their health would assist in predicting their likelihood of developing heart disease. After the data were collected, they were used to develop a well-known risk estimator for predicting the development of cardiovascular diseases, based on certain risk factors. The early version of the Framingham equation [39] identified heart disease risk factors that included age, sex, systolic blood pressure(SBP)/diastolic blood pressure (DBP), serum cholesterol, level of cigarette smoking (if any), glucose intolerance, and left ventricular hypertrophy (LVH). Later versions used a slightly different set of risk factors. Similarly

to the Framingham equation, we propose a decision support algorithm that uses the Framingham risk factors for determining patient risk of developing heart disease over a future time interval of five years.

There are currently several tools in use to assist clinicians with understanding an individuals risk of developing CVD (and the need for more intensive investigation and risk factor modification). The joint European Societies charts and the widely used Framingham score are such examples. However these formulas were validated using data from specific populations (UK and US populations respectively), and may be less accurate predictors of CVD risk in other jurisdictions. The Canadian Diabetic Association, Canadian Hypertension Education Program and the Canadian Dyslipidemia Guidelines (2009) recommend using the modified Framingham risk score to determine an individual's risk of developing CVD. An incremental learning algorithm that is based on the modified Framingham score will advantageously generate a more refined formula derived from the target population that more accurately identifies individuals at high risk of CVD from that population, compared to currently used calculations and formulas.

The objective of developing a decision support system is to assist physicians with estimating the heart disease risk of their patients. Such a system would be used to predict the risk of patients for developing heart disease and to notify physicians of cases with high risks. These predictions would help physicians to send patients at risk for more detailed screening using electro-cardiograms, stress tests, etc. A decision support system based on the algorithms proposed in this thesis would be trained

with information collected from patients who are known to have heart disease. Data mining algorithms employed in this system use this known data in order to learn from it and to estimate the risk of new patients with unknown prognosis of becoming ill with heart disease in the future. The patient parameters mentioned above are used to derive risk factor information that the algorithm can use to estimate this risk.

In order to simulate the real world, the records of 58 heart disease patients of Stonechurch clinic were used for testing this system. We also used the health information of patient records in the Canadian Heart Health Database [48] as a training set for the data mining classifiers that were tested. A proposed incremental classifier that extends an existing classifier DWMV Learn++ was developed for use as this system's learning algorithm. This classifier allows the system to incrementally improve its performance as new training data becomes available and , unlike the original one, this has no assumption on dataset distribution. This work was approved by the McMaster/Hamilton Health Sciences Research Ethic Board.

This thesis is divided into 5 chapters. The structure of the document is as follows:

- **Chapter 1** provides an introduction to the proposed system and its motivation.
- **Chapter 2** gives an overview of some related literature on the topic.
- **Chapter 3** describes two versions of the incremental Learn++ algorithm. This chapter also presents a proposed version of Learn++ called modified DWMV Learn++.

- **Chapter 4** describes the training and testing datasets. It also presents the method of data preprocessing that was employed before the data were used for training and testing.

- **Chapter 5** compares the effectiveness and the accuracy of batch data mining classifiers and the incremental Learn++ classifiers. It also describes some potential future work that could be developed from the research presented in this thesis.

Chapter 2

Literature Review

2.1 Framingham Equation

The Framingham algorithm is a widely used practical method for estimating the five and ten year risk of developing cardiovascular disease, for individuals between the ages of 30 and 74 years. This method is based on the analysis of data collected from the long-term continuous Framingham Heart Study. The data that were used to develop the algorithm was collected from residents of the town of Framingham, Massachusetts. The study started in 1948 with 5,209 individuals. The original Framigham equations were introduced in 1991 by Anderson et al [39] for predicting an individual's risk of developing coronary heart disease. This technique is employed for people who exhibit no cardiovascular disease symptoms at the time of examination. The algorithm's risk prediction is based on heart disease risk factors including age, gender, systolic blood pressure (SBP), diastolic blood pressure (DBP), total serum cholesterol, HDL cholesterol, smoking habits, the presence of diabetes, and ECG-LVH (Electrocardiography Left Ventricle Hypertrophy). The Framingham prediction tool is presented in the

form of equations and a scoring system, see appendix for more details. An update to the Framingham algorithm was proposed in 1998 by Wilson et al [76]. The updated version removed ECG-LVH from the risk factors in the model. A more complete version was proposed in 2008 to evaluate the risk of specific cardiovascular disease (CVD) events i.e., coronary heart disease, cerebrovascular disease, peripheral vascular disease, and heart failure [68]. This version also excluded ECG-LVH from the risk factors but added blood pressure.

The accuracy of the Framingham equation has been validated in several populations. According to a 2003 study [64] the Framingham equation is a good tool for estimating cardiovascular risk for New Zealanders. Grundy et. al [65] recommended this equation as a good risk estimator for use in the United States. An additional study, [36] showed that the Framingham system more precisely estimates the risk of populations in North America and Australia than it does for Europeans.

There are other risk prediction models that are based on the Framingham model such as the New Zealand risk tables [56] and the joint European Societies' charts [38]. Several alternative risk prediction models have been proposed including, for example: SCORE [55], a scoring system for use in Europe, ASSIGN [52], a risk score developed in Scotland, and PROCAM [16], a scoring prediction model developed for the German population. The accuracy of these models varies substantially across different populations, indicating that it is difficult to generalize the risk estimation algorithms from one population to another. For this reason, the research reported in this thesis attempted to use data obtained directly from the local target population to validate

the model that was developed.

2.2 Knowledge Discovery and Data Mining

Different definitions have been given in the literature for knowledge discovery, or data mining. Knowledge Discovery in Databases (KDD) is defined as the infusion of implicit, previously unknown, and valuable knowledge from large databases for decision making in real-world applications [27]. Fayyad et al. [74] define KDD as "the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". The KDD procedure is divided into the following steps [44]: 1) Selecting data sets, 2) Integrating the data sets, 3) Preprocessing and cleaning the data, 4) Developing models, 5) Choosing proper data mining algorithms, 6) Interpreting and visualizing the results, and 7) Testing and verifying the results.

From one viewpoint, data mining is an analogous term for KDD. From another viewpoint, data mining can be considered to be a fundamental step in the process of knowledge discovery. Data mining is categorized into two categories: descriptive and predictive. Descriptive data mining aims to derive new, nontrivial patterns to describe data and predictive data mining generates a model that uses some database variables to classify, predict or estimate unknown or future values of other variables. Machine learning techniques have been used for deriving knowledge from data and generalizing this knowledge for classification and prediction purposes. Inductive machine learning algorithms can be employed to construct models from training data. These models generalize the knowledge acquired from training data into unseen instances. The model is constructed and trained in either supervised or unsupervised

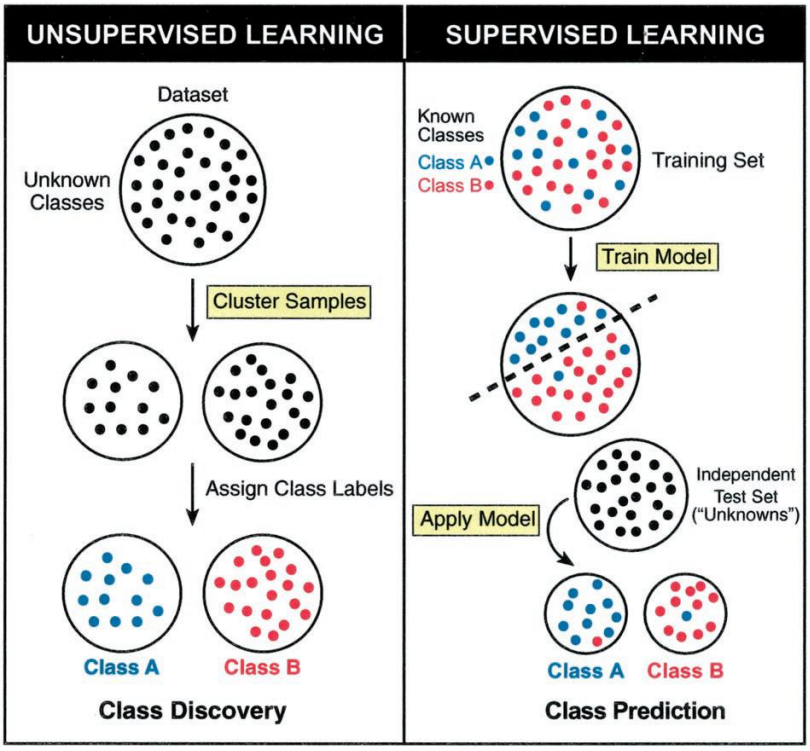


Figure 1: Supervised and Unsupervised Learning Methodology Adapted from Ramaswamy et al.(2002) [67]

learning mode, as demonstrated in Figure 1.

Supervised learning approaches generate learning models where the inputs are mapped to corresponding predefined labels. In other words, a supervised learning method attempts to classify an input instance into a specific label (class). However, unsupervised learning techniques attempt to generate models for unlabelled datasets. In unsupervised learning methods, similarities between instances are used in order to categorize an instance. Supervised or unsupervised learning models are trained through either a batch or incremental mode, based on the availability of a training data set.

2.3 Non-incremental (Batch) Supervised Learning Classifiers

Non-incremental classifiers belong to a category of classifiers which access the entire training data while constructing the learning model. These classifiers require a large amount of computation time and memory. Also, they are most suitable for prediction in environments where new data does not become available after the initial classifier training process. Various batch classifiers have been proposed in the literature, such as decision trees, neural networks, etc. The following section will introduce some of the non-incremental classifiers which were considered for use in this research.

2.3.1 Decision Tree Classifier

A decision tree [62] or classifier tree, Figure 2, is a hierarchical tree-structured model. At each internal node, an attribute's value is checked and the result of the test is represented by a branch or a subtree. Each leaf node specifies a class label. A decision tree is constructed from a training set by the divide-and-conquer method. If all instances belong to the same class label, the tree is a leaf node which prescribes the instances' class label. Otherwise, the instances are partitioned based on a chosen attribute test such as an information gain test. A decision tree is either pruned forward or backward to avoid overfitting [28]. Overfitting happens when a decision tree performs well on training instances and performs poorly on testing instances. The pruning forward (or pre-pruning) fashion involves halting unnecessary expansion of the tree's branches while building the tree. However, pruning backward (or post-pruning) constructs the tree first and then removes unreliable branches based

on certain statistical measurements.

Unlike some classifiers, such as neural networks, which perform like black boxes, a decision tree can be interpreted as "IF-THEN" rules which perform like glass boxes and are more easily understandable. This ease of interpretation makes the decision tree a popular classifier in many different domains such as expert systems, activity recognition [29], medical diagnosis [37],etc.

Several decision tree algorithms have been proposed in the literature including ID3 [34], C4.5 [35], C5 [6] and CART [41]. C4.5 is a descendant of the ID3 algorithm proposed by Quinlan (1993). Unlike ID3, C4.5 handles continuous and discrete attributes as well as attributes with missing values. In this thesis, we will use J48 which is a version of the C4.5 algorithm. This algorithm is implemented in the WEKA data mining tool. WEKA [45] is a free, open-source machine learning software repository including an Application Programming Interface (API). WEKA is implemented in Java by the University of Waikato, New Zealand. This software contains many data mining functions, ranging from pattern recognition to knowledge visualization. For this thesis, we used WEKA for comparing batch classifiers and implementing the incremental algorithms described later in the thesis.

