uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Dawn Rybchynski
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)
GRADE / DEGREE

School of Information Technology and Engineering
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Design of an Artificial Neural Network Research Framework to Enhance the Development of Clinical Prediction Models

TITRE DE LA THÈSE / TITLE OF THESIS

M. Frize
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

R. Goubran

M. Yagoub

Gary W. Slater
LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

Design of an Artificial Neural Network Research Framework to Enhance the Development of Clinical Prediction Models

Submitted by

Dawn Rybchynski

A thesis submitted to the Faculty of Graduate Studies and Research
In partial fulfillment of the requirements for the degree of

Master of Applied Science in Electrical Engineering

School of Information Technology and Engineering

University of Ottawa

Your file Votre référence
ISBN: 0-494-11399-5
Our file Notre référence
ISBN: 0-494-11399-5

Abstract

This thesis presents an Artificial Neural Network Research Framework (ANN RFW) for predicting medical outcomes. The ANN RFW along with other new and pre-existing applications, and the steps linking them are presented as part of an Outcome Prediction Model Definition Process (OPMDP). Proof-of-concept experiments are performed on three outcomes from two Canadian Neonatal Network (CNN) databases.

Successful results were obtained from the ANN RFW and a number of the subsequent applications. Results obtained using an ANN plus case based reasoner (CBR) were not yet favourable. In one of the intermediary steps, a modified method for extracting relative importance of ANN inputs was developed. The resulting relative weight results indicated that the importance of input variables of continuous outcomes may vary over the course of outcome's duration. Observing relative weights for three outcomes indicated that each outcome must have its own prediction model.

Acknowledgments

I would like to thank my supervisor Dr. Monique Frize for allowing me the freedom to direct the path of my work and for allowing me the opportunity to take time off from my studies to seek employment among her contacts in the clinical engineering world.

I would like to thank my parents, Elizabeth and Brian Rybchynski, for financing the purchase of the single most important factor to my finishing this thesis – my laptop. With the job I got through Dr. Frize's contacts, I will be able to pay them back.

I would also like to thank Christina Catley for her assistance in proofreading this work and for acting as a tester for the ANN RFW and CCVT.

Table of Contents

List of Figures

List of Tables

Nomenclature

APGAR	Activity, Pulse, Grimace, Appearance and Respiration
ANN	Artificial Neural Network
CBR	Case Based Reasoner
CBRS	Case Based Reasoner System
CDSS	Clinical Decision Support System
CHEO	Children's Hospital of Eastern Ontario
EPIC	Evidence-based Practice Identification & Change
k-NN	k-Nearest Neighbours
LOS	Length of Stay
MIRG	Medical Information-technology Research Group
MORT	Infant Mortality
NICU	Neonatal Intensive Care Unit
ODBC	Open DataBase Connectivity
PADS	PArents Decision Support
SNAP	Score for Neonatal Acute Physiology
SNAPPE-II	Score for Neonatal Acute Physiology – Version 2 with the Perinatal Extension
UML	Unified Modeling Language
VENT	Duration of Artificial Ventilation

1 Introduction

1.1 Motivation

1.1.1 Health Care Perspective

Data mining in health care has three main focuses: the first to define trends in illnesses to assist both administrative and clinical members of staff, the second to define indicators for development of prevention initiatives and education, the third to provide active real-time decision support in the clinical environment.

Clinical decision support systems (CDSSs) encompass a range of software tools that provide physicians with clinical statistics and other patient information to facilitate the making of a diagnosis and the creation of treatment plans. CDSSs can display vital statistics and trends, or can provide outcome predictions. These outcomes can be truly clinical (e.g. predicting mortality or respiratory disease) or administrative (e.g. estimating the number of hours artificial ventilation that will be required, or the length of stay in an intensive care unit; information that can be used for equipment and staff scheduling).

When clinical databases are mined for information, it is often years after the patients have left the health care facility, however, it is possible to use these past cases in an intelligent manner to assist the patients currently under clinician care. The information gleaned from these datasets increases clinical knowledge of illness trends and treatment successes. Automated data mining systems that use real-time clinical data and provide a predicted outcome can be of great value to clinicians, particularly when there is a possibility to change the course of treatment and the associated clinical outcome.

Many data mining tools function on a specific set of given information that they then apply to each subsequent case, without adding the information from the new cases into the base information. Therefore, it is necessary to retrain the prediction system either on a periodic basis or when changes are made to the accepted course of clinical treatment. For example, if a prediction model existed, that gave the prognosis of a cancer patient; it would be vital to retrain the system if a new cancer treatment that extended patient longevity was put into use. Additionally, it is good practice to regularly update the existing case base with new patient information and remove old cases.

1.1.2 Technical Perspective

The Medical Information technology Research Group (MIRG) uses an artificial neural network (ANN) application written in MatLab for mining medical data. MIRG researchers use the application to develop prediction models for clinical and administrative outcomes.

A new Artificial Neural Network Research Framework (ANN RFW) must aid in addressing the problem of predicting continuous outcomes with the ANN tool, which is not feasible with the ANN in its current state. Two complimentary methods were proposed to overcome this limitation: (1) A method of overlapping categorical ANN results, (2) the introduction of a k-nearest neighbours (k-NN) application which imputes missing values. While case based reasoning had previously been used to match full patient cases in an adult intensive care unit (ICU) for inspection by clinicians [30] and to

impute missing values in a Neonatal Intensive Care Unit (NICU) database [18], it had not previously been considered for specific outcome prediction.

1.2 Thesis Objective

This work has three key objectives, listed below:

1. To design and develop an Artificial Neural Network Research Framework (ANN RFW) to increase automation allowing for more complete exploration of the data when creating prediction models. A framework that operates quickly enough to enable the researcher to feasibly explore various ANN network results is essential. Researchers must be able to: (1) attempt multiple structures altering the number of hidden layers and number of nodes in the hidden layer (2) research of a number of individual sub-outcomes within an outcome that is not binary (i.e. a continuous outcome).
2. Using the large set of previously unavailable results:
    a. To attempt a simple collaborative method for single ANN outcome results: a committee of classifiers approach using ANNs trained on different structures as the committee members.
    b. To investigate the usefulness of two other collaborative methods for predicting continuous outcomes: (1) a hybrid of an ANN and a k-nearest neighbour (k-NN) matching case based reasoner (CBR) and (2) a range prediction solution which takes results from ANNs trained on different sub-outcomes created from a single continuous outcome and compares the overlapping end result.

3. To define an Outcome Prediction Model Definition Process (OPMDP) for MIRG using the new applications and processes developed for this thesis, and incorporating existing applications and processes where applicable. A functional integration of the applications and the linking steps between them is presented as a step towards defining the OPMDP.

1.3 Thesis Contributions

1. Limitations of weight extraction and relative weight calculation models for multilayer perceptron artificial neural networks (MLP ANNs) found in the literature were discovered. A modified relative weight calculation method was created to expand the functionality of an existing method to include the types of networks used frequently with medical data: non-linear MLP ANNs with one hidden layer where the hidden to output layer connection weights are non-similar.

2. Through investigations of the relative weights extracted using the modified model, three important points were demonstrated:
    a. Short-term, long-term and possibly mid-range-term divisions in a continuous outcome may be defined using the differences in relative weights of input variables of an ANN,
    b. Continuous outcomes may need to be treated as separate duration-based outcomes (short-term, long-term and any existing mid-range-terms) for prediction purposes due to different relative weights and different prediction models, and

c. Different outcomes, beyond those sub-outcome within a single continuous outcome, may have different input variable weights requiring that every outcome to be predicted be given its own prediction model.

3. A method of predicting continuous outcomes using an ANN plus CBR (artificial neural network plus a case based reasoner) collaboration in a similar fashion to imputing missing values in a database was attempted. The results were interesting to discover even though they were not yet deemed favourable.

4. Design and implementation of an Artificial Neural Network Research Framework (ANN RFW) that has improved automation and increased functionality led to the successful introduction of a committee of classifiers to increase the generalization prediction ability of ANNs and a majority rules range overlap solution to broaden the outcome presentation ability of ANNs.

1.4 Thesis Outline

Chapter 1: Introduction provides thesis motivation, objective and summary of contributions.

Chapter 2: Overview of Relevant Concepts provides a brief introduction to Clinical Decision Support Systems (CDSSs), artificial neural networks (ANNs), k-nearest neighbours (k-NN) matching, as well as collaborative data mining methods.

Chapter 3: Methodology provides a description of the various software applications developed to meet the objectives: an Artificial Neural Network Research Framework (ANN RFW), an application to verify single ANN prediction models and set up committee of classifiers (CCVT), and k-nearest neighbours-based (k-NN-based) case based reasoner (CBR) software to impute missing values and predict outcomes. A formalized Outcome Prediction Model Definition Process (OPMDP) for MIRG including the new applications is also presented.

Chapter 4: Verification of Applications and Process provides a description of the databases used to validate the Outcome Prediction Model Definition Process and the tools therein.

Chapter 5: Results follows the results for the steps of the new Outcome Prediction Model Definition Process for a database and three outcomes, comparing them with results obtained previously by MIRG.

Chapter 6: Discussion of Results provides conclusions, contributions to knowledge and future work.

2 Overview of Relevant Concepts

This chapter provides an introduction to clinical decision support systems (CDSSs) as well as data mining and outcome prediction methods. The two data mining methods discussed are artificial neural networks (ANNs) and k-nearest neighbours (k-NN) matching. Various methods of predicting outcomes are discussed using ANNs, k-NNs and a few collaborative data mining techniques.