2.3.2 K Nearest Neighbor Classifier

The K-Nearest Neighbor (KNN) classifier is a widely used classification model for real world classification problems. This classifier predicts the class label of a test instance based on its similarity to known samples. In other words, the classifier selects

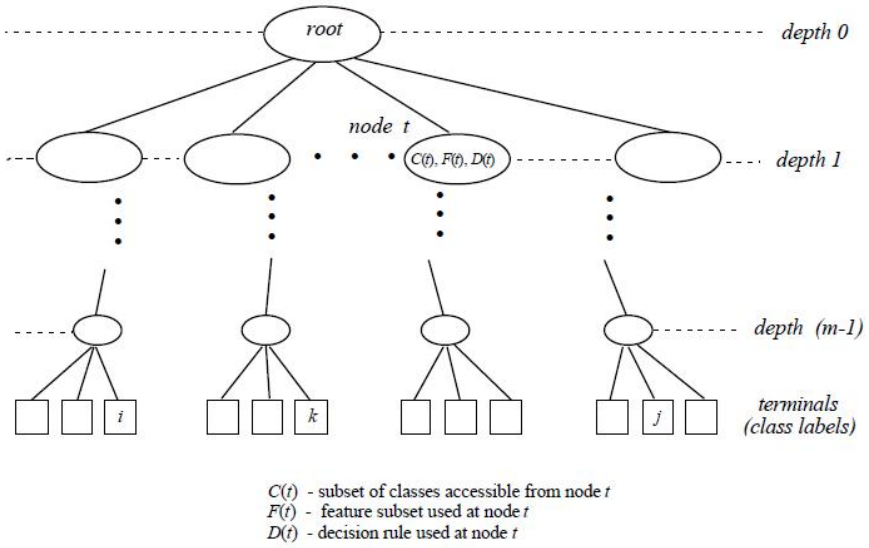


Figure 2: General Decision Tree Adapted from Safavian et al.(1991)[62]

a class label (C) for an instance (x) where the majority of the K nearest neighbors of training instances to x are categorized in class (C). Different distance functions, such as Euclidean and Manhattan functions, can be employed for determining the nearest neighbors. The most common distance function used for this classifier is the Euclidean distance function [28].

In previous studies, different refinements of this algorithm have been proposed to improve its accuracy and time complexity. The fuzzy K-nearest neighbor algorithm [46] assigns a weight to each of the k nearest neighbors based on the importance of their contributions in classifying an instance. The Nearest Cluster (NC) [23] algorithm is an augmented version of KNN. NC divides the training set into several clusters, and then assigns to each of their centres a label representing the majority of

the cluster members. The NC algorithm categorizes a new test instance to the class label of the closest cluster centre. NC is faster and more accurate than the original KNN algorithm.

The simplicity and good performance of this algorithm have encouraged its use by many researchers. Sarkar and Leong [50] showed that the KNN algorithm performed well for diagnosing breast cancer in a Wisconsin-Madison Breast Cancer study. Another application of KNN is discussed by Liao and Vemuri [42], where KNN is used for classifying a computer program as either normal or intrusive.

2.3.3 Naïve Bayes Classifier

The Naïve Bayes Classifier is a probabilistic classifier based on Bayes theorem. This classifier is so-named because it is based on a naive assumption about data, which is the independence of the attributes' distributions. Even though this assumption is unrealistic in real-world applications, this classifier has surprisingly good performance in the majority of classification problems. The probability of categorizing a new instance (e), defined by the attributes ($A_1A_2..A_n$), to class (C) is calculated as a posterior probability by the Bayes theorem:

$$\text{Posterior probability of C} : Pr(C|A_1A_2..A_n) = \frac{Pr(A_1A_2..A_n|C)*Pr(C)}{Pr(A_1A_2..A_n)}$$

The Naïve Bayes Classifier classifies the instance (e) to the value of C with the maximum $Pr(C|A_1A_2..A_n)$. This classifier has a few advantages. One of them is its comparatively low computational complexity which is simply O(nm), where "n" is

the number of instances and "m" is the number of classes. As well, this algorithm has a simple structure and high space efficiency [54].

2.3.4 Bayesian Neural Network Classifier

The Bayesian Neural Network (BNN), also known as the Belief Network or the Probabilistic Network encodes the joint probability distribution of a set of variables. This network is a directed acyclic graph where each node represents a random variable. The random variable refers to an actual attribute or a hidden variable shown as a relationship. Edges of the network show the conditional dependencies between random variables. For each node, a conditional probability table (CPT) contains the probability of each state of the variable for any possible combination of its parents' states. Learning in a Bayesian network involves the determination of network structures and the probability values in CPT.

BNN has been used for various applications. Pan et.al [77] used a Bayesian neural network ensemble for forecasting rainfall. Auld et. al [69] showed that the BNN classifier exhibited good performance for Internet traffic identification. Auld et. al. [70] compared the back-propagation neural networks, negative binomial regression, and Bayesian neural networks for predicting motor vehicle collisions, and concluded that BNN had better generalization performance than the other algorithms.

2.3.5 Multilayer Perceptron Neural Network Classifier

A Multilayer Perceptron (MLP), Figure 3, is a feed-forward neural network with one input layer, one output layer and one or more hidden layers with multiple nodes.

These layers are fully connected, meaning that the nodes in a layer (other than the output layer) are connected to all the nodes in the next layer with a corresponding weight applied to each connection. There are activation functions associated with each node which are linear for the nodes in the input and the output layers, and nonlinear for the nodes in the hidden layers. The most common nonlinear activation function used in this type of network is the sigmoid function. The sigmoid function is defined as:

$$S(t)=\frac{1}{1+e^{-t}}$$

The MLP network is trained with the backpropagation supervised learning algorithm [9]. The backpropagation algorithm adjusts the networks' weights in order to minimize the overall error function value.

The Multilayer Perceptron Neural Network is often used to solve pattern recognition, prediction and classification problems. According to the literature, this algorithm performs well in real world applications, ranging from marketing to medicine. It has been applied extensively to the medical domain for clinical diagnosis, image analysis and drug development. Hongmei et. al [32] used MLP to develop a decision support system for diagnosing heart disease. Their results showed that the MLP had an accuracy of over 90 % for heart disease diagnosis. Bourd'es et. al [75] compared the performance of MLP-NN with standard logistic regression (LR), and found that MLP-NN was more accurate than LR in predicting breast cancer.

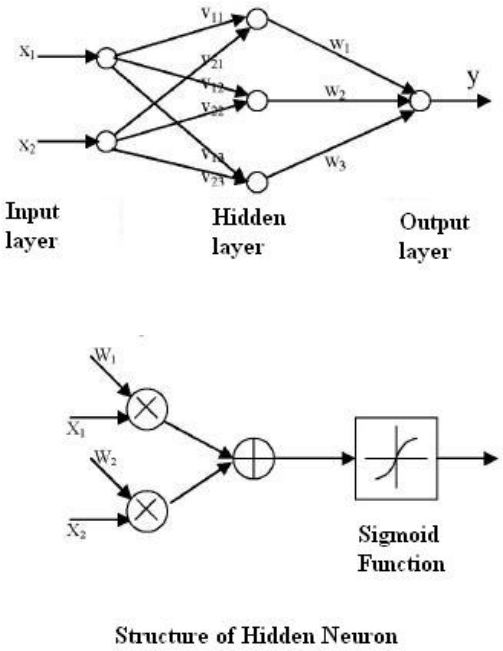


Figure 3: MLP Neural Network Structure Adapted from Vasantha et al.(2007)[33]

2.3.6 Radial Basic Function Neural Network Classifier

The Radial Basis Function (RBF) network is a single-hidden-layer feed-forward neural network with a non-linear RBF function in the hidden layer and a linear transfer function in the output layer. The Gaussian function is usually suggested as an activation function in the network's hidden units for pattern classification problems [49] [17]. The Gaussian function is defined as

$$\phi_j(X) = exp\lfloor -(X-\mu_j)^T \sum_j^{-1} (X-\mu_j) \rfloor$$

where j=1,...,L(the number of hidden units), X is the input feature vector, μ_j and $\sum_j^{-1}$ are the mean and the inverse of covariance matrix of the j^{th} Gaussian function.

These networks have been used in various real-world applications. For example, two studies[8] [26] showed that RBF neural networks performed well in face recognition applications. Also, Delican [78] proposed a decision tree based on this neural network with an Artificial Bee Colony algorithm (ABC) for diagnosing Parkinson's disease. The RBF algorithm has also been used for classification of protein sequences [72][4].

2.4 Incremental Supervised Learning Classifiers

Incremental supervised learning classifiers are classification models which learn from consecutively available data sets while retaining previously learned knowledge. Incremental classifiers gradually improve their performance using incremental learning algorithms. Polikar et al. [59] defined an incremental learning algorithm as: 1) an algorithm which is not allowed to access the previously used training data sets, and 2) it can be introduced to a new class at any time in a new dataset during training process.

As opposed to incremental learning algorithms which learn through several stages, non-incremental algorithms learn through only one stage by accessing the entire training set at once. Non-incremental training classifiers must be retrained from scratch each time new data become available. Since real-world environments often change as time goes on, there is less demand for non-incremental classifiers for use in prediction and classification, while incremental classifiers continue to play key roles in many real-world applications.

Learning novel information from newly available data while maintaining formerly

acquired knowledge brings the incremental learning up against the stability-plasticity dilemma [22][47]. A completely stable classifier preserves the knowledge it already has and refuses to acquire new information. On the other hand, a completely plastic classifier easily loses knowledge it has acquired as it is faced with new information. Incremental learning classification algorithms have to balance between these two issues in a manner such that they learn new knowledge while they preserve previously acquired valuable knowledge.

Several incremental learning algorithms have been proposed in the literature. One group of these algorithms implements incremental learning by changing the structure of the model during the learning process. A second group learns by adjusting the learning parameters of the incremental model. A third and final group uses a combination of these methods to implement the incremental learning process. ID4 [30] and its descendants, ID5 [12], ID5R [13] are incremental versions of the decision tree algorithm ID3. These algorithms build a decision tree by sequentially restructuring the decision tree as new training data instances arrive. Kidera et. al. [71] proposed an incremental learning algorithm composed of an ensemble of neural network classifiers, which they called the Resource Allocating Network with Long-Term Memory (RAN-LTM). A fixed number of classifiers are built when the first training set is applied, combined with a majority voting algorithm, called AdaBoost.M1. In this algorithm, as new training sets arrive, the classifiers learn incrementally and the weights of the classifiers in the voting combination are updated. Various incremental learning algorithms have been introduced with the support vector machine (SVM) to sequentially learn knowledge [7] [15].

2.5 Combinations of Classifiers

Classifier combinations have also been developed and used, motivated by the idea that combining the opinions of more than one expert can lead to more confident decision making than the opinion of only one expert. The Combining Classifiers approach in data mining merges classifier outputs, or opinions, to improve the accuracy of the final decision. Different classifier combination methodologies have been proposed in previous studies [14]. Bagging [40] and boosting [63] are two examples where combined classifiers have been successful.

In the bagging method, a set of unstable base classifiers are trained on randomly selected instances from the training set. An unstable classifier is a classifier where a small change in the training set causes a big change in the classifier's output. These classifiers create a large diversity in results, and as a result, they improve the accuracy [25]. In the bagging method, the base classifiers' outputs are combined by majority voting. It means that the most frequently predicted output is selected as the final classified output. In other words, if C_1, $C_2...C_n$ are a set of base classifiers trained on randomly selected instances of a training set, and the majority of these classifiers classify instance x to class label y, the output of the bagging combination of these classifiers for instance x will be y.

In the boosting method, weak classifiers are trained consecutively on a weighted

training data set. The training elements' weights are updated after training each weak classifier, based on the accuracy of the current classifier on the training examples. The adaBoost learning algorithm is a popular boosting algorithm. The following section will explain this algorithm in more detail.

2.5.1 AdaBoost algorithm

The AdaBoost (Adaptive Boosting) algorithm was proposed by Freud and Schapire in 1997 [79]. The Adaboost.M1 algorithm is shown in Figure [4]. This algorithm takes the training set $\{(x_1,y_1),(x_2,y_2),...(x_m,y_m)\}$ as its input. x_i is the feature vector from the feature space and y_i is the associated class set for vector x_i. T base (weak) classifiers are built iteratively by calling a weak learning algorithm, which is an algorithm that generates a learner(classifier) that performs slightly better than a random guess. A classifier with performance of about 50% is called a weak learner/classifier. In each iteration step except the first one, training instances are chosen that are based on distribution weights assigned to instances. In the first iteration, the same weight is given to all training instances. A higher weight is given to a training instance which was misclassified in the previous iteration step. The error (ϵ_t) is calculated for each weak hypothesis (h_t) by the following formula as D_t is the current distribution on the training set in iteration t.

$$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$$

Given: $(x_1,y_1),...,(x_m,y_m)$
where $x_i \epsilon$ X, $y_i \epsilon$ Y = {-1,+1}
Initialize: D_1=1 / m
For t=0,...,T:

- Train weak learner using distribution D_t.
- Get weak hypothesis h_t: X $\longrightarrow$ {-1,+1}
- with error $\epsilon_t = Pr_{i\sim D_t}\ [h_t(x_i) \neq y_i]$
- Choose $\alpha_t = \frac{1}{2}\ln(\frac{1-e_t}{e_t})$
- Update: D_{t+1}(i)=$\frac{D(i)}{Z_t} * \begin{cases} e^{-\alpha_t} if h_t(x_i)=y_i \\ e^{\alpha_t} if h_t(x_i)\neq y_i \end{cases}$ where Z_t is a normalization factor.

Output the final hypothesis:

$$\text{H(x)=sign}(\sum_{t=1}^{T} \alpha_t h_t(x))$$

Figure 4: AdaBoost.M1 Algorithm Adapted from Qahwaji et al.(2008)[61]

2.6 K-Means Clustering

The K-Means clustering algorithm is a popular unsupervised learning algorithm. This algorithm groups the given data into K clusters which are represented by their centroids. In the first step, K centroids are initialized. Then, each data instance in the dataset is assigned to its nearest centroid. In the next step, the centroids are updated by taking the mean of data points in each of the current clusters and again all the data examples in the given data set are assigned to the closest of the resulting new centroids. This assignment and update process is repeated until the centroids remain the same after each cycle.

The Euclidean function is a common distance function that is used for calculating the distance between data points and centroids in the K-means algorithm. The accuracy of the K-mean algorithm is impacted by the assignments of the initial centroids. The original K-mean algorithm randomly initialized centroids. This arbitrary selection causes different results (different centroids) every time the algorithm operates on the same data set. Therefore, in the literature, several enhanced methods have been proposed to overcome this weakness. Abdul Nazeer and Sebastian [2] proposed a refined method in which K data-point clusters, with sizes less than a specific threshold, are created from the given dataset. The initial centroids are the mean vectors of the clusters. Vijayakumar et. al [51] proposed a method for selecting the initial centroids where the data-points are sorted based on their distances from the origin. Then the sorted set is divided into k sets. The initial centroids are the middle points of these k sets.

Chapter 3

Learn++ Incremental Algorithm and its Modified Version

3.1 Incremental Classification Learning

For many real-world classification problems, data collection is an expensive and time-consuming process. Therefore, data is collected through batches over a period of time. Earlier batch learning algorithms, such as the one for MLP neural network, have high computing and timing cost because every time a new batch of data becomes available, new classifiers have to be built and trained on the combination of the old and new data, so previously learned knowledge is lost and a phenomenon called catastrophic forgetting [47] happens. A feasible solution is offered by incremental learning algorithms. This type of algorithm builds classifiers which incrementally learn new information from new data without forgetting previously gained knowledge.

Recently, the attention of researchers in this field has shifted toward incremental

learning algorithms and several algorithms of this type have been proposed. Polikar et.al [59] proposed Learn++ which performs well on tested data sets. This algorithm learns incrementally by building an ensemble of weak classifiers, each trained on a subset of the existing training data set. These classifiers are combined through weighted majority voting in the testing phase. In the following section, we describe original Learn++ in more details and then in section 3.1.2, a later version of Learn++ called dynamically weighted majority voting (DWMV) Learn++ [58] is presented. Finally, in the last section 3.1.3, a modified version of DWMV Learn++ is proposed. This algorithm is the main contribution of this thesis. The modified DWMV Learn++ has a major advantage over the original DWMV Learn++ in that it requires pre-assumption on the distribution of the training dataset (original Learn++ assumes that it is Gaussian).

3.1.1 Original Learn++

Learn++ (Figure 5) is an incremental learning algorithm which was mainly inspired by the AdaBoost algorithm [79]. As with AdaBoost, Learn++, iteratively creates a number of (weak) classifiers from the available dataset and combines them through weighted majority voting [53]. In both algorithms, the training samples for each classifier are chosen from the updated distribution. In AdaBoost, distribution update rule optimization is done to improve classifier accuracy, while in Learn++ this is done to improve the incremental learning of new data.

As shown in Figure 6, Learn++ requires D_k (the available datasets) and T_k (the number of classifiers iteratively generated for each dataset) as inputs. In the training

step, elements of the current dataset, D_t, are first equally weighted as 1/m, where m is the number of elements in D_t. After normalizing the element weights, training and testing instances for building the base classifier are randomly selected based on the weight distribution. When the base learner is trained, the accuracy error of the classifier is computed. If the error is less than 1/2, the algorithm considers the classifier to be suitable and keeps it. Otherwise, the classifier is discarded. In the next stage, all existing classifiers, trained through a subset of the current training set D_t, are combined through weighted majority voting. As long as the composite error is less than 1/2, the current classifier is kept. Otherwise, the classifier is discarded, a new training subset is selected, and a new classifier is generated. The normalized composite error is used for updating the instances' distribution. Misclassified instances in the current hypotheses are assigned higher weights in order to have a higher probability of being selected in the subsequent training set. The final hypothesis decision is made by integrating the hypothesis using weighted majority voting. Note that weighted majority voting gives higher weight to classifiers with higher accuracy in training and testing subsets. As mentioned, Learn++ uses a weak learning algorithm to build weak classifiers [11].

3.1.2 Dynamically Weighted Majority Voting (DWMV) Incremental Learning

DWMV Incremental Learning (Figure 7) is a modified version of Learn++. As we have seen, Learn++ assigns voting weights to base classifiers based on the performance of each classifier on its own training set. The weights in Learn++ are set during training and stay fixed from this point. However, the DWMV Incremental

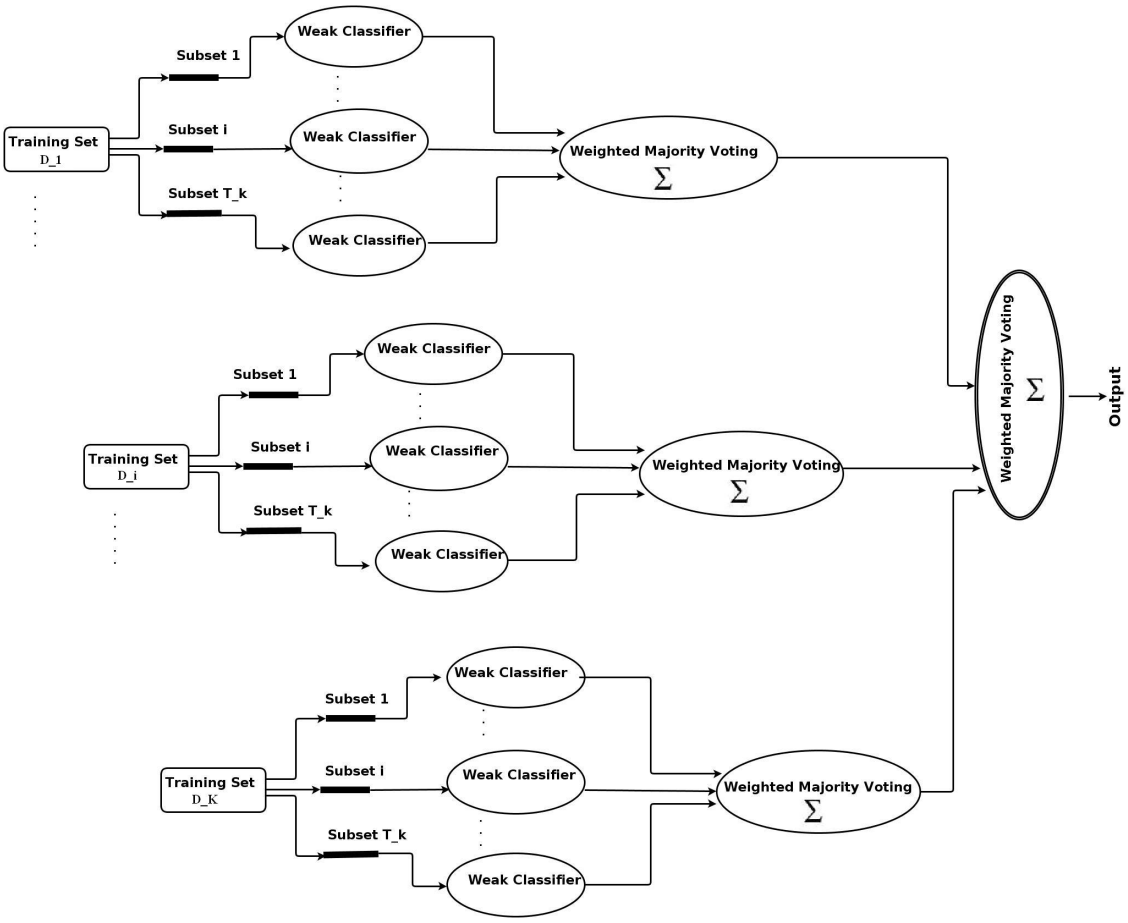


Figure 5: Original Learn++

Learning algorithm dynamically weights the base classifiers. This means that the algorithm updates the classifiers' weights based on the location of the test input. The idea behind this algorithm is that the classifier whose training data is closest to the unknown instance is most likely to be correct, and should therefore get the higher weight. The distance between an unknown instance and a training set is calculated using Mahalanobis distance [20]. This distance measurement between a multivariate vector x and a set of vectors with mean m and covariance matrix C is calculated as:

$$D^2 = (x - m)^T * C^{-1} * (x - m)$$

Input: For each database drawn from D_k *k=1,2,...,K*

- Sequence of m training examples S=[(x_1,y_1),(x_2,y_2),...,(x_m,y_m)].
- Weak learning algorithm WeakLearn.
- Integer T_k, specifying the number of iterations.