2.1 Clinical Decision Support Systems (CDSSs)

Clinical Decision Support System (CDSS) are defined as any application used to help physicians make decisions about patient care. CDSSs can provide many different types of information to the user, including from simplest to most complex:

- Display vital statistics and lab results for a patient, presented together in an electronic patient record; [33]
- Find and Display patient cases that closely match vital statistics of a current case, presented such that the clinician can see how past cases progressed and better assess the current case; [37]
- Perform outcome prediction, presented as probability of occurrence, a continuous value or range of possible values; [33]
- Issue warnings or alerts to clinicians if adverse conditions are suspected through predictions. [37, 10]

- Provide on-line interfacing between clinicians, other healthcare staff members and families of patients in order to make educated and informed decisions on courses of treatment. [33, 84]

Clinical decision support systems (CDSSs) are becoming more and more popular, with increased evidence that their use reduces medical errors. They are being implemented in a diverse range of healthcare environments from acute care to ambulatory practice. The increased availability of large data repositories and data mining applications performing tasks from patient case matching and displaying to outcome and diagnosis predictions make the CDSS feasible. [7]

The focus of this thesis will be on outcome prediction and the appropriate and effective presentation of that outcome. A number of outcomes can be presented from administrative (e.g. hours on a ventilator or number of nursing care hours required) to clinical (e.g. the possibility of contracting a specific respiratory disease or mortality within the first month of life).

2.2 Performance Measures of Data Mining Methods

There exist many statistics used to measure the performance of any data mining method. Classification rate, sensitivity, specificity, average squared error (ASE), area under the ROC curve (C-index) and logarithmic sensitivity index are presented in this section. Table 2-1 shows the statistical split of results for any two-value outcome dataset. The

true and false positives and negatives are used throughout this section to describe and compare the statistical methods of performance measure.

Table 2-1: Confusion Matrix for Two-Value Classification Outcome

		Actual Outcome	
		Negative Outcome (-1 or 0)	Positive Outcome (+1)
Predicted Outcome	Negative Outcome (-1 or 0)	True Negative (**TN**)	False Negative (**FN**)
	Positive Outcome (+1)	False Positive (**FP**)	True Positive (**TP**)

A Priori statistics are used as benchmarks in performance measurements because they are consistent across data mining methods. The a priori statistics denote the outcome split of the actual data, i.e. if an outcome has 75% negative and 25% positive values, these do not change regardless of the data mining method or performance measure chosen. [21]

2.2.1 Correct Classification Rate (CCR/CR) and Constant Predictor (CP)

Correct Classification Rate (CCR or CR) is the total number of cases that were predicted correctly. i.e. if 35 of 75 cases with the outcome equal to -1 were predicted correctly and 20 of 25 cases with the outcome equal to +1 were predicted correctly (35+20=55 correct of a total 75+25=100 cases); the CR would be 55%: the number of each correctly classified divided by total number of cases.

$$CR = \frac{TN + TP}{TN + FN + TP + FP} \qquad \textbf{Equation 2-1}$$

The Constant Predictor (CP) is the highest number of cases predicted correctly if a constant value is used as a predictor. For example, if 75% of the data outcome is a -1 and 25% is a 1, -1 would be chosen as the predictor and the CP would then be 75%. A priori stats are used to calculate the constant predictor (CP).

2.2.2 Average Squared Error (ASE)

The Average Squared Error (ASE) is the difference between each predicted value and its corresponding actual value squared and averaged over all cases. In classification based prediction methods a numeric error is very difficult to calculate as the outcome is classified into one of a series of categories. Because of the difficulty in assigning a numeric distance to the error, ASE is not often used for categorical results.

$$ASE = \frac{(desired_i - actual_i)^2}{\sum_{i=allCases} (desired_i - actual_i)^2}$$ **Equation 2-2**

2.2.3 Sensitivity and Specificity

Sensitivity (the true positive rate) and specificity (the true negative rate) are often used to measure the performance of a data mining method when classification rate (CR) and average squared error (ASE) are not appropriate. To allow researchers to measure how many positives and how many negatives have been predicted correctly, sensitivity (true positive rate) and specificity (true negative rate) were introduced. Sensitivity is the measure of how many actual positives were predicted correctly while specificity is the

measure of how many actual negatives were predicted correctly. [21] (*See Table 2-1 for a description of TP, FP, TN, FN*)

$$sensitivity = \frac{TP}{TP + FN}$$ **Equation 2-3 [21]**

$$specificity = \frac{TN}{TN + FP}$$ **Equation 2-4 [21]**

Often a two value classification outcome has an imbalance in the result (e.g. the dataset is split with 90% negatives and 10% positives). By convention, the less frequent classification is defined as the positive. Traditional performance measures fail to give a good indication of the effectiveness of a prediction model when the classification is highly imbalanced. For example, say we have the outcome mortality/survival with a 10%/90% split. The outcome is set "Mortality" with the cases that died been set as the positives. For this outcome a constant predictor (CP) of all negatives gives a classification rate (CR) of 90%. At first glance this appears to be a good predictor with 90% of the cases correctly classified. Upon closer inspection, it is seen that the less frequent value is never predicted correctly i.e. by predicting all negative values, all the positives have been classified incorrectly. The reality for most outcomes is that it is more important to predict the less frequent occurrence correctly. [81]

In medical data, the outcomes are often highly skewed, meaning there is very often a much less frequent classifier. Predicting an infrequent case can be very difficult and often that less frequent outcome is the more serious therefore, prediction models that predict less frequent outcomes with greater success are often chosen. [8] When these

predictions are used to assist clinicians in directing courses of treatment, the less frequent case generally causes a change in the normal actions.

Similar issues are seen in other fields, such as an oil spill prediction for open ocean surfaces. Images taken of the ocean's surface show many instances with only a few being oil spill with the remaining being natural and generally weather induced. The oil spill being missed is of far greater concern than a weather patch being mistaken as a spill and warranting a physical inspection. [81]

2.2.4 Receiver Operator Characteristics (ROC) Curves

The receiver operator characteristic (ROC) curve is a "...simple yet complete empirical description of [a] decision threshold effect." [57] Generally, the area under the ROC curve is used for comparison rather than the curve itself – the C - index. The larger the C-index, the better the prediction. [43] ROC analysis is most often used on outcomes that comprise nearly equal amounts of positive and negative cases. [57]

ROC curves are drawn using one of two methods. The first method is to use the probability of correctness for each predicted case used in data mining methods which give a probability of a certain outcome being correct such as with Bayes rule. [81] The second method uses specificity and sensitivity and is used primarily in data mining methods that do not provide probabilities for the outcomes predicted.

In a multilayer perceptron artificial neural network (MLP ANN), such as the one used by the Medical Information-technology Research Group (MIRG), results are given as a real number between -1 and +1 with no attached probability of correctness. The real value result is then categorized into a positive or a negative based on a specific cut-off point in the continuous range -1 to +1. The standard choice for the cut off point is to be half way between the minimum numeric minimum and maximum returned from the ANNs output transfer function. i.e. If the ANN returns a real continuous value between -1 and +1, 0 as the midpoint would be chosen as the cut off point. To plot the ROC, the cut-off point is moved and the sensitivity and specificity at each cutoff point is calculated. The results are best when the ROC curve is a step function and the results are of a redundant classifier when the ROC curve is at a 45 degree angle [21].

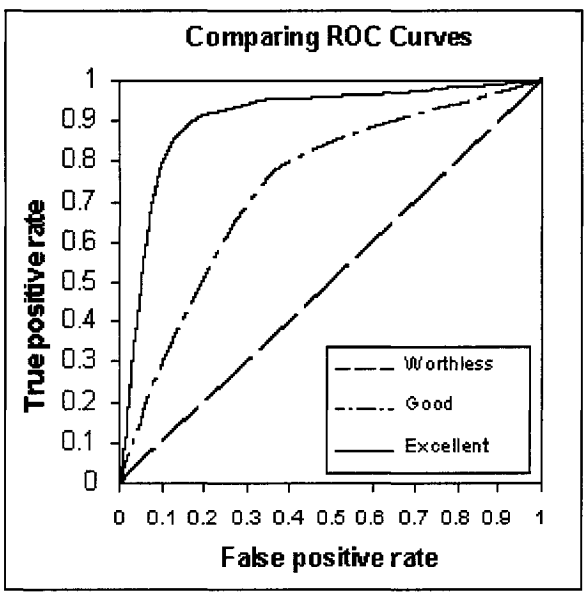

Figure 2-1: ROC curves. [69]

The true positive rate (sensitivity) is the number of positives correctly classified divided by the total number of actual positives. The false positive rate (1 – specificity) is one minus the number of negative correctly classified divided by the total number of actual negatives.

The **Worthless** curve in Figure 2-1 indicates that the sensitivity and specificity are both low and change greatly as the cut point moves. In an ANN with a hyperbolic tangent (tanh) transfer function (shown in Figure 2-2), an ROC curve along the 45 degree line indicates that the ANN results are all in the sloped region in the middle on the tanh curve where very small changes in the input create large changes in the output. This is not a very well generalized or stable prediction environment.

The **Excellent** curve in Figure 2-1 indicates that the sensitivity and specificity values are both high and do not change greatly as the cut point moves. In an ANN with a hyperbolic tangent (tanh) transfer function (shown in Figure 2-2), an ROC curve that is nearly a right angle (a step function) indicates that most of the results are in the tail regions of the tanh curve where larger changes are needed in the input to create changes in the output. This is a more stable and generalized prediction model.

An ROC curve that is a smooth arc indicates that the changing cut off point changes the positive and negative result division. An ROC curve that is at a sharp angle indicates that the changing cut off point does not change the positive and negative result division as the points are all collected together in one spot.