Do for*k=1,2,...,K*:
Initialize w_1(i)=D(i)=1/m,$\forall i$,unless there is prior knowledge to select otherwise.
Do for*t=1,2,...,T_k:*

1. Set D_t=$w_t/\sum_{i=1}^{m} w_t(i)$so that D_t is a distribution.
2. Randomly choose training TR_t and TE_t subsets according to D_t.
3. Call WeakLearn, providing it with TR_t.
4. Get back a hypothesis h_t: X→Y, and calculate the error of h_t:$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$ on $S_t = TR_t + TE_t$. If ϵ_t >1/2, set t = t-1, discard h_t and go to step 2. Otherwise, compute normalized error as $\beta_t = \epsilon_t/(1-\epsilon_t)$.
5. Call weighted majority, obtain the composite hypothesis

   H_t = arg $max_{y\epsilon Y} \sum_{t:h_t(x)=y} \lg(1/\beta_t)$, and compute the

   composite error $E_t = \sum_{i:H_t(x_i)\neq y_i} D_t(i)\,[|H_t(x_t) \neq y_i|]$

   If E_t>1/2, set t = t-1, discard h_t and go to step 2.

6. Set $B_t = E_t/(1-E_t)$(normalized composite error), and update the weights of the instances:

   $$w_{t+1}(i) = w_t(\mathrm{i}) \times B_t, \text{ if } H_t(x_i) = y_i$$

   Call weighted majority on combined hypotheses and **output** the final hypothesis:

   $$H_{final} = \arg\, max_{y\epsilon Y} \sum_{k=1}^{K} \sum_{t:H_t(x)=y} \lg(1/B_t)$$

Figure 6: Original Learn++ Algorithm Adapted from Polikar et al. (2001)[59]

Where D is Mahalanobis distance, T indicates that the vector should be transposed (switching the rows and column of the vector) and C^{-1} is the inverse of C. The covariance matrix is defined as a matrix whose element at the position i,j describes the covariance of the i^{th} and j^{th} random vectors. The Mahalanobis distance function requires the means and covariance matrices of the training set for calculating the distance. DWMV Learn++ keeps the mean and covariance matrix of the training sets and ignores the training data. The weights of classifiers in the majority voting combination are calculated as follows:

Assume TR_{tc} is the subset of the training dataset in the t^{th} iteration. TR_{tc} includes instances which are categorized to class c. Class-specific Mahalanobis distance is computed from the Mahalanobis distance formula as:

$$M_{tc}(x) = (x - m_{tc})^T * C_{tc}^{-1} * (x - m_{tc})$$

Where m_{tc} is the mean and C_{tc} is the covariance matrix of TR_{tc}. Then, the dynamic weight of the t^{th} classifier in this algorithm is calculated as:

$$DW_t(\text{x}) = \frac{1}{\min(M_{tc}(x))} \text{ c=1,...,C;t=1,...,M}$$

Where M is the total number of generated classifiers. [58] shows that the generalization performance of the modified Learn++ is better than the original Learn++ and the AdaBoost algorithms. Also, DWMV Learn++ has more tolerance to stability-plasticity than the other two algorithms, as it retains more of its previously gained knowledge than other two algorithms.

The Mahalanobis distance function implicitly assumes that the data distribution is

Gaussian, which is not applicable to most real-world applications. This shortcoming of DWMV Learn++ is eliminated in the following proposed modified DWMV Learn++ by using clustering techniques in order to find the distance between training sets and unknown instances.

3.1.3 Modified DWMV Learn++

Modified DWMV Learn++ is the main contribution of this thesis. It, similar to DWMV Learn++ and original Learn++, combines weak learners (classifiers) in order to classify unknown instances. All three algorithms iteratively update the training instances' distribution weights based on the performance of the composite classifiers.

The modified and original versions of DWMV Learn++ both dynamically designate the voting weight to each base classifier using the distance between a classifier's training set and the unknown instance. There are a few differences between these two algorithms: 1) The distance between a training set and a testing (unknown) instance is calculated differently, and, 2) The weights of classifiers for weighted majority voting are also calculated differently. The differences are explained in more detail as follows. As mentioned in the previous section, DWMV Learn++ uses a *class-specific Mahalanobis distance function* to determine the distance between the training set and an unknown instance. However, the modified DWMV Learn++ clusters the training instances with the same class label into a predefined number of clusters using the K-means clustering algorithm. The minimum Euclidean distance of the unknown instance to the centroids of the clusters is determined as a *class-specific distance*. Suppose TR_{tc}, a subset of a training set TR_t, contains the instances which are categorized

Input: For each database drawn from D_k *k=1,2,...,K*

- Sequence of m training examples S=$\{(x_i,y_i)|$ i=1,...,$m_k\}$.
- Weak learning algorithm BaseClassifier.
- Integer T_k, specifying the number of iterations.

Do for *k=1,2,...,K*:
Initialize w_1(i)=D_1(i)=$1/m_k$, $\forall$ i = 1,2,...,T_k

If k >1, Go to Step 5, evaluate current ensemble on new data set D_k, update weight distribution.
End If
Do for *t=1,2,...,T_k:*

1. Set D_t=$w_t/\sum_{i=1}^{m} w_t(i)$so that D_t is a distribution.
2. Drawing training TR_t and TE_t subsets from D_t.
3. Call BaseClassifier to be trained with TR_t.
4. Obtain a hypothesis h_t: X→Y, and calculate its error of h_t:
   $\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$ on $S_t = TR_t + TE_t$.
   If ϵ_t >1/2, set t = t-1, discard h_t and go to step 2.
5. Call dynamically weighted majority voting (DWMV) to obtain the composite hypothesis

   H_t = arg $max_{y\epsilon Y}$ $\sum_{t:h_t(x)=y} DW_t$(x),
6. compute the error of the composite hypothesis
   $E_t = \sum_{i:H_t(x_i)\neq y_i} D_t(i) = \sum_{i=1}^{m_k} D_t(i)$ $[|H_t(x_t) \neq y_i|]$

   If E_t>1/2, discard H_t and go to step 2.
7. Set $B_t = E_t/(1\text{-}E_t)$, and update the weights of the instances:
   $w_{t+1}(i) = w_t$(i) × B_t, if $H_t(x_i) = y_i$

   Call DWMV and **output** the final hypothesis:

$$H_{final}(\text{x}) = \text{arg } max_{y\epsilon Y} \sum_{k=1}^{K} \sum_{t:h_t(x)=y} DW_t(x)$$

Figure 7: DWMV Learn++ Algorithm Adapted from Polikar et al. (2005)[58]

as class c, that is,

$$TR_{tc}=\{x_i \mid x_i \epsilon\ TR_t\ \&\ y_i =\text{c}\ \}$$

and TR_{tc} is clustered into m distinctive clusters with c_1,c_t,...c_m centroids, then, *class-specific distance* is defined as:

$$M_{tc}(\text{x})=argmin_{j=1,...,m}\text{ED}(c_j,\text{x})$$

where ED is Euclidean distance function. And the distance based on the dynamic weight of the t^{th} classifier which is trained on TR_t during the t^{th} iteration is computed as:

$$DW_t(\text{x})=\frac{1}{e_t}\ *\ \frac{1}{\min_c(M_{tc}(x))}\ \text{c=1,...,C;t=1,...,M}$$

where C is the total number of class labels, M is the total number of classifiers and e_t is the error of the classifier on TR_t. In the final step, for the final output calculation, the dynamic weights of the composed classifiers are calculated as:

$$\acute{DW_t}(x)=\frac{1}{E_t}\ *\ \frac{1}{\min_c(M'_{tc}(x))}\ \text{c=1,...,C;t=1,...,K}$$

where K is the number of iterations, E_t is the error in the composed classifiers, H_t, in the training set of each iteration and M'_{tc} is the *class specific distance* on D_t where t=1,...,K.

$$H_{final}(\text{x}) = \arg\ max_{y\epsilon Y}\ \sum_{t=1}^{K}\ \acute{DW_t}(x)$$

Unlike the original DWMV Learn++, the proposed version is not based on any assumptions about the training dataset's distribution. Therefore, this algorithm has an advantage over the original version which assumes that the training set has a Gaussian distribution. Thus, this algorithm can performs better on real-world applications than the original version.

Chapter 4

Data Gathering and Preparation

4.1 Data Sources

In this research, we gathered training and test data from two databases : the Canadian Heart Health Database [48] and the Stonechurch Health Clinic's EMR database. In this section, we will describe the data sources and explain the process of preprocessing data for this research.

4.1.1 Training Dataset - Canadian Heart Health Database

The training data comes from the Canadian Heart Health Database. The Database contains 23,129 individuals' records. This information was collected through a two-stage survey in the ten Canadian provinces. The survey was executed between 1986 and 1992. The participants were aged between 18 and 74. In the first stage of the survey, demographic data and knowledge of cardiovascular risk factors, attitudes and knowledge of people about heart disease related issues were collected. Also, two

clinical blood pressure readings were taken during this step. In the second step of the survey, anthropometric measurements and two additional blood pressure readings were recorded.

4.1.2 Test Dataset - Stonechurch Database

The test data were gathered from the Stonechurch Clinic's EMR [1] database. Permission to access this database was obtained from the Hamilton Health Science/ McMaster University Faculty of Health Sciences Research Ethic Board. Privacy of data was carefully secured at all times. For those records that were selected, patient identifiers were removed and the remaining data were stored in encrypted form on the computer that was used for algorithm development and testing.

The medical information of two groups of patients were collected for testing the data mining models as follows: 1) Patients who were diagnosed with heart disease in the past 5 years and 2) Patients who had not developed heart disease over the past 5 years. 140 patients had developed heart diseases such as Heart Failure, Hypertensive Heart Disease and Atrial Fibrillation. Also, the records of 30 patients for whom no report of heart disease had been recorded at the time of data retrieval were collected for use as a test dataset. The following section describes the data preprocessing procedure which was used.

[1] Hamilton, Ontario

4.2 Preprocessing Data

From both the Canadian Heart Health (CHH) and the Stonechurch databases, we selected records with recorded attribute values which had the required heart disease risk factors. This information, as reported in Table 1, includes sex, age, systolic blood pressure (SBP), diastolic blood pressure (DBP), total cholesterol (TC), high-density lipoprotein (HDL), body mass index (BMI), diabetic status, smoking status and physical activity. For the Canadian Heart Health training set, we eliminated the records of patients younger than 30 or records with at least one missing attribute value. The final set included 11,556 patient records. As mentioned in the previous section, there were four (systolic/diastolic) blood pressure readings in this database, so, in order to have a more precise blood pressure, we used the average of these four blood pressure readings for the required attribute values. In order to get correct results from the classification models enough instances of males and females with and without diabetes were included in the training set.

For the Stonechurch clinic test set, patient records with one or missing attribute values were deleted. The final test set included 28 patients who developed heart disease over the previous 5 years and 30 patients who did not develop heart disease over the 5 year period. To use the data for supervised classification, the intent was to categorize the test records into two heart disease risk levels, class A and class B with low and high risk of developing heart disease respectively.