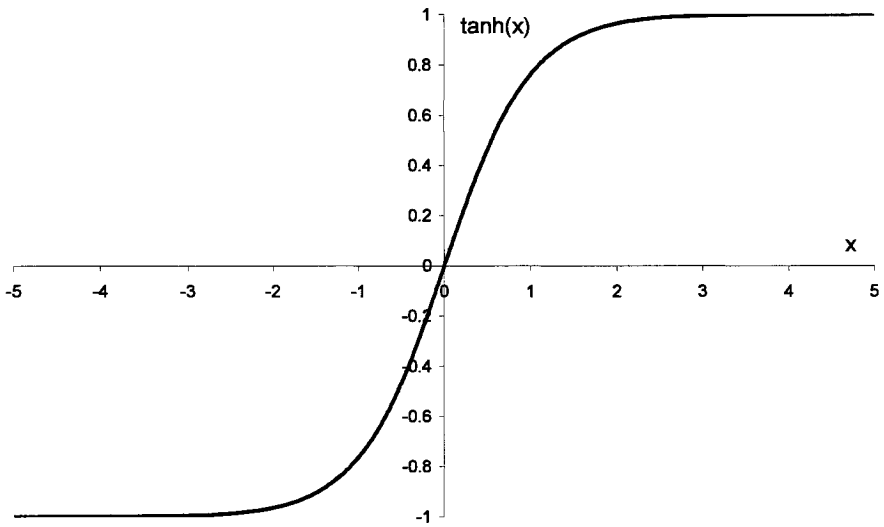

Figure 2-2: Hyperbolic Tangent transfer function.
x = the argument of the tanh function

2.2.5 Logarithmic Sensitivity Index

For most outcomes, the impact of choosing the positive or negative of an outcome incorrectly is different. For example, predicting a patient outcome as mortality incorrectly can have a higher impact on the patient's course of treatment than incorrectly predicating survival. [18]

The performance measure portion of the stopping criterion is used to measure how well the network performs at any given epoch. The dataset's specific distribution over the chosen outcome is an important factor when choosing a stopping criterion. Those datasets that have an outcome that is highly skewed in favour of one output (e.g. 10% positives and 90% negatives) are difficult to train. The logarithmic sensitivity index was

created to give more weight to the sensitivity and therefore better predict the less frequent outcome. [24]

$$\textbf{\textit{logsens}} = -\textit{sensitivity}^{n} * \log(1 - \textit{sensitivity} * \textit{specificity}) \qquad \textbf{Equation 2-5 [24]}$$

n: a factor to increase the results toward predicting a higher sensitivity

2.3 Artificial Neural Networks (ANNs)

An artificial neural network (ANN) is modeled after the function of biological neurons. Figure 2-3 shows an ANN with no hidden layer and an ANN with one hidden layer containing three hidden nodes.

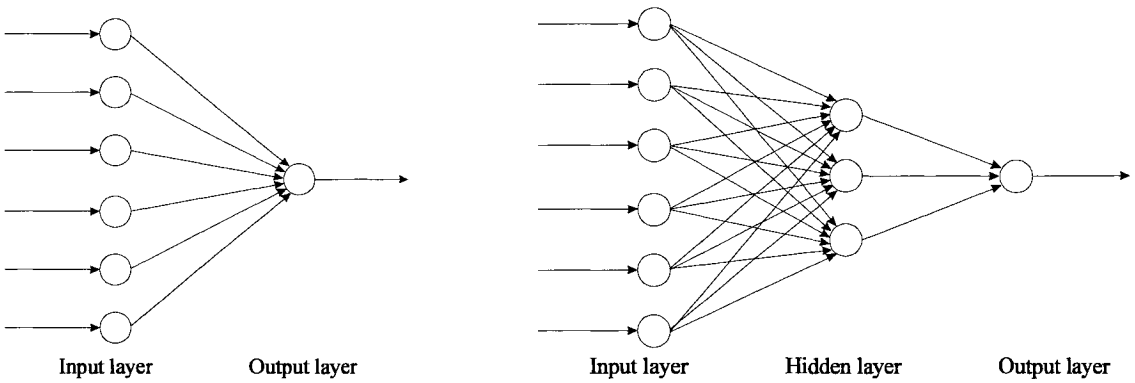

Figure 2-3: (*left*) A Single layer network (2-layer network), (*right*) a Double layer network (3-layer network) [60]

Every node in the hidden and output layers operates modeled after a neuron in physiology. In mammalian neurology, each neuron has a series of receptors (inputs) that sense electrical stimuli and an output that sends an electrical signal if triggered. An electrical stimulus can be either excitatory (adds to triggering a response) or inhibitory (subduing possible response). If the overall combination of all excitatory and inhibitory

stimuli reaches the trigger point, the neuron sends a signal out through the axon to the inputs of other neurons. This biological process is mathematically adapted by ANNs.

Inputs to a node, in combination with the connection weight between the node and the input values, are summed. The sum is then presented to a transfer function. In a linear network (no hidden layer), that transfer function is a step function. In a non-linear network (at least one hidden layer) the transfer function is not a step function but a non-linear mapping function. Hyperbolic tangent and sigmoid functions are most commonly used. The function used in this work is the hyperbolic tangent function. Each connection between nodes in the network has a weight. This weight defines how strongly that particular input will affect the node it is connected to. An example ANN node is shown in Figure 2-4. The bias term is summed along with the input/weight multiplied results. The bias provides for an offset between nodes in a network. A greater the magnitude of the bias the greater the offset effect to the node and the network.

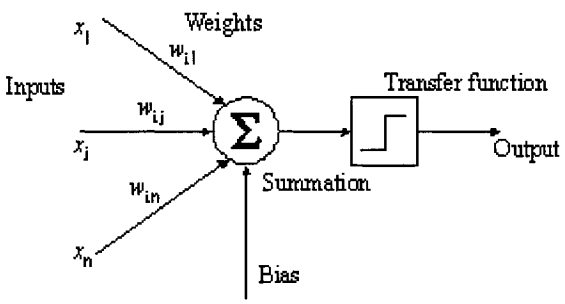

Figure 2-4: An ANN node (neuron). [60]

A limitation of ANNs is that ANN outputs are made binary through the use of a transfer function and subsequent categorization result. The transfer function used may provide a

real value as an output, but that value is categorized based on a cut point. Any value greater than the cut point is categorized as a positive and any value less than the cut point is categorized as a negative. Training an ANN on results for an outcome that is not two-valued (binary) can be difficult and generally requires a domain expert to assist with the significance of different regions of a continuous outcome.

An ANN is architecturally defined by two factors: its structure and its connectivity. An ANN structure is the number of hidden layers and number of nodes in each layer. An ANN's connectivity can be either fully-connected or partially-connected. Fully connected ANNs have each node of a layer connected to every node in both the next and previous layers (as shown in Figure 2-3 (*right*)). Partially connected ANNs can have any combination of nodes connected across any layers.

A feed forward network has all result calculations directed from input to output. An ANN with connections that route the result of a node back to previous layer are feed backward or recurrent networks.

A back propagation learning algorithm updates connection weights and biases after the error has been calculated backward from output layer to input layer. The back propagation algorithm is responsible for the learning of the network. It changes the weights and biases of the network as a function of the error calculated from the results. The changes are made to follow a descending gradient search along the error function in an attempt to minimize the error. The error function is often created of many hills (local

maxima) and valleys (local minima). The gradient search along the error function does not guarantee that the direction traveled will lead to the global minima on the error curve.

2.3.1 Driving Parameters

Nine driving parameters control the fashion in which the ANN learns from one epoch to the next. These parameters change the weights and biases of the network as a function of the error between the desired and calculated outputs.

"*Learning rate (lr)*: The value of the learning rate determines the speed at which the network attains a minimum in the criterion function so long as it is small enough to insure convergence. If the learning rate is too high, it may oscillate around the global minimum, and is unable to converge.

Learning rate increment (lr_inc): The learning rate's incremental value.

Learning rate decrement (lr_dec): The learning rate's decrement value.

Weight-decay constant (λ): The weight-elimination constant determines how strongly the weights are penalised.

Weight-decay constant increment (λ_inc): The weight-decay constant's incremental value.

Weight-decay constant decrement (λ_dec): The weight-decay constant's decrement value.

Weight-elimination scale factor (w_o): Weight-elimination scale factor defines the sizes of "large" and "small" weights. When w_o is small, the small weights will be forced to zero resulting in fewer large weights (i.e., weight-

elimination). A large w_o causes many small weights to remain and limits the size of large weights (i.e., weight-decay).

Momentum (momentum): The momentum parameter adds a proportion of the previous weight-change value to the new value, thereby giving the algorithm some "momentum" to prevent it from getting caught in local minima.

Error ratio (err_ratio): The error ratio controls how the backpropagation makes adaptive changes in the learning rate, the weight-decay constant, and the momentum term." [23]

The parameters learn rate, the learn rate incremental multiplication factor, the learn rate decremental multiplication factor, momentum and error ratio are always used in the ANN learning algorithm. Lambda, the lambda incremental multiplication factor, the lambda decremental multiplication factor and the weight scale are used when the weight elimination cost function is turned on.

2.3.2 Weight Decay and Weight Elimination

The role of the weight decay constant (lambda, λ) is to determine how strongly weights are penalized. Weight elimination adds the following penalty term in addition to the main mean squared error function [77], this is the weight elimination cost function.

$$\lambda \times \sum_i \frac{\frac{w_i^2}{w_0^2}}{1+\frac{w_i^2}{w_0^2}} \qquad \textbf{Equation 6 [77]}$$

Weight decay limits the size of the connection weights penalizing large weights by forcing them to not increase. Weight decay creates outputs with less variance in turn

creating a more stable network. Weight elimination tries to reduce the small weights to zero. Following from this, an approximation of connection weight importance can be made that the smaller the connection weight, the smaller the effect the connection has on the network result. Weight elimination can reduce connection weights to a level that the node effectively has been removed from the network. Weight decay and weight elimination work best when using a large initial network structure, small initial weights and a small learning rate. [21]

$$E(W) = E_0(W) + \lambda \times \sum_{ij} \frac{\frac{w_{ij}^2}{w_0^2}}{1 + \frac{w_{ij}^2}{w_0^2}}$$ **Equation 7 [21]**

Where:

E_0 = SSE (sum squared error).

λ = weight decay factor, used to determine the relative importance of the weight elimination term. A larger λ value means that a weight must be closer to zero to be considered part of the noise, increasing the pressure on small weights to further reduce their size.

w_{ij} = weight of connecting node i and j.

w_0 = weight scale factor, defines the sizes of large and small weights. When w_0 is small the small weights will be forced to zero resulting in fewer large weights (weight elimination), if w_0 is large many small weight remain and the large weights are limited (weight decay).

2.3.3 Artificial Neural Networks (ANNs) for Mining Medical Data

Artificial neural networks (ANNs) have been used extensively to perform data mining in medical environments. A strength of ANNs as a data mining method is their ability to find a result with no prior knowledge of the interaction of the variables. [59] The learning process of the ANN has often been referred to as a black box. In the last decade, methods for retrieving information about the input variables from their connection weights has somewhat illuminated the black box of ANNs. [59, 25]

The multilayer perceptron (MLP) ANN with one hidden layer is very popular for use in medical data mining. The hidden layer provides for a non-linear interaction between the variables, with hidden nodes acting as calculators of higher order statistics. There has been no evidence that using more than one hidden layer provides a better result than using one hidden layer when mining medical databases. [62] Medical data is believed to be non-linear in nature with many unknown interactions, therefore the use of an ANN with a hidden layer is a sound choice. [31]

2.3.4 Predicting Outcomes with an Artificial Neural Network (ANN)

An ANN can be used to predict an outcome by taking a structure with weights and biases and the appropriate mapping function normalization values that had previously been defined. Using these pieces, a new case can be normalized and then applied to the network with the outcome being calculated. Once this outcome has been calculated and categorized into a positive or negative the prediction has been complete. [17]

The first step to defining an ANN prediction model is to reduce the number of available input variables to the smallest possible subset. This data subset is the minimum dataset and represents the most important input variables to predicting a specific output. These variables are also known as indicators or factors leading to an outcome.

2.3.4.1 Input Variable Reduction: Finding a Minimum Dataset

Medical databases are generally large, containing many useable cases and more variables than are needed to predict the desired outcome. After the database has been cleaned (removal of outliers and ambiguous data), the first step is to pare down the database to a workable size by removing unimportant input variables. The difficulty lies in knowing which variables to remove. A method of finding the importance of each input variable to the prediction of the desired outcome is needed. When only the most important variables remain, the minimum dataset has been reached.

The minimum dataset not only defines the most important input variables for a prediction model, but also defines a list of indicators and their relative importance. When a database is mined, sometimes the ability to predict case-by-case results is not the goal. In these cases, often a set of indicators is desired in order to make over-all declarations about the factors that lead to certain results. The members of the minimum dataset with the largest relative weights are these factors or indicators. For example, a database containing information on injuries in children seen in the emergency room may be mined to find the indicators towards severe injuries received in ATV related incidents. [27]

There exist a number of methods for input variable elimination with ANNs. Methods of input variable reduction for use with back-propagation neural networks exist from complex to simple. Complex, math intensive examples will not be explored, including examples such as independent component analysis and higher order cross statistics [3], p-value test reduction [55], and complex dimensionality reduction [82]. Two popular simple methods of input variable reduction are sensitivity analysis [59, 55] and relative weight reduction [36, 54, 59].

Sensitivity analysis [59, 55] requires varying one input over its entire range and holding the remainder steady. Statistics are calculated to indicate which of the parameters or inputs most effect the results. One suggestion for simple sensitivity analysis involves removing a single input variable, calculating results, replacing that variable and removing the next variable until all input variables have been switched out. The result from each iteration is compared, and the variable whose removal had the least impact on the results is permanently removed from the minimum dataset. The single variable removal process begins again until the results begin to degrade. Sensitivity analysis, whether studied via complicated statistics for each input or across a series of single input removals is thorough but time consuming.

2.3.4.1.1 Relative Weight Reduction Method

Relative weight reduction has been used in many different fields from house price predictions [54] to ecological modeling of fish populations [1, 59]. A specific example proposed for medical databases [36] involves training a single layer ANN (no hidden

layers) on a full set of input variables. The magnitude of the connection weights between the input and output nodes are scaled to between 0 and 100%. The input nodes with the smallest weights are removed and a new single layer ANN training cycle is begun. The train-scale-remove cycle is repeated until degradation in the results is seen. The relative weights method is faster than sensitivity analysis with comparable simplicity. Due to the quicker operating times and its previous successful application in medical data mining, the relative weight reduction method was chosen for this work.

A weakness of the relative weight reduction method as proposed in [36] is its ability to deal with single layer networks only. A single layer network requires the relationships between across the ANN to be linear. Restricting ANN structures to single layer only prevents problems with non-linear solutions from using the relative weight reduction method. A method of calculating relative weights of multilayer perceptron artificial neural networks (MLP ANNs) is needed to include the non-linear ANNs. Such a method was proposed by Garson in 1991 and was presented by Goh in 1995 [41]. A comparison of the sensitivity analysis method and relative weight reduction using the Garson-Goh method concluded that the relative weight reduction to be the better of the two because of its simplicity and faster speed [59], adding support for choosing relative weight reduction for this work.

Garson's relative weight method finds the relative weight or importance of each input variable through a series of simple equations multiplying, adding and dividing the connection weights between the input nodes and the hidden nodes as well as the

connection weights between the hidden nodes and the output nodes. The equations for Garson-Goh method have been set forth in Figure 2-5.

A weakness of Garson's relative weight algorithm is its use of the absolute value of the weights. The use of the absolute value loses some of the interaction between weights stemming forward from the same input node and makes it possible for a result to be misread. In the presence of a high magnitude positive and a high magnitude negative weight stemming from the same input node, the two high magnitude values may effectively cancel each other out in the ANN calculation but are counted twice towards the relative importance of the input node. [59]

A second weakness of Garson's algorithm was discovered in this work. The method performs well for linear networks (no hidden layer) and for situations where the hidden-output connection weights are all of similar magnitudes. In situations with a non-linear network (with a hidden layer) with non-similar hidden-output connection weights, the equation in step (2) of Figure 2-5 effectively divides the hidden-output weights into a multiplication factor of 1. The effect of these non-similar hidden-output connections are not truly included in the weight extraction. Figure 2-5 shows the Garson-Goh equations used in an example as published by Goh [41]. After the figure, an elaboration of step (2) is shown.

Hidden neurons	Weights			
	Input 1	Input 2	Input 3	Output
Hidden 1	−1·67624	3·29022	1·32466	4·57857
Hidden 2	−0·51874	−0·22921	−0·25526	−0·48815
Hidden 3	−4·01764	2·12486	−0·08168	−5·73901
Hidden 4	−1·75691	−1·44702	0·58286	−2·65221

The computation process is as follows:

(1) For each hidden neuron i, multiply the absolute value of the hidden-output layer connection weight by the absolute value of the hidden-input layer connection weight. Do this for each input variable j. The following products P_{ij} are obtained:

	Input 1	Input 2	Input 3
Hidden 1	$P_{11} = 1{\cdot}67624 \times 4{\cdot}57857$	$P_{12} = 3{\cdot}29022 \times 4{\cdot}57857$	$P_{13} = 1{\cdot}32466 \times 4{\cdot}57857$
Hidden 2	$P_{21} = 0{\cdot}51874 \times 0{\cdot}48815$	$P_{22} = 0{\cdot}22921 \times 0{\cdot}48815$	$P_{23} = 0{\cdot}25526 \times 0{\cdot}48815$
Hidden 3	$P_{31} = 4{\cdot}01764 \times 5{\cdot}73901$	$P_{32} = 2{\cdot}12486 \times 5{\cdot}73901$	$P_{33} = 0{\cdot}08168 \times 5{\cdot}73901$
Hidden 4	$P_{41} = 1{\cdot}75691 \times 2{\cdot}65221$	$P_{42} = 1{\cdot}44702 \times 2{\cdot}65221$	$P_{43} = 0{\cdot}58286 \times 2{\cdot}65221$

(2) For each hidden neuron, divide P_{ij} by the sum for all the input variables to obtain Q_{ij}. For example for Hidden 1, $Q_{11} = P_{11}/(P_{11} + P_{12} + P_{13}) = 0{\cdot}266445$.

(3) For each input neuron, sum the product S_j formed from the previous computations of Q_{ij}. For example, $S_1 = Q_{11} + Q_{21} + Q_{31} + Q_{41}$.

	Input 1	Input 2	Input 3
Hidden 1	$Q_{11} = 0{\cdot}266445$	$Q_{12} = 0{\cdot}522994$	$Q_{13} = 0{\cdot}210560$
Hidden 2	$Q_{21} = 0{\cdot}517081$	$Q_{22} = 0{\cdot}228478$	$Q_{23} = 0{\cdot}254441$
Hidden 3	$Q_{31} = 0{\cdot}645489$	$Q_{32} = 0{\cdot}341388$	$Q_{33} = 0{\cdot}013123$
Hidden 4	$Q_{41} = 0{\cdot}463958$	$Q_{42} = 0{\cdot}382123$	$Q_{43} = 0{\cdot}153919$
Sum	$S_1 = 1{\cdot}892973$	$S_2 = 1{\cdot}474983$	$S_3 = 0{\cdot}632044$

(4) Divide S_j by the sum for all the input variables. Expressed as a percentage, this gives the relative importance or distribution of all output weights attributable to the given input variable. For example, for the input neuron 1, the relative importance is equal to $(S_1 \times 100)/(S_1 + S_2 + S_3) = 47{\cdot}3\%$.