Table 1: Training and Test DataSet Attributes

	CHH DataSet		**Stonechurch DataSet**	
DataSet Attributes	Minimum	Maximum	Minimum	Maximum
Sex	0 (female)	1 (male)	0 (female)	1 (male)
Age	30	74	32	73
SBP	95	208	100	172
DBP	54	110	58	110
TC/HDL	2	11	2	7
BMI	19	43	18	48
Diabetic Status	0 (non-diabetic)	1 (diabetic)	0 (non-diabetic)	1 (diabetic)
Smoking Status	0 (non-smoker)	1 (smoker)	0 (non-smoker)	1 (smoker)
Physical Activity	0 (non-active)	1 (active)	0 (non-active)	1 (active)

4.2.1 Labelling Data

Since supervised classifiers were used throughout this research all training data were labelled. For this reason, patient records in the training set who had a history of heart disease were classified as high risk (level B). 128 cases were randomly selected from this group as level B class-label instances for the training set. We used the Framingham equation for labelling the records with low risk as level A using the 1991 version of Framingham equation to calculate the heart disease risk for patients from the training database who had not developed heart disease. 139 patients were selected from the training database with a low risk ($\leq 10\%$) of heart disease.

Chapter 5

Experimental Results

5.1 Evaluation Metrics

There are several performance measurement criteria proposed for comparison of classifiers, such as accuracy [24], error rate (ER) [24], sensitivity [5], specificity [5], precision [10], recall [10] and F-score [10]. In this study, we used accuracy, sensitivity and specificity criteria to compare the performance of different classifiers. Additionally, in order to identify classifier parameters, such as the optimal number of hidden neurons for neural network models or K parameter values for k nearest neighbor classifiers, mean squared error was used as an indicator for comparing neural networks and KNN classifiers with different parameter values.

Accuracy, sensitivity and specificity are defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = \frac{TN}{FP + TN} \tag{3}$$

- **True Positives (TP)** are individuals who are predicted as high risk among patients diagnosed with heart disease.
- **True Negatives (TN)** are individuals who are predicted to be low risk among the patients who are not diagnosed with heart disease.
- **False Positives (FP)** are individuals who are predicted to be high risk among people who are not actually diagnosed with heart disease.
- **False Negatives (FN)** are individuals who are predicted to be low risk among patients who have actually developed heart disease.

A confusion matrix [57] is a tabular representation of TP, TN, FP and FN, to provide a way to visualize information about classifier performance. The instances in a predicted class and in an actual class are shown in columns and rows of the matrix. Table 2 shows the general confusion matrix for the classifiers in this thesis. According to the given definitions for TP, FP, FN and TN, sensitivity in our classification problem is the ability of a classifier to correctly identify patients who are at a high risk of developing heart disease and specificity is the ability of a classifier to

Table 2: General Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	TP	FN
Low Risk	FP	TN

correctly identify patients who are at a low risk of developing heart disease.

Batch classifiers in this research were evaluated based on a cross-validation techniques. This is a technique for evaluating the performance of a classifier using the performance of models generated through a separate partition of the training set. This method prevents overfitting [66] of a classification model (high accuracy on training data and low accuracy on test data). K-fold cross-validation is a common type of cross-validation. In K-fold cross validation, the training set is randomly split into K partitions called folds. Then, K-1 folds are used for training the model and the remaining fold is retained as the evaluation set during the testing step. This process is repeated K times until all the partitions are used for validation of the model. The final performance result is calculated as the average of the results in the K iterations.

5.2 Comparison of Batch Classifier Performance

The following sections (sections 5.2.1-6) present the results of running individual batch classifiers on the test set in terms of the confusion matrix. In section 5.2.7, batch classifiers are compared according to sensitivity, specificity and accuracy measurement criteria.

5.2.1 C4.5 Decision Tree Classifier

The Java implementation of the C4.5 decision tree classifier is called J48 in WEKA. Default parameters in WEKA for J48 were used in this evaluation. The confidence factor was 0.25 and the minimum number of instances per leaf was 2. The confusion

matrix of this classifier is shown in Table 3.

Table 3: J48 Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	16	12
Low Risk	5	25

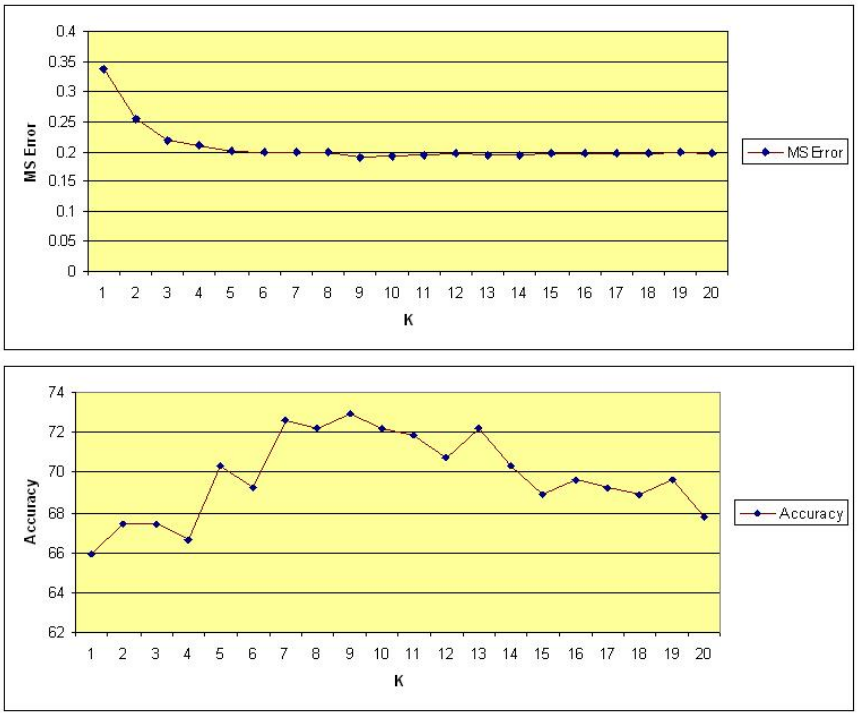
5.2.2 K Nearest Neighbour (KNN) Classifier

In this classifier, the Euclidean function is used as a distance measurement for calculating the distance between a new instance and the training instances. In order to choose an optimal K parameter value for this classifier, the accuracy and the mean square error of KNN for different K values (the number of neighbors) in the training set were calculated. The 100-fold cross-validation technique was used for this calculation. Figure 8 presents this effect. According to Figure 8, k=9 is the best choice for the parameter k as accuracy is maximum and mean squared error (MSE) is minimum at this point. The accuracy of the KNN hypothesis was tested with the test set. The result is presented in terms of the confusion matrix in Table 4.

Table 4: KNN Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	16	12
Low Risk	11	19

Figure 8: Effect of Parameter K on Classifier Accuracy



5.2.3 Naïve Bayes Classifier

This is a very simple algorithm which is based on the assumption that numeric attributes are conditionally independent. There are no parameters to set. The following table presents the confusion matrix of this classifier on the test dataset.

Table 5: Naïve Bayes Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	18	10
Low Risk	13	17

5.2.4 Bayesian Neural Network (BNN) Classifier

In this research, the K2 algorithm [19] was used as a search and score method for learning the BNN structure. The confusion matrix for this classifier is reported in Table 6.

Table 6: Bayesian Neural Network Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	19	9
Low Risk	9	21

5.2.5 Multilayer Perceptron Neural Network Classifier

We considered using only one layer as a hidden layer in this network since almost all applications perform well using single-hidden layer MLPNN classifiers. The number of input and output neurons in input and output layers were set to 9 and 2 with respect to the number of features and the number of risk levels (class labels). In order to choose the optimal number of hidden neurons, we checked the accuracy and MSE of this classifier for different numbers of hidden neurons for the training set. Figure 9 shows the result of this test. η (learning rate) and α (momentum) were set at 0.3 and 0.2 respectively. We used 1000 for the number of epochs. The best number of hidden neurons was 3, as Figure 9 shows, since the accuracy of the classifier was at its maximum and the MSE has the second smallest value.

After training the MLP classifier with the above topology, the performance of the

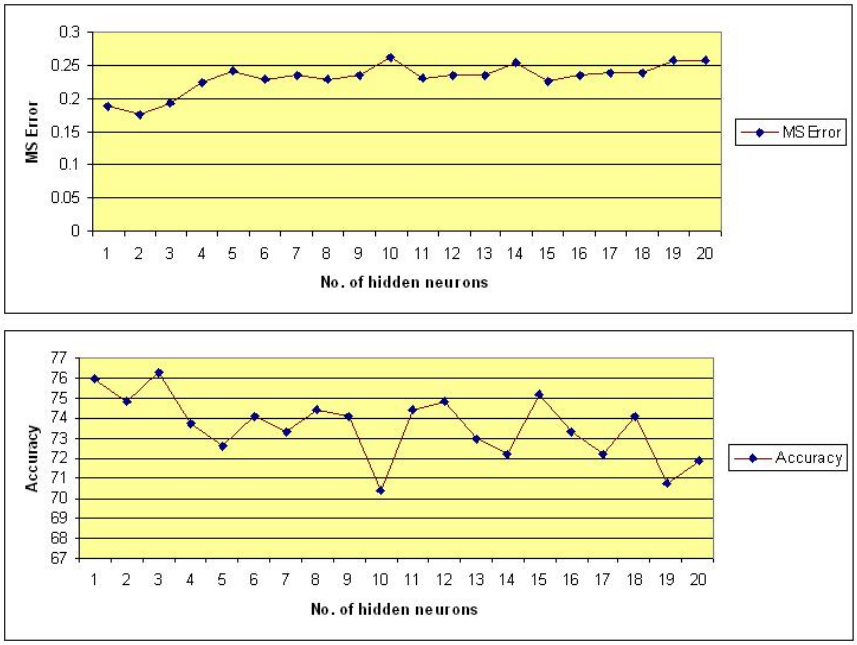


Figure 9: Variation of RMS Error and Accuracy When Increasing the Number of Hidden Neurons in MLP

model was evaluated on the training set. Table 7 presents its performance in confusion matrix terms.

Table 7: Multilayer Neural Network Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	17	11
Low Risk	9	21

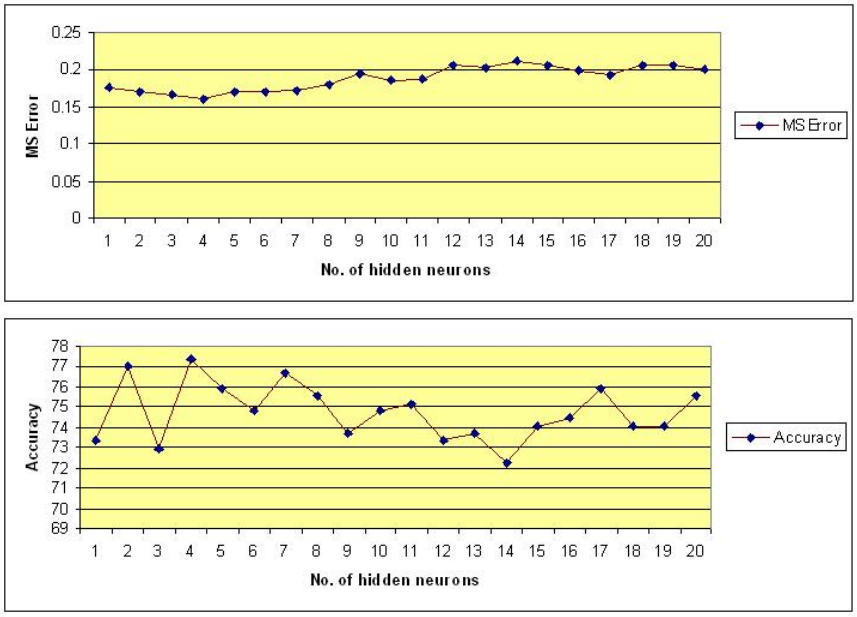


Figure 10: Variation of RMS Error and Accuracy When Increasing the Number of Hidden Neurons in RBF Neural Network

5.2.6 Radial Basis Function (RBF) Neural Network Classifier

The RBF neural network classifier in WEKA employs the K-means clustering algorithm for determining the centres of radial basis functions. The logistic regression algorithm was applied to determine the weights in this network. We tested the accuracy of the classifier for different numbers of hidden neurons. As Figure 10 shows, the classifier has the highest accuracy and the lowest MSE when the number of hidden neurons is 4.