	Input 1	Input 2	Input 3
Relative importance (%)	47·3	36·9	15·8

Figure 2-5: Garson-Goh Equations. [41, Appendix]

In step (2) of Figure 2-5, the weights from the hidden nodes have cancelled themselves out. An expansion of the equation in step (2) is shown below.

$$Q_{11} = \frac{1.67624 \times 4.57857}{(1.67624 \times 4.57857) + (3.29022 \times 4.57857) + (1.32466 \times 4.57857)}$$

$$Q_{11} = \frac{1.67624}{(1.67624) + (3.29022) + (1.32466)} \times \frac{4.57857}{4.57857}$$

$$Q_{11} = \frac{1.67624}{(1.67624) + (3.29022) + (1.32466)} \times 1$$

As the Hidden1 to output connection weight (4.57857) factors out to be 1, any non-zero and non-infinite value of Hidden1 to Output connection weight would yield the same result for Q_{11}. The same is true for all the Q values.

2.3.4.1.2 Relative Weight Calculation Equations

In an attempt to integrate the hidden-output weights into the algorithm while maintaining the basic approach of Garson-Goh's relative weight calculation, new equations have been developed in this work using the following parameters:

i: the number of input nodes.

j: the number of hidden nodes.

k: the number of output nodes (always one in our case).

w_{ij}: the connection weight between input node i and hidden node j.

w_{jk}: the connection weight between hidden node j and output node k.

rw_{ij}: the relative connection weight between input node i and hidden node j.

rw_{jk}: the relative connection weight between hidden node j and output node k.

RI_i: the Relative Importance of input node i.

Relative Weight Equations:

Step 1: Calculate the relative weighting of the hidden-output connections into the output node. For a single output node set k=1.

$$rw_{jk} = \frac{abs(w_{jk})}{\sum_{j} abs(w_{jk})} \qquad \textbf{Equation 2-8}$$

Step 2: Calculate the relative weighting of the input-hidden connections into each hidden node. For each hidden node j:

$$rw_{ij} = \frac{abs(w_{ij})}{\sum_i abs(w_{ij})} \qquad \textbf{Equation 2-9}$$

Step 3: Multiply each relative weighting input-hidden connection value by the relative weighting value of its connected hidden-output weight. For each input i:

$$P_{ij} = rw_{ij} * rw_{jk} \qquad \textbf{Equation 2-10}$$

Step 4: Sum the final relative weighting products (P_{ij}). For each input i:

$$S_i = \sum_i P_{ij} \qquad \textbf{Equation 2-11}$$

Step 5: Divide the sum of the final relative weightings for each input node by the total of all relative weightings. For each input i:

$$RI_i = \frac{S_i}{\sum_i S_i} \qquad \textbf{Equation 2-12}$$

An example of the relative weights calculated from a two layer ANN with nine inputs and the same training dataset is shown in Figure 2-6 and Figure 2-7 for an outcome titled 'Vgt0 outcome'. The results are shown for five different numbers of hidden nodes: 7, 11, 13, 17 and 18. Examples of the weights obtained by using the Garson-Goh equations is shown in Figure 2-6 and the newly developed relative weight equations Figure 2-7.

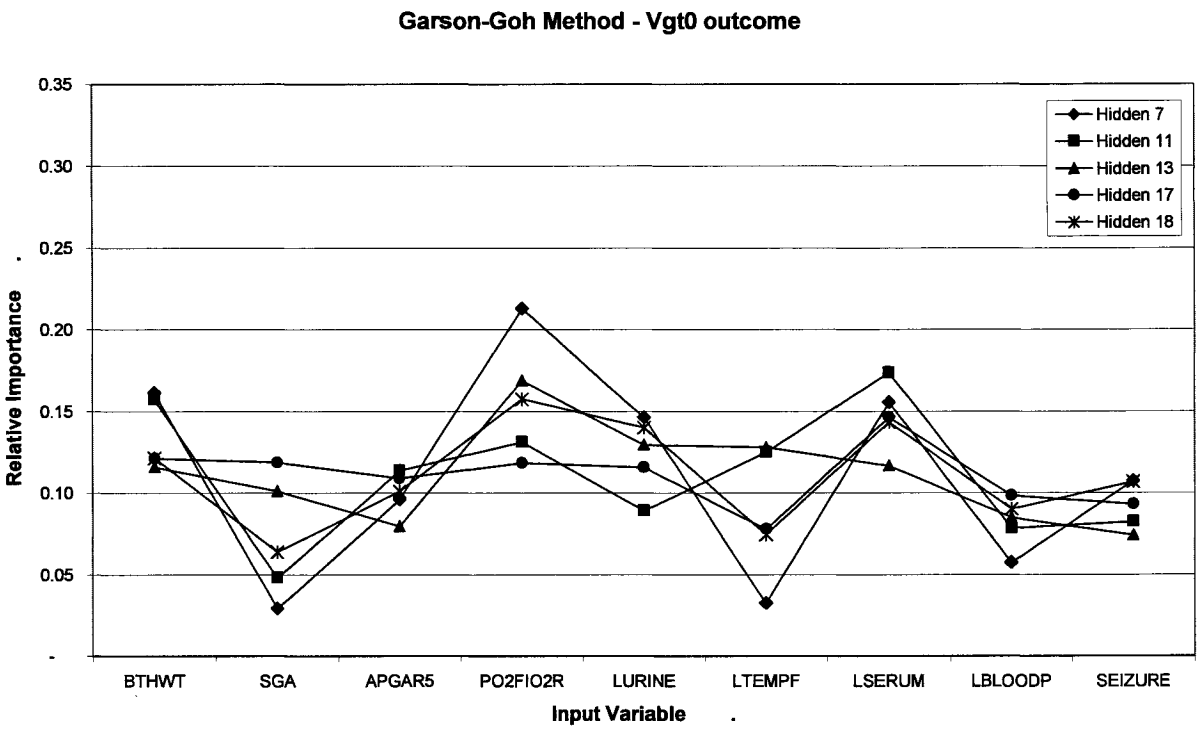

Figure 2-6: Example of Relative Importance from Garson-Goh Method.

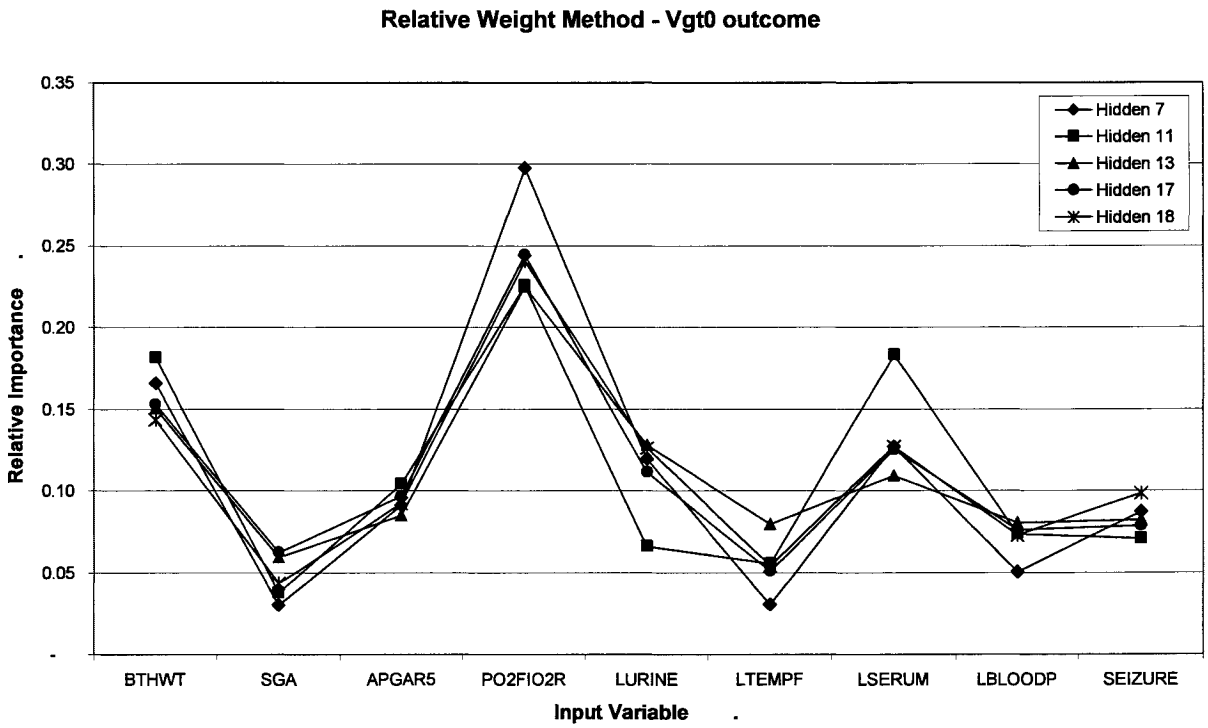

Figure 2-7: Example of Relative Importance from relative weights method.

The relative weight equations provide a more closely aligned set of relative importance values for the inputs than is seen with the Garson-Goh equations. The alignment is expected when training for the same outcome. The closer alignment makes it easier to define a final relative importance which includes all five structures. From these final relative importance values, inputs with the smallest relative importance can be removed until a minimum dataset is created.

Once a minimum data set has been defined the simplest outcome prediction method is to perform a single ANN prediction. These single ANN predictions can be used in collaboration to create a committee or ensemble.

2.3.4.2 Single Artificial Neural Network (ANN) Prediction

At every epoch of an ANN training run, weights and biases are set by the back propagation algorithm. Once data statistics (mean, standard deviation of each input variable) have been gathered, the prediction model definition is completed by extracting the structure (number of hidden layers and hidden nodes), and weights and biases at the epoch which gives the best results. The extracted weights and biases are assembled and a single epoch is executed on an input pattern to predict the outcome. The single epoch calculation method can be used to predict the outcome of an input pattern on a case-by-case basis in a clinical setting [63] or on mass data in the research environment.