Table 8 presents the performance of the RBF neural network on the test set.

Table 8: RBF Neural Network Confusion Matrix

Actual Risk Level	Predicted Risk Level	
	High Risk	Low Risk
High Risk	16	12
Low Risk	10	20

5.2.7 Comparison of Results

A summary of batch classifier performance in terms of accuracy, sensitivity and specificity is shown in Table 9. According to the analysis, the decision tree (J48) had the highest accuracy and specificity of all the classifiers, while the Bayesian Neural Network had the highest sensitivity. Although in most applications, accuracy is used to evaluate model performance, in medical applications sensitivity and specificity are more important.

Higher sensitivity in Table 9 shows the higher ability of the corresponding classifier to recognize patients with a high risk of developing heart disease in five years. When physicians need to determine which patients are at high risk of developing heart disease in order to send them for a further checkup, sensitivity is the best predictor for evaluating classifier performance. Therefore, in our result, the optimal batch classifier for predicting high risk of heart disease is the Bayesian Neural Network (BNN).

Higher specificity in Table 9 describes a higher ability of the classifier to identify patients with low risk of developing heart disease in five years. In cases where identifying low risk patients is valuable, classifiers with higher specificity are useful. As our result shows, the Decision Tree (J48) classifier provides the best result in this case.

Table 9: Sensitivity, Specificity and Accuracy

	Sensitivity	Specificity	Accuracy
Decision Tree (J48)	57.1%	83.3%	70.7%
K Nearest Neighbor	57.1%	63.3%	60.3%
Naïve Bayes	64.3%	56.7%	60.3%
Bayesian Neural Network	67.9%	70.0%	69.0%
Multilayer Preceptron Neural Network	60.7%	70.0%	65.5%
Radial Basis Function Neural Network	57.1%	66.7%	62.1%

5.3 Comparison of Incremental Classifier Performance

In order to evaluate the ability of the incremental classification algorithms presented in this study, we split the training set into three batches. The data distribution is given in Table 10. We used the MLP classifier as the base classifier for the original Learn++, original DWMV Learn++ and modified DWMV Learn++. The base classifier (weak learner) was a single hidden layer MLP neural network with 40 hidden and two output nodes with an MSE goal of 0.3. In each training session, a maximum of 10 weak hypotheses (T_k=10) were generated in each training step. The modified DWMV Learn++ had two additional parameters for the number of clusters for each class-specific instances of TR_t and S_t. After experimenting with different numbers of clusters, the best optimal values were obtained for c_t=8 and 9 clusters for TR_t and S_t, respectively.

Tables 11, 12 and 13 illustrate the generalization performance of original Learn++, original DWMV Learn++ and modified DWMV Learn++ on the datasets (S_1,S_2,S_3,S_{test}).

Table 10: Data Distribution Of Training and Testing Sets

DataSet	*LowRisk*	*HighRisk*
S1	47	40
S2	44	49
S3	48	39
TEST	**30**	**28**

The result is shown in the compact format given in [59] instead of illustrating the result in confusion matrix. The numbers shown in Table 11-13 were calculated as the mean of the generalization performance of the algorithms after 10 runs. According to the results shown in the last rows of the tables, modified DWMV Learn++ improved on the performance of the original Learn++ algorithm by about 5%, and outperformed the original DWMV Learn++ by about 12%. The decrease in training performance shown in Tables 11-13 over all training sessions is explained by the stability-plasticity dilemma that incremental learners deal with. Algorithms ran on the same computer, and the running times of the original Learn++, the original DWMV Learn++, and the modified DWMV Learn++ on average after 10 runs were 25.6, 26.3 and 26.5 seconds respectively. As these tests have shown, the running times of these incremental learning algorithms on our dataset were not significantly different.

Table 11: Original Learn++ Performance

DataSet	$TS1$	$TS2$	$TS3$
S1	87.2%	74.2%	71.2%
S2	-	94.2%	80.3%
S3	-	-	75.3%
TEST	**55.2%**	**57.9%**	**61.6%**

Table 12: Original DWMV Learn++ Performance

$DataSet$	$TS1$	$TS2$	$TS3$
S1	69.2%	59.4%	60.9%
S2	-	73.5%	66.3%
S3	-	-	64.8%
TEST	**51.9%**	**53.9%**	**54.6%**

Table 13: Modified DWMV Learn++ Performance

$DataSet$	$TS1$	$TS2$	$TS3$
S1	90.9%	80.5%	80.3%
S2	-	82.9%	84.9%
S3	-	-	84.7%
TEST	**56.9%**	**61.0%**	**66.4%**

In order to compare the ability of the incremental learning algorithms to learn new classes and to retain the information that was previously learned, another test was run on the training sets with the distribution given in Table 14. As shown, the high level class-label instances were included only in the second and third batch of the training set. On the first batch of data, we have instances from the low level category. Performance from the original Learn++, the original DWMV Learn++ and modified DWMV Learn++ algorithms are shown in Tables 15, 16 and 17 respectively. The entries in each row represent the classification performance per class of the testing set in each training step. The last column shows the overall classification performance of the algorithms on the testing set. As the results show, the modified DWMV Learn++ performs better than the original Learn++ for learning new class instances (42.0% compared to 50.0%). However, the original DWMV Learn++ performs better than the modified DWMV Learn++ (78.6% compared to 50.0%). In this case, attributes

in our training dataset such as DPB, SBP and BMI have Gaussian distributions for high risk instances and the Mahalanobis distance has an implicit assumption of this distribution. The original DWMV Learn++ outperforms the modified version which calculates the distance based on the k-mean. However, for instances in our training dataset the attribute values for low risk instances do not have Gaussian distribution, therefore, Mahalanobis distance measurement is not precise and as a result, the original DWMV Learn++ performs worse than the other two algorithms.

Table 14: Data Distribution Of Training and Testing Sets

$DataSet$	$LowRisk$	$HighRisk$
S1	47	0
S2	44	89
S3	48	39
TEST	**30**	**28**

Table 15: Original Learn++ Performance

$TrainingDataSet$	$LowRisk$	$HighRisk$	$OverallPerformance$
S1	100.0%	-	51.7%
S1,S2	60.0%	42.0%	54.6%
S1,S2,S3	63.3%	46.4%	56.5%

Table 16: Original DWMV Learn++ Performance

$TrainingDataSet$	$LowRisk$	$HighRisk$	$OverallPerformance$
S1	100.0%	-	51.7%
S1,S2	26.7%	78.6%	51.7%
S1,S2,S3	20.0%	85.7%	51.7%

Table 17: Modified DWMV Learn++ Performance

$TrainingDataSet$	$LowRisk$	$HighRisk$	$OverallPerformance$
S1	100.0%	-	51.7%
S1,S2	63.3%	50.0%	57.9%
S1,S2,S3	63.3%	64.3%	64.7%

5.4 Comparison of Batch and Incremental Classifiers

According to the results from this study, the best incremental learning algorithm (the modified DWMV Learn++) in terms of accuracy in this research performs slightly better than the best batch learning algorithm for multilayer perceptron (MLP) neural network(65.5% compared to 66.4%). In addition, the modified DWMV Learn++ has the advantage that it is able to incrementally learn new information as it becomes available. However, the MLP batch learning algorithm rebuilds a new classifier from both old and new data which makes the overall time and space complexity of the heart disease monitoring system higher.

In comparison, the best batch classifier (J48) in terms of accuracy in this work performs better than the best incremental classifier (modified DWMV Learn++) (70.7% compared to 66.4%). However, the incremental classification system is still more valuable for our application, as data most likely becomes available through small batches over a period of time and thereby learns continuously from the new incoming data. In addition, there might be situations where access to the previous data is

impossible as a result of data corruption or lost data.

Table 18: Batch MLP Classifiers and Modified DWMV Learn++ Performance

	$BatchMLP$	$ModifiedDWMVLearn++$
Accuracy	65.5%	66.4%

Table 19: J48 (Decision Tree) Classifiers and Modified DWMV Learn++ Performance

	$J48(DecisionTree)$	$ModifiedDWMVLearn++$
Accuracy	70.7%	66.4%

5.5 Heart Disease Monitoring Estimator Decision Support System

The availability of a good incremental classifier for classifying patient risk for developing heart disease creates an opportunity for future development that implements the results of this research to work in the form of a decision support system. Figure 11 is an overall conceptual design of the proposed system.

The learning component of this system would be the modified DWMV Learn++ classifier. This system would be trained on an initial training set, and it would improve its performance over time as new training data become available. This system would assist healthcare providers in screening patients at potential risk of heart disease.

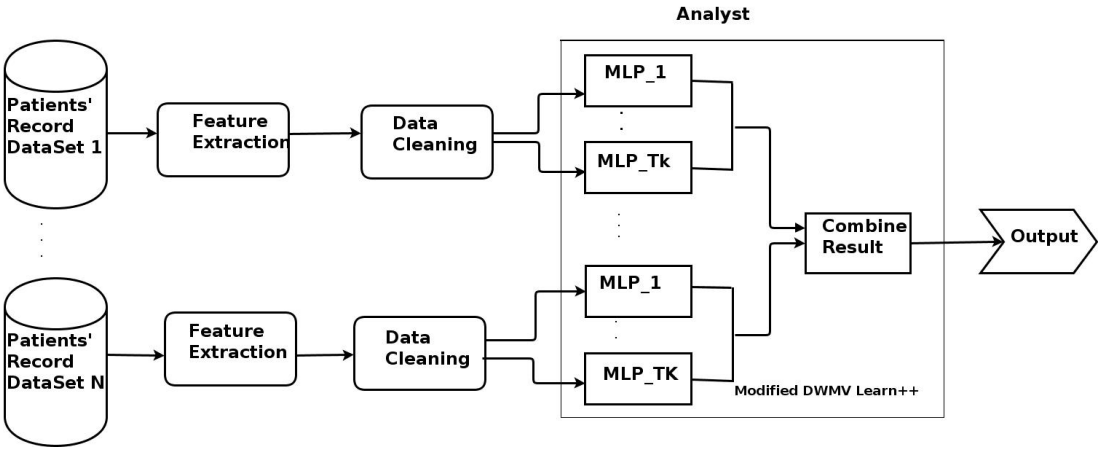


Figure 11: Heart Disease Monitoring CDSS

5.6 Future Work

There are a variety of research directions that can be further developed using part of this thesis. One of them is to improve the performance of the proposed system. Another is the domain it is applied in. In order to make the system more accurate and more compatible with new and existing heart disease guidelines, one can add other heart disease risk factors as predictive attributes, such as family history/ alcohol consumption/etc. Adding more related risk factors to the predictive attributes provide more information to the algorithm, thus improving the overall predictive accuracy of the system. An additional improvement in accuracy would result from applying physicians' opinions in labeling the training data records in the decision support system, might also result in improving the accuracy of the decision support system. Access to more patient medical records for validating the classifiers would also be useful in order to find the best optimal classification model for the learning component of the heart disease screening decision support system. More research could be done by comparing the accuracy of other incremental learning algorithms

such as Learn++.MI with the proposed modified DWMV Learn++.