The Medical Information-technology Research Group (MIRG) currently has a prototype ANN tool[a] used for clinical trial by Qi. [63] The weights and biases are retrieved from the final 'best' resulting epoch of the training ANNs and embedded into a tool that accepts clinical values from a user by way of a graphical user interface (GUI) in MatLab. The tool then normalizes the input case information appropriately, executes a single epoch and delivers a prediction for the requested outcome.

The ANN prototype provides a single ANN prediction environment. The GUI and supporting software have been designed for specific use to predict three outcomes from nine inputs in the Neonatal Intensive Care Unit (NICU) and therefore the change to a different dataset would be intensive. Future trials of NICU prediction models by way of single ANN, committee of classifiers, case based reasoning (CBR) with k-nearest neighbours (k-NN) or range overlap will be performed using the clinical user interface provided by PArents Decision Support (PADS). PADS is an ethical decision making tool providing user interfaces and interactive environments for clinicians and guardians of patients in the NICU [83]. For further description of PADS see section 3.1.

[a] ANN Prototype tool: Written in MatLab with a GUI designed to predict 5 outcomes (mortality, ventilation 0 hours, ventilation 24 hours, length of stay 7 days, length of stay 28 days) for 9 inputs (SNAPPE-II variables as described in section 4.1). Each outcome had a predefined ANN prediction model. Real-value input values are entered by hand and a prediction of each outcome is displayed to the screen with an option to save to a file: Created by Rybchynski in 2003. (The prediction model for 3 of the 5 outcomes were retrained on a larger dataset and updated for use by Qi in 2005 [63].)

2.3.4.3 Range overlap: Continuous Value Outcome Prediction with the Artificial Neural Network (ANN)

One of the limitations of a multilayer perceptron artificial neural network's (MLP ANN's) prediction ability is that the single binary output is not ideal for predicting a continuous valued (or any non-binary) output. Continuous outputs must be defined by inequalities based on ranges. For example: the outcome Length of Stay (LOS) in a Neonatal Intensive Care Unit (NICU) is a continuous value equal to the number of days a patient was in the NICU. The ANN requires that the outcome is a binary value (usually paired as 0 and 1 or -1 and +1). That binary requirement creates the possibility of a number of sub-outcomes each representing a single inequality. One such sub-outcome may be LOSgt7 days which would have all length of stay amounts greater than seven set to the positive and all values less than or equal to seven set to the negative result. e.g. LOS = 9days => LOSgt7 = 1 and LOS = 5days => LOSgt7 = -1.

Given the outcome length of stay (LOS) and four separate inequality based sub-outcomes LOSgt7, LOSgt14, LOSgt21 and LOSgt28 there are five possible range results: less than 8, 8-14, 15-21, 22-28 and greater than 28. Say that for a particular patient case, we have the results from the four separate sub-outcomes as follows:

- LOSgt7 = 1 (true),
- LOSgt14 = -1 (false),
- LOSgt21 = -1 (false) and
- LOSgt28 = -1 (false).

Using a simple majority rules voting technique the most popular range is found. The range overlap of this example is illustrated in Figure 2-8.

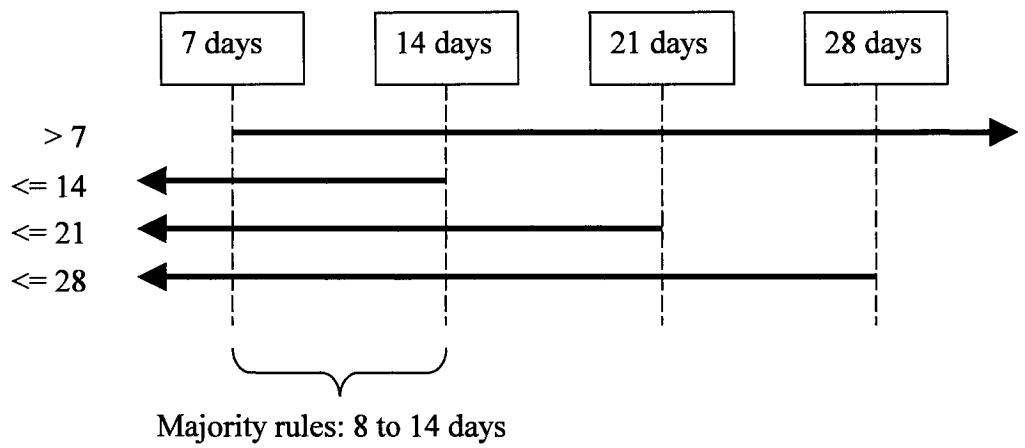

Figure 2-8: Range Overlap.

The range and votes from left to right are 3, 4, 3, 2 and 1. The range with the most votes is 8-14 days. This overlapped range result provides a more specific result than is obtained with a single inequality result.

2.4 K-Nearest Neighbour (k-NN) Matching

K-nearest neighbour distance algorithms effectively plot all points in k-dimensional space. The distance from the target case to all other cases in the case-base is calculated. The cases with the smallest distance to the target case are chosen as the most similar. In a weighted k-NN algorithm the axes of k-space are not uniformly set, input variables with higher weights are plotted on a new elongated axis, forcing the difference in values along

the axis to be more important. Input variables with lower weights are plotted along a shrunken axis forcing the differences between values on that axis to be less important.

One of the strengths of the k-NN classifier is that it deals with missing values. The variable with a missing value is not drawn on the axis for that particular case and a distance is still calculated. One of the weaknesses of using a weighted algorithm is the need to have predefined knowledge of the weights of the inputs.

The k-NN algorithm used by MIRG is a weighted matching algorithm based on the work of Mitchel [58] and adapted for use with user-defined weights by Cotea and Jiwani [16]. The adapted algorithm follows [16]:

i: the number of input variables.

n: number of cases in the Discharged Patient database used in the matching.

w_i: the weight of the input variable i.

max_i: the maximum value of the input variable i.

min_i: the minimum value of the input variable i.

t_i: value of the ith input variable of the Open Patient Case.

x_i: value of the ith input variable of the Discharged Patient Case.

Step 1: Find min and max of all input variables.

Step 2: Calculate the maximum distance.

$$\max\_dist_i = \sqrt{\sum_{i=1}^{n}[(\max_i * w_i) - (\min_i * w_i)]^2} \qquad \textbf{Equation 2-13}$$

Step 3: For each record in the Discharged Patient database, calculate the distance between the Discharged Patient record and the Open Patient Case record.

$$dist(t_i, x_i) = \sqrt{\sum_{i=1}^{n}[(t_i * w_i) - (x_i * w_i)]^2} \qquad \textbf{Equation 2-14}$$

Step 4: For each record in the Discharged Patient database, calculate the similarity.

$$similarity(t_i, x_i) = 1 - \frac{dist(t_i, x_i)}{\max\_dist} \qquad \textbf{Equation 2-15}$$

The cases in the Discharged Patient database with the highest similarities are the closest matching cases.

Using a weighted algorithm that allows sources external to the algorithm to provide reliable weights for the input variables creates a collaborative or composite approach. "The key to the composite approach lies in the determination of the most effective set of weights to use ..." [15] A second data mining method that defines the relative importance and therefore an effective set of weights is often used.

The k-nearest neighbours (k-NN) algorithm is used for a number of functions within the Medical Information-technology research Group (MIRG):

- Match closest cases for inspection by a clinician. [30, 37]
- Impute missing variable values into patient cases. [18, 20, 25]
- Extend datasets to include missing minimum data set variables when merging databases to create a multi-source test set. [6]
- Predict outcomes.

2.4.1 Match Closest Cases for Inspection

Matching closest cases matches an entire patient case, displaying the ten closest-matching cases for clinicians to inspect fully. No specific conclusions are drawn from the resulting cases by the application. All conclusions are left to the user. In certain situations, a warning is triggered if any of the matched cases have an adverse condition. i.e. if any of the matched patients have died a warning would trigger as a possible result in the current case. [37]

The relevant inputs and their corresponding weights to be used in the k-NN have been defined with the assistance of a domain expert. As there is no specific outcome to train an ANN toward and thereby create a minimum dataset with relative weights, the domain expert's knowledge is essential.

2.4.2 Impute Missing Values

To impute missing values into a database the closest Discharged Patient (complete) cases are matched to the current Open Patient case (case with missing values). The mean of the fields with missing values is calculated for the 10 closest-matching cases. The mean is placed into the target case, creating a complete case. This process of match-average-impute can be performed for two tasks: vertical expansion and horizontal expansion. Vertical expansion matches on a series of inputs and imputes values into those inputs creating an increased number of complete cases. Horizontal expansion matches on a series of inputs and imputes into those and other inputs creating complete cases with a greater number of variables. [20, 18]

As with matching closest case for inspection, the relevant inputs and their corresponding weights to be used in the k-NN for imputation of missing values have been defined with the assistance of a domain expert.

2.4.3 Extend Data Sets

It has been hypothesized by a member of MIRG that the k-NN could also be used to extend a data set. [6] Often databases originating from multiple sources or facilities do not always have the same variables available. A missing variable could be added to a database from a second database using the impute missing value horizontal expansion described in the previous section.

The weights and input variables would need to be set by defining a minimum dataset with the database containing all the relevant variables and then using those weight and variables to extend the second dataset which is missing a minimum dataset variable.

2.4.4 Predict Outcomes

As a possible solution to the artificial neural networks (ANNs) inability to deal well with continuous outcomes, the k-nearest neighbours (k-NN) algorithm can also be used to predict outcomes. When presented with a complete set of required inputs a case, an outcome can be predicted by calculating the mean of the desired outcome from the ten closest matching complete cases. To predict a binary outcome, an odd number of cases

should be taken and a majority rules or voting system employed. The inputs used and their relative weights must be predefined for each outcome. MIRG uses the weights from the ANN prediction model to define the k-NN weights [18]. See section 3.7 for weight transformation from ANN to k-NN format.

2.5 Collaborative Data Mining Methods

There exist different data mining methods, each with strengths and weaknesses. Many times combining methods overcomes the weaknesses, resulting in a stronger classification ability. Collaborations are arranged either in parallel or in series. Classification methods that are used in parallel operate side by side and have their results combined to form an overall result calculated by voting or averaging. Classification methods used in series operate one after another with results or parameters from one being fed into the next and final results emerging from the final member of the group.

2.5.1 Parallel Collaborations: Committee of Classifiers and Ensembles

A committee of classifiers uses a majority rules, also called voting, system to help predict outcomes more accurately than a single prediction system. [85] A committee can be homogeneous (also called an ensemble) or heterogeneous. An ensemble contains predictors all of the same data mining type (e.g. a set of ANNs). A heterogeneous committee contains predictors of different varieties (e.g. ANNs, decision trees, and case-based reasoners). Members of a heterogeneous committee are trained with different

methods and can also be trained with different data subsets. Ensembles are usually trained with different data subsets.

There are three main methods of dividing data for training with different data subsets: bagging, boosting and clustering. Bagging divides a training dataset into random subsets with or without resampling. [5] Four examples of Bagging are shown in Figure 2-9. Boosting trains each subsequent member of the committee with a random subset of the training set skewing the subsets to contain a larger percent of cases that the previous member classified incorrectly. [85] Boosting could be considered a combination of a parallel and a series system as the dataset used for each member of the group is defined with results from previous members (series) but the overall result is comprised of a combination of all members' results (parallel). Clustering divides data on specific feature information focusing on intra-cluster similarity and inter-cluster diversity. [45]

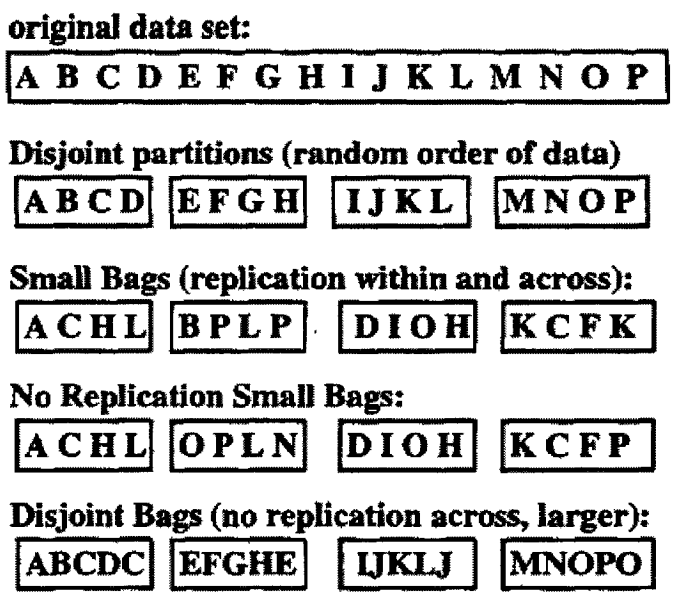

Figure 2-9: Four Approaches to a Bagging Committee of Classifiers. [13, Figure 1]

In heterogeneous committees, a small number of committee members is often used, such as five or three [42, 48]. The result found by each committee member is used in the majority rules system and the most popular result is returned. In the case of an ensemble, trained with different subsets of the original data and then brought together, the number of members in the committee is often quite high, 40 [13] and up to 100 [66] in others.

With a collaboration, it is important to ensure that each member has a high ability to classify correctly and that each member provides the ability to predict a different feature or operates a different local minima. Taking the average results of a set of diverse members will account for many local minima. [66]

2.5.2 Series Collaboration: Hybrid Data Mining Systems

Series collaborations require the output, parameters or other results from one member or data mining method to be used as parameters or inputs to another. The strengths of the members are intended to compensate for weaknesses in the remaining members. There have been a number of hybrid systems used in medical data mining. Some examples are:

- Used for medical diagnosis of breast cancer, diabetes and hepatitis, an ANN leading into a decision tree can create a system with strong generalizability from the ANN and strong comprehensibility from the rule induction of the decision tree. [85]
- Used for identification of electromyography (EMG - the study of electrical activity of muscle) signals, parametric pattern recognition algorithm statistical

method for feature definition and ANNs derived with a self-organizing map (SOM) for their ability to "1) exhibit adaptation or learning 2) pursue multiple hypothesis in parallel" [61].

- Used for predicting risk of osteoporosis prevalence, a decision tree and ANN collaboration was used. [76]
- A hybrid of an ANN for its ability to discover the relative importance of inputs into a data mining system and a case based reasoner (CBR) for its ability to work with missing values were used together in a process testing on neonatal intensive care data is presented in the next section.

2.5.2.1 Artificial Neural Network plus Case Based Reasoner (ANN+CBR) Hybrid: Imputation of Missing Values in Medical Databases

A series collaboration, or hybrid, of an artificial neural network (ANN) and an out-of-box case based reasoner (CBR)[b] and was proposed by Ennett and Frize to impute missing values into medical databases. [18, 20] The process requires a single layer ANN to be trained and tested on a complete cases dataset determining a minimum dataset. The weights from the single layer ANN inputs are then extracted and scaled to a format accepted by the CBR (whole numbers from 1 to 100). A database of cases with complete values in all required input variable fields is used as the 'match' database

[b] The Haley Corporation Easy Reasoner.

(Discharged Patients). A current case (Open Patient) with missing values in some input variables is matched against the 'match' database with the CBR.

The CBR uses a weighted matching algorithm to match closest cases based on the relative weights taken from the ANN. The 10 closest matching cases are taken and for any missing value in the current case, the mean of the 10 closest cases' value is calculated and imputed into the current case. [18, 20] The inputs were defined by a domain expert, Dr. C. R. Walker, as the SNAPPE-II variables; training was done on the mortality output. The original dataset had approximately 5000 cases with complete values in all of the SNAPPE-II variables and the dataset was imputed to fill to approximately 20000 cases. [18]

3 Methodology

This chapter briefly describes clinical decision support systems (CDSSs) with an example of a CDSS currently being developed by the Medical Information-technology Research Group (MIRG). The conceptual design of this CDSS is used as the top node of a hierarchical structure that leads to the description of the outcome prediction model definition process (OPMDP). The limitations of the current artificial neural network (ANN) application used by MIRG to define prediction models and solutions to those limitations are presented. As part of these solutions, the design of an Artificial Neural Network Research Framework (ANN RFW) that replaces the existing ANN application is presented. A committee of classifiers and a case based reasoner (CBR) with a k-nearest neighbour (k-NN) matching algorithm are also presented. Finally, a method of extracting relative weights, or importance values, from ANNs is discussed.

The focus of much of MIRG's research is to define prediction models for clinical and administrative outcomes for medical databases. In some cases a list of indicators, or the most important input variables (minimum dataset) is desired. In other cases, the prediction model is intended for use in a clinical environment for case-by-case outcome prediction. The specific result depends upon the database and outcome for which the model is being defined. For example, a database for childhood injuries may be mined to define a minimum dataset and therefore a list of indicators for severe injuries involving all-terrain vehicles (ATVs) [27] or a specific prediction model or series of prediction

models may be used as part of a clinical decision support system [33] or other real-time prediction system.

An example of the clinical decision support system (CDSS) is MIRG's PAtient Decision Support (PADS) framework intended for use in the Neonatal Intensive Care Unit (NICU) of the Children's Hospital of Eastern Ontario (CHEO).

3.1 Clinical Decision Support System: PArent Decision Support (PADS)

One of MIRG's projects is to create a Clinical decision support system (CDSS) for use in the Children's Hospital of Eastern Ontario (CHEO) in the Neonatal Intensive Care Unit (NICU). The PArent Decision Support (PADS) framework has been designed to provide parents and clinicians an on-line environment to interact with one another with the goal of making joint and educated decisions on treatment of a hospitalized infant. [83] The use cases defining the PADS operation are shown in Figure 3-1. The decision support being provided by the outcome prediction in the PADS environment is to decide when to "...initiate, withhold, withdraw or terminate treatment." [84]. The PADS framework eventually will be connected through web services to clinical and administrative databases incorporated in the facility's Electronic Health Information System (HIS) [9], outcome prediction applications [33] and automatic alerts [10] to allow clinicians and guardians to access up-to-date information on both active and past patient cases.