This system could also be applied to domains other than health care. For example, it could be applied in the finance field where credit risk assessment and credit evaluation are a concern. The system could then be customized to predict the chances of prompt payment for new customers, based on the financial history of previous clients or other factors deemed relevant.

Appendix

The Framingham method is a well-known risk prediction method for estimating the risk of cardiovascular disease (CVD) over the course of 5 or 10 years. This is driven by the data collected from the Framingham Heart Study. In this work, the 1991 version of the prediction model is used for calculating the risk of developing heart disease within 5 years for finding the patients record with low risk from the training database. This method can be only employed on the sample population age between 30 and 74 without any cardiovascular heart disease. This prediction approach used the following risk factors: age, sex, systolic blood pressure (SBP), diastolic blood pressure (DBP), cigarette smoking status, total and HDL cholesterol, and diagnoses of diabetes and ECG-LVH. In the cases where the status of having diabetes and / or LVH is not known, then it was assumed that the subject did not suffer from those conditions. Function and scoring representation of the Framingham approach are described as follows.

The equation form is calculated with SBP or DBP and two separate equations are given for this result. The final results of both algorithms are similar, but SBP measurement has been chosen as it is more precise, and is the recommended measure in [39].

Systolic blood pressure equation:

diabetes = 0,1

smoking = 0,1

ECG-LVH=0,1

a = 11.1122 - 0.9119 * log(SBP) - 0.2767 * smoking - 0.7181 * log(cholesterol / HDL) - 0.5865 * ECG-LVH

m = a - 1.4792 * log(age) - 0.1759 * diabetes **for men**

m = a - 5.8549 + 1.8515 * $[log(age/74)]^2$-0.3758 * diabetes **for women**

μ = 4.4181 + m

σ = exp(-0.3155 - 0.2784*m)

u = (lg(t)-μ)/(σ)

p = 1 - exp(-e^u)

Diastolic blood pressure equation:

diabetes = 0,1

smoking = 0,1

ECG-LVH=0,1

a = 11.0938-0.8670 * log(DBP) - 0.2789 * smoking - 0.7142 * log(cholesterol / HDL) - 0.7195 * ECG-LVH

m = a - 1.6346 * log(age) - 0.2082 * diabetes **for men**

m = a - 6.5306 + 2.1059 * $[log(age/74)]^2$ - 0.4055 * diabetes **for women**

μ = 4.4284 + m

σ = exp(-0.3171 - 0.2825 * m)

u =(lg(t)-μ) / (σ)

p = 1 - exp(-e^{u})

As shown in Figure 12, the Framingham risk score is a system where the scores are assigned to risk factor values. The risk of developing cardiovascular disease is determined by calculating the percentage risk in the summation of all related points, and by finding the corresponding percentage risk in the chart. See the chart below to see what Framingham scoring charts look like.

1. Find points for each risk factor

Age (if female) (yr)				Age (if male) (yr)				HDL cholesterol			
Age	Points	Age	Points	Age	Points	Age	Points	HDL	Points	HDL	Points
30	−12	41	1	30	−2	48–49	9	25–26	7	67–73	−4
31	−11	42–43	2	31	−1	50–51	10	27–29	6	74–80	−5
32	−9	44	3	32–33	0	52–54	11	30–32	5	81–87	−6
33	−8	45–46	4	34	1	55–56	12	33–35	4	88–96	−7
34	−6	47–48	5	35–36	2	57–59	13	36–38	3		
35	−5	49–50	6	37–38	3	60–61	14	39–42	2		
36	−4	51–52	7	39	4	62–64	15	43–46	1		
37	−3	53–55	8	40–41	5	65–67	16	47–50	0		
38	−2	56–60	9	42–43	6	68–70	17	51–55	−1		
39	−1	61–67	10	44–45	7	71–73	18	56–60	−2		
40	0	68–74	11	46–47	8	74	19	61–66	−3		

Total cholesterol (mg/dl)				Systolic blood pressure (mm Hg)					Points	
Chol	Points	Chol	Points	SBP	Points	SBP	Points	Other factors	Yes	No
139–151	−3	220–239	2	98–104	−2	150–160	4	Cigarette smoking	4	0
152–166	−2	240–262	3	105–112	−1	161–172	5	Diabetes		
167–182	−1	263–288	4	113–120	0	173–185	6	Male	3	0
183–199	0	289–315	5	121–129	1			Female	6	0
200–219	1	316–330	6	130–139	2			ECG-LVH	9	0
				140–149	3					

2. Add points for all risk factors

____ (Age) + ____ (Total chol) + ____ (HDL) + ____ (SBP) + ____ (Smoking) + ____ (Diabetes) + ____ (ECG-LVH) = ____ (Total)

Note: *Minus points subtract from total.*

3. Look up risk corresponding to point total

	Probability (%)			Probability (%)			Probability (%)			Probability (%)	
Points	5 yr	10 yr	Points	5 yr	10 yr	Points	5 yr	10 yr	Points	5 yr	10 yr
≤1	<1	<2	9	2	5	17	6	13	25	14	27
2	1	2	10	2	6	18	7	14	26	16	29
3	1	2	11	3	6	19	8	16	27	17	31
4	1	2	12	3	7	20	8	18	28	19	33
5	1	3	13	3	8	21	9	19	29	20	36
6	1	3	14	4	9	22	11	21	30	22	38
7	1	4	15	5	10	23	12	23	31	24	40
8	2	4	16	5	12	24	13	25	32	25	42

4. Compare with average 10-year risk

Age	Probability (%)		Age	Probability (%)		Age	Probability (%)	
(yr)	Women	Men	(yr)	Women	Men	(yr)	Women	Men
30–34	<1	3	45–49	5	10	60–64	13	21
35–39	<1	5	50–54	8	14	65–69	9	30
40–44	2	6	55–59	12	16	70–74	12	24

HDL, high density lipoprotein; SBP, systolic blood pressure; ECG-LVH, left ventricular hypertrophy by electrocardiography.

Figure 12: Framingham Heart Study Coronary Heart Disease Risk Prediction Chart Adapted from Anderson et al. (1991)[39]

Bibliography

[1] Statistics Canada 2007. Leading causes of death. *http://www.statcan.gc.ca/daily-quotidien/101130/dq101130b-eng.htm.*

[2] Abdul Nazeer K. A. and M. P. Sebastian. Improving the accuracy and efficiency of the kmeans clustering algorithm. *International Conference on Data Mining and Knowledge Engineering (ICDMKE) Proceedings of the World Congress on Engineering (WCE-2009)*, vol. 1:pp. 978–988, 2009.

[3] Montgomery AA., Fahey T., Peters TJ., MacIntosh C., and Sharp DJ. Evaluation of computer based clinical decision support system and risk chart for management of hypertension in primary care: randomised controlled trial. *BMJ 2000*, 320(7236):pp. 686–690, 2000.

[4] Bors A.G. and Pitas I. Median radial basis functions neural network. *IEEE Trans. on Neural Networks*, vol. 7:pp. 1351–1364, 1996.

[5] Akobeng AK. Understanding diagnostic tests 1: sensitivity, specificity and predictive values. *Acta Paediatr*, 96:pp. 338–341, 2006.

[6] Hunt E. B. *Concept Learning: An Information Processing Problem.* Wiley., 1962.

[7] Liu Bing-xiang and Cheng Xiang. An incremental algorithm of support vector machine based on distance ratio and k nearest neighbor. *IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 1:pp. 18 – 20, 2011.

[8] Mu chun Z. Face recognition based on fastica and rbf neural networks. *2008 International Symposium on Information Science and Engieering*, vol. 1:pp. 588 –592, 2008.

[9] Rumelhart D.E., Hinton G.E., and Williams R.J. Learning representations by back-propagating errors. *Nature*, vol. 323:pp. 533–536, 1986.

[10] Powers D.M.W. Evaluation: from precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2:pp. 37–63, 2011.

[11] Schapire R. E. The strength of weak learnability. *Machine Learning*, vol. 5:pp. 197–227, 1990.

[12] Utgoff P. E. Id: An incremental id3,. *Proceedings of the Fifth National Conference on Artificial Intelligence. San Francisco*, pages pp. 107–120, 1988.

[13] Utgoff P. E. Incremental induction of decision trees. *Machine Learning*, vol. 4:pp. 161–186, 1989.

[14] Roli F., Giacinto G., and Vernazza G. *Methods for Designing Multiple Classifier Systems, Lecture Notes in Computer Science*. Springer, 2001.

[15] Zhu F., Ye N., Pan D., and Ding W. Incremental support vector machine learning: an angle approach. *Fourth International Joint Conference on Computational Sciences and Optimization (CSO)*, pages 288 – 292, 2011.

[16] Assmann G., Cullen P., and Schulte H. Simple scoring scheme for calculating the risk of acute coronary events based on the 10-year follow-up of the prospective cardiovascular munster (procam) study. *Circulation 2002*, vol. 105:pp. 310–315, 2002.

[17] Bors A. G. and Gabbouj M. Minimal topology for a radial basis function neural network for pattern classification. *Digital Signal Processing: a review journal*, vol. 4:pp. 173–188, 1994.

[18] Danaei G., Ding EL., Mozaffarian D., Taylor B., Rhem J., Murray CJ., and Ezzati M. The preventable causes of death in the united states: comparative risk assessment of dietary, lifestyle, and metabolic risk factors. *PLoS Med.*, vol. 6:pp. e1000058, 2009.

[19] Cooper G.F. and Herskovits E. A bayesian method for the induction of probabilistic networks from data. In *Machine Learning*, pages pp. 309–347. Klumer Academic Publishers, 1992.

[20] McLachlan GJ. Mahalanobis distance. *Resonance*, vol. 4:pp. 20–26, 1999.

[21] Michael Glick. Screening for traditional risk factors for cardiovascular disease. *J Am Dent Assoc 2002*, vol. 133:pp.291–300, 2002.

[22] S. Grossnerg. Nonlinear neural networks: principles, mechanisms and architectures,. *Neural Networks*, vol. 1:pp. 17–61, 1988.

[23] Alizadeh H., Minaei-Bidgoli B., and K.Amirgholipour S. A new method for improving the performance of k nearest neighbor using clustering technique. *Jounal of Convergence Information Technology*, vol. 4:pp.84–92, 2009.

[24] Ian H.W. and Eibe F. *Data Mining Practical Machine Learning Tools and Techniques with JAVA Implementations*. Morgan Kaufmann Series in Data Management Systems, 1999.

[25] Kuncheva L. I. *Combining Pattern Classifiers Methods and Algorithms*. John Wiley & Sons, Inc., Hoboken, New Jersey., 2004.

[26] Dang J., Wang Y., and Zhao S. Face recognition based on radial basis function neural networks using subtractive clustering algorithm. *The 6th World Congress on Intelligent Control and Automation,*, vol. 2:pp. 10294 – 10297, 2006.

[27] Frawley W. J., Piatetsky-Shapiro G., and Matheus C. J. Knowledge discovery in databases: An overview. *AI Magazine*, vol. 13(3):pp. 57–70, 1992.

[28] Han J. and Kamber M. *Data Mining Concepts and Techniques*. Diane Cerra, 2006.

[29] P*ä*rkk*ä* J. Personalization algorithm for real-time activity recognition using pda, wireless motion bands, and binary decision tree. *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14:pp. 1211 – 1215., 2010.

[30] Schlimmer J. and Fisher D. A case study of incremental concept induction,. *Proceedings of the Fifth National Conference on Artificial Intelligence*, pages pp. 496–501, 1986.

[31] Meigs JB., Cagliero E., Dubey A., Murphy-Sheehy P., Gildesgame C., and et. al Chueh H. A controlled trial of web-based diabetes disease management: the mgh diabetes primary care improvement project. *Diabetes Care 2003*, 26(3):750–757, 2003.

[32] Hongmei Yand Yingtao Jiang, Jun Zheng, Chenglin Peng, and Qinghui Li. A multilayer perceptron-based medical decision support system for heart disease diagnosis. *Expert Systems with Applications*, vol. 30:pp. 272–281, 2006.

[33] Vasantha Rani S.P. Joy and Kanagasabapathy P. Multilayer perceptron neural network architecture using vhdl with combinational logic sigmoid function. *ICSCN '07. International Conference on Signal Processing, Communications and Networking.*, pages 404–409, 2007.

[34] Quinlan J.R. Induction of decision trees. *Machine Learning*, vol. 1:pp. 81–106, 1986.

[35] Quinlan J.R. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, Los Altos, 1993.

[36] Eichler K., Puhan MA., Steurer J., and Bachmann LM. Prediction of first coronary events with the framingham score: a systematic review. *Am Heart J*, vol. 153:pp. 722–31, 2007.

[37] Polat K. and Güne S. Classification of epileptiform *EEG* using a hybrid system based on decision tree classifier and fast fourier transform. *Applied Mathematics and Computation*, vol. 187:pp. 1017 – 1026., 2007.

[38] Pyörälä K., de Backer G., and Graham I. et. al. Prevention of coronary heart disease in clinical practice: recommendations of the task force of the european society of cardiology. *Eur Heart J*, vol. 15:pp. 1300–1331, 1994.

[39] Anderson KM., Odell PM., Wilson PW., and Kannel WB. An updated coronary risk profile: a statement for health professionals. *Circulation. 1991*, vol. 83:pp. 357–363.

[40] Breiman L. Bagging predictors. *Machine Learning*, vol. 524(2):pp. 123–140, 1996.

[41] Breiman L., Friedman J., Olshen R., and Stone C. Classification and regression trees. *Wadsworth Int. Group*, 1984.

[42] Yihua Liao and V.Rao Vemuri. Use of k-nearest neighbor classifier for intrusion detection. *Computers & Security*, vol. 21:pp. 439–448., 2002.

[43] Glick M. Screening for traditional risk factors for cardiovascular disease. *J Am Dent Assoc 2002*, vol. 133:pp. 291–300, 2002.

[44] Goebel M. and Gruenwald L. A survey of data mining and knowledge discovery software tools. *SIGKDD Explorations, ACM SIDKDD*, vol. 1(1):pp. 20–33, 1999.

[45] Hall M., Frank E., Holmes G., Pfahringer B., Reutemann P., and Witten I. H. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.

[46] Keller J. M., Gray M. R., and Givens J. A. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 15:pp. 580–585, 1985.

[47] McCloskey M. and Cohen N. Catastrophic interference in connectionist networks: the sequential learning problem. *The psychology of learning and motivation,*, vol. 24:pp. 109–164, 1999.

[48] Nargundkar M. The canadian heart health database, 1986-92. *DAIS Metabase and Heart Health in Canada CD-ROM.*

[49] Sanner R. M. and Slotine J. E. Gaussian networks for direct adaptive control. *IEEE Trans. On Neural Networks*, vol. 3:pp. 837–863, 1994.

[50] Sarkar M. and Leong TY. Application of k-nearest neighbors algorithm on breast cancer diagnosis problem. *Proc AMIA Symp.*, pages pp. 759–763., 2000.

[51] Vijayakumar M., Prakash S., and Parvathi R.M.S. Inter cluster distance management model with optimal centroid estimation for k-means clustering algorithm. *WSEAS Transactions on Communications*, vol. 10, 2011.

[52] Woodward M., Brindle P., and Tunstall-Pedoe H. Adding social deprivation and family history to cardiovascular risk assessment: the assign score from the scottish heart health extended cohort (shhec). *Heart 2007*, vol. 93:pp. 172–176, 2007.

[53] Littlestone N. and Warmuth M. Weighted majority algorithm. *IEEE Symposium on Foundation of Computer Science.*, 1989.

[54] Domingos P. and Pazzani M. On the optimality of the simple bayesian classifier under zero-one loss. *Mach Learning*, vol. 29::pp. 103–130., 1997.

[55] Conroy R., Py*örälä* K., Fitzgerald AP., Sans S., Menotti A., De Backer G., De Bacquer D., Ducimetière P., Jousilahti P., Keil U., Njlstad I., Oganov RG.,

Thomsen T., Tunstall-Pedoe H., Tverdal A., Wedel H., Whincup P., Wilhelmsen L., Graham IM., and SCORE project group. Estimation of ten-year risk of fatal cardiovascular disease in europe: the score project. *Eur Heart J*, vol. 24:pp. 987–1003, 2003.

[56] Jackson R. Updated new zealand cardiovascular risk-benefit prediction guide. *BMJ*, vol. 320:pp. 709–710, 2000.

[57] Kohavi R. and Provost F. Glossary. machine learning. vol. 30:pp. 271–274, 1998.

[58] Polikar R. and Gangardiwala A. Dynamically weighted majority voting for incremental learning and comparison of three boosting based approaches. *Proceedings. 2005 IEEE International Joint Conference on Neural Networks (IJCNN 2005)*, vol. 2:pp. 1131 – 1136, 2005.

[59] Polikar R., Udpa L., Udpa L., and Honavar V. Learn++: An incremental learning algorithm for supervised neural networks. *IEEE Trans. on System, Man and Cybermetics (C)*, vol. 31:pp. 497–508, 2001.

[60] Preis S. R., Hwang S., Coady S., Pencina M.J., D'Agostino R.B., Savage P. J.and Levy D., and Fox C.S. Trends in all-cause and cardiovascular disease mortality among women and men with and without diabetes mellitus in the framingham heart study, 1950 to 2005. *The American Heart Association 2009*, vol. 119:pp. 1728–1735, 2009.

[61] Qahwaji R., Al-Omari M., Colak T., and Ipson S. Using the real, gentle and

modest adaboost learning algorithms to investigate the computerised associations between coronal mass ejections and filaments. *in Proceedings of the International Conference on Communications, Computers and Applications*, pages pp. 37–42, 2008.

[62] Safavian S. R. and Landgrebe D. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man and Cybernetics*, vol. 21:pp. 660–674, 1991.

[63] Schapire R.E., Freund Y., Bartlett P., and Lee W.S. Boosting the margin: a new explanation for the effectiveness of voting methods. *Annals of Statistics*, vol. 26(5):pp. 1651–1686, 1998.

[64] Milne RJ, Gamble GD, and Whitlock G et al. Framingham heart study risk equation predicts first cardiovascular event rates in *N*ew *Z*ealanders at the population level. *NZ Med J*, vol. 116:pp. 1185., 2003.

[65] DAgostino S., Grundy S., Sullivan LM., and Wilson P. for the chd risk prediction group. validation of the framingham coronary heart disease prediction scores: results of a multiple ethnic groups investigation. *JAMA.*, vol. 286::pp. 180–187., 2001.

[66] Lawrence S. and Giles C.L. Overfitting and neural networks: Conjugate gradient and backpropagation. *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN 2000).*, vol. 1:pp. 114–119, 2000.

[67] Ramaswamy S. and Golub T.R. Dna microarrays in clinical oncology. *Clinical Oncology*, vol. 20:pp. 1932–1941, 2002.

[68] DAgostino RB Sr, Vasan RS, Pencina MJ, Wolf PA, Cobain M, and Massaro JM et. al. General cardiovascular risk profile for use in primary care: the framingham heart study. *Circulation 2008*, vol. 117:pp. 743–753., 2008.

[69] Auld T., Moore A. W., and Gull S. F. Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, vol. 18:pp. 223–239, 2007.

[70] Auld T., Moore A. W., and Gull S. F. Predicting motor vehicle collisions using bayesian neural network models: An empirical analysis. *Accident Analysis & Prevention*, vol. 39:pp. 922–933, 2007.

[71] Kidera T., Kobe Univ., Ozawa S., and Abe S. An incremental learning algorithm of ensemble classifier systems. *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, pages pp. 3421 – 3427, 2006.

[72] Poggio T. and Girossi F. Networks for appoximation and learning. *Proc. IEEE*, vol. 78:pp. 1481–1497, 1990.

[73] Dawber T.R., Meadors G. F., Moore F. E., National Heart Institute, National Institutes of Health, Public Health Service, D. C. Federal Security Agency, Washington, and Epidemiological Approaches to Heart Disease. The framingham study presented at a joint session of the epidemiology, health officers, medical care, and statistics sections of the american public health association, at the seventy-eighth annual meeting in *S*t. *L*ouis, *M*o., *N*ovember 3, 1950.

[74] Fayyad U., Piatetsky-Shapiro G., and Smyth P. Knowledge discovery and data

mining: Towards a unifying framework. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR,*, vol. 13(3):pp. 82–88, 1996.

[75] Bourd'es V., Bonnevay S., Lisboa P., Defrance R., Perol D., Chabaud S., Bachelot T., Gargi T., and Negrier S. Comparison of artificial neural network with logistic regression as classification models for variable selection for prediction of breast cancer patient outcomes. *Advances in Artificial Neural Systems*, vol. 2010, 2010.

[76] Wilson P. W.F., DÁgostino R.B., Levy D., Belanger A.M., Silbershatz H., and Kannel W.B. Prediction of coronary heart disease using risk factor categories. *Circulation 1998*, vol. 97:pp. 1837–1847, 1998.

[77] Pan X. and Wu J. Bayesian neural network ensemble model based on partial least squares regression and its application in rainfall forecasting. *International Joint Conference on Computational Sciences and Optimization*, vol. 2:pp. 49–52, 2009.

[78] Deli Y., Özyilmaz L., and Yildirim T. Evolutionary algorithms based rbf neural networks for parkinsons disease diagnosis. *7th International Conference on Electrical and Electronics Engineering (ELECO)*, pages II–311 – II–315, 2011.

[79] Freund Y. and Schapire R.E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, vol. 55:pp. 119–139, 1997.