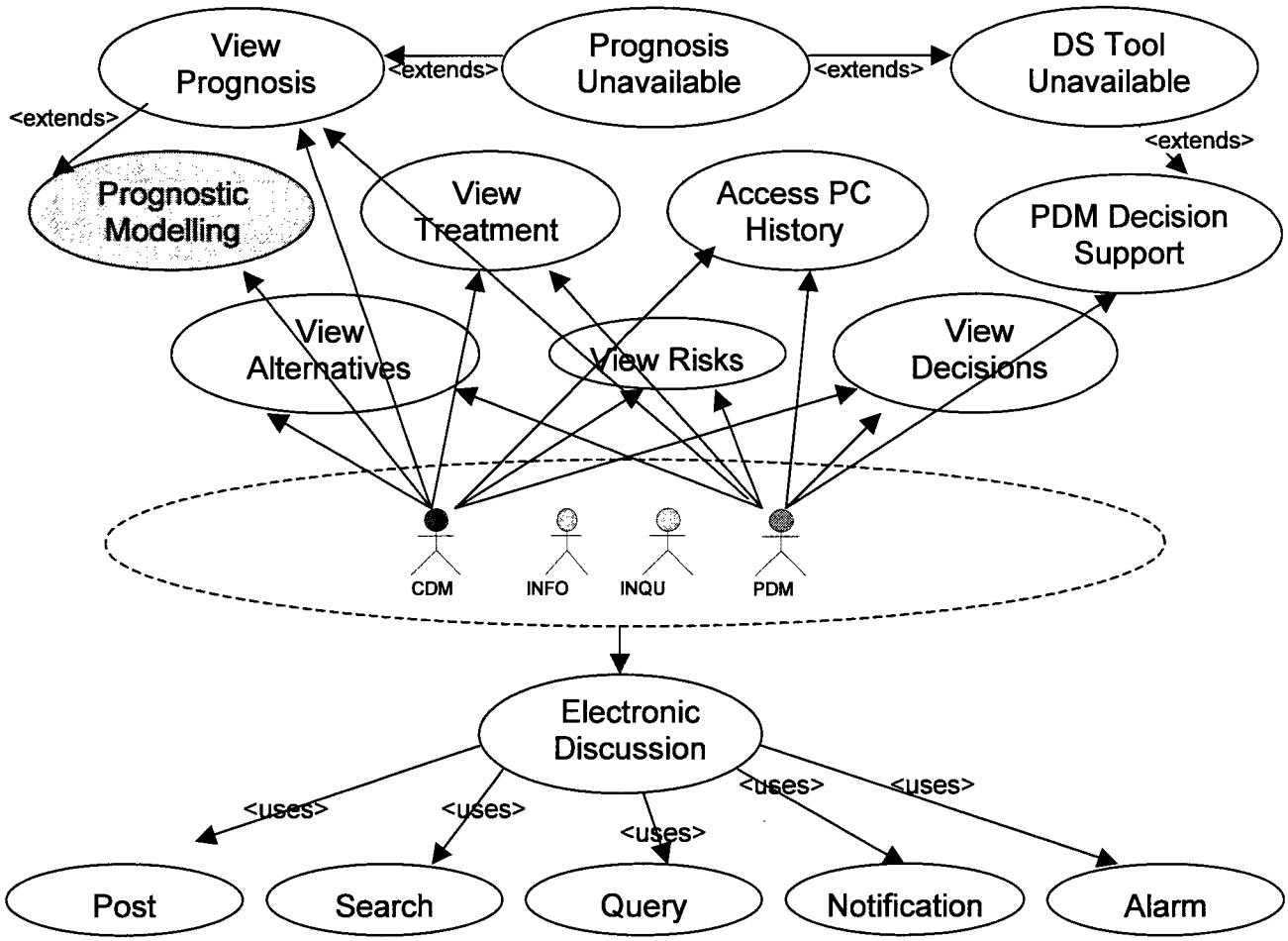

Figure 3-1: PADS: Use Cases. [83, Figure 5-A]
PC: Patient Care, DS: Decision Support, CDM: Clinical Decision Maker,
INFO: Informant, INQU: Inquirer, PDM: Patient Decision Maker.

The participants that will use the PADS framework are given in the List of Actors below.

List of Actors in Figure 3-1:

The actors in the Use Cases in Figure 3-1 are presented by Yang [83] as follows: "

1. Patient Decision-Maker (PDM): patient-parent (or the person acting in the role of parent on behalf of the family) who will be working with the physician to make a shared decision. There is a maximum of two PDM per patient-child.

2. Clinical Decision-Maker (CDM): the neonatologist, attending physician, or any other doctor who has the authority to make a diagnosis, prognosis, or a doctor's order. There is no limit on the number of CDM per patient-child .
3. Informant (INFO): any user who, or device that, participates in the care of the patient-child, or participates in servicing the patient-parent, or family members who/that may contribute to the knowledge pool. Examples of INFO are nurses, social workers, pharmacist, specialists, perinatologists, ethicist, and spiritual guidance.
4. Inquirer (INQU): any user who, or device that, may need to access information on the patient, or other information that will aid in the care of patient. " [83]

A number of use cases are presented in Figure 3-1, the one dealing with outcome prediction and therefore relevant for this work is **Prognostic Modeling**. An example of the display of patient diagnostic and estimated outcomes for neonatologist from the PADS tool is shown in Figure 3-2.

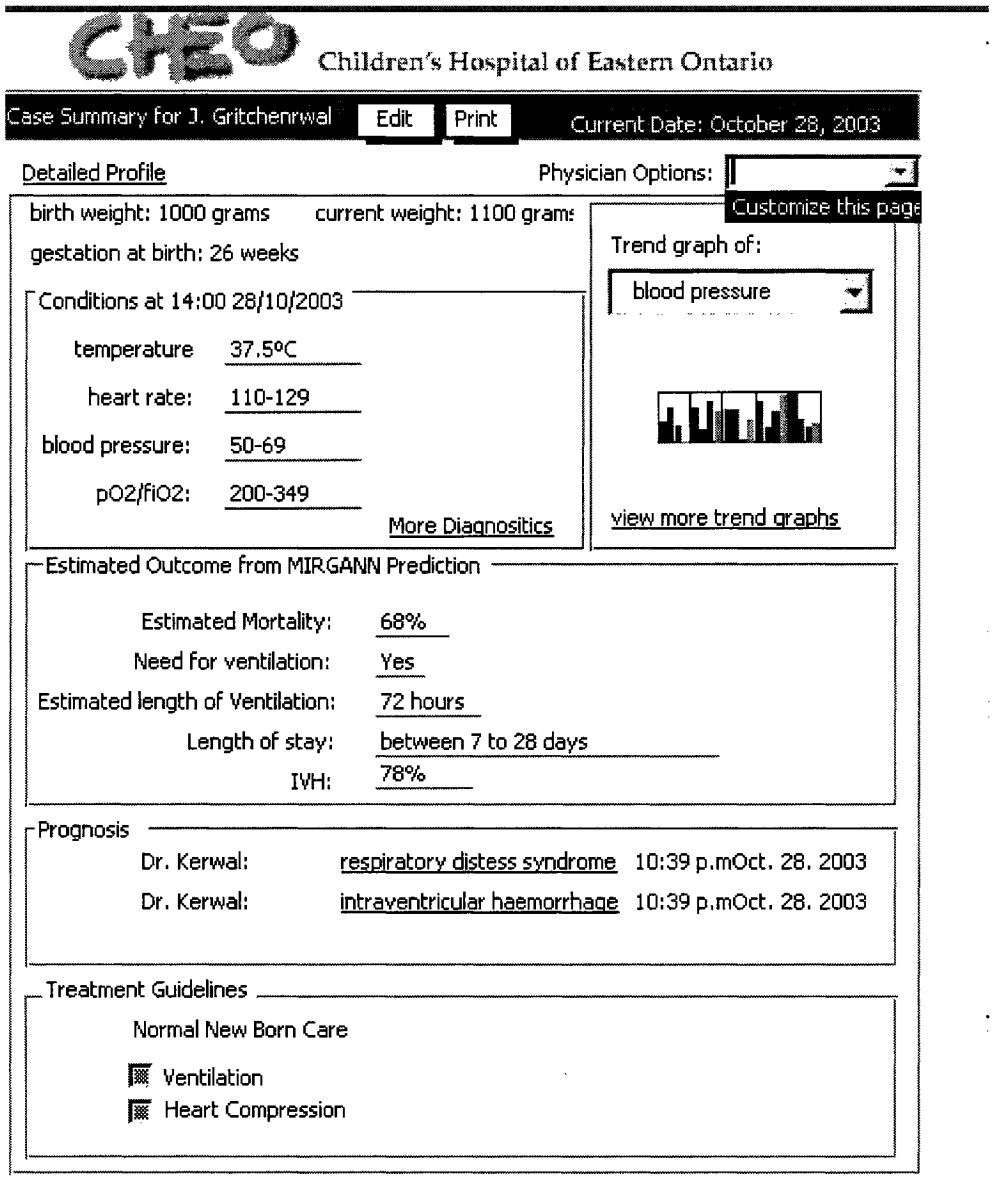

Figure 3-2: PADS: Example Clinician User Interface Screen Shot. [33 , Figure 7]

Outcomes can be in various forms:

- binary categorical (e.g. respiratory complications? yes or no),
- multiple categorical (e.g. 10 possible discrete values of an APGAR score),
- discrete continuous (e.g. blood pressure values that are all whole numbers over a continuous range) and

- real continuous (e.g. temperature in Fahrenheit is a decimal value bracketed only by physiological bounds).

For the purpose of this work: all outcomes that can be broken down into a single two-valued outcome are considered categorical (binary categorical), all outcomes that can *not* be broken down into a single two-valued outcome are considered continuous (multiple categorical, discrete continuous and real continuous).

The "Estimated Outcome from MIRGANN Prediction" box presents the prediction information. A method of predicting a binary categorical outcome (e.g. as shown by the 'Need for Ventilation' line in Figure 3-2) has already been developed by way of ANNs through MIRG. This work will concentrate on methods to more quickly and better define and deliver binary categorical outcomes as well as present methods for determining discrete continuous results (e.g. as shown by the Estimated length of Ventilation line in Figure 3-2) and range divided continuous results (e.g. as shown by the Length of Stay line in Figure 3-2).

The prediction ability will be available to the Clinical Decision Maker (CDM) with filtered results made available to the Patient Decision Maker (PDM). [83] Administrative outcomes such as hours or days on a ventilator, length of stay in the NICU or hours of nursing care, could be viewed by administrative members of staff, which are not included in the actors or use cases of the PADS design. The focus of this work will be on fulfilling some of the requirements for providing the **Prognostic**

Modeling portion to the PADS framework. The methods developed in this work for the categorical and continuous result prediction are transferable to other outcomes and other applications.

3.2 Clinical Decision Support System Conceptual Design

A lower level look at the general components required to support the **Prognostic Modeling** requirement of the CDSS of Figure 3-1and Figure 3-2 is shown in a UML 2.0 package diagram seen in Figure 3-3.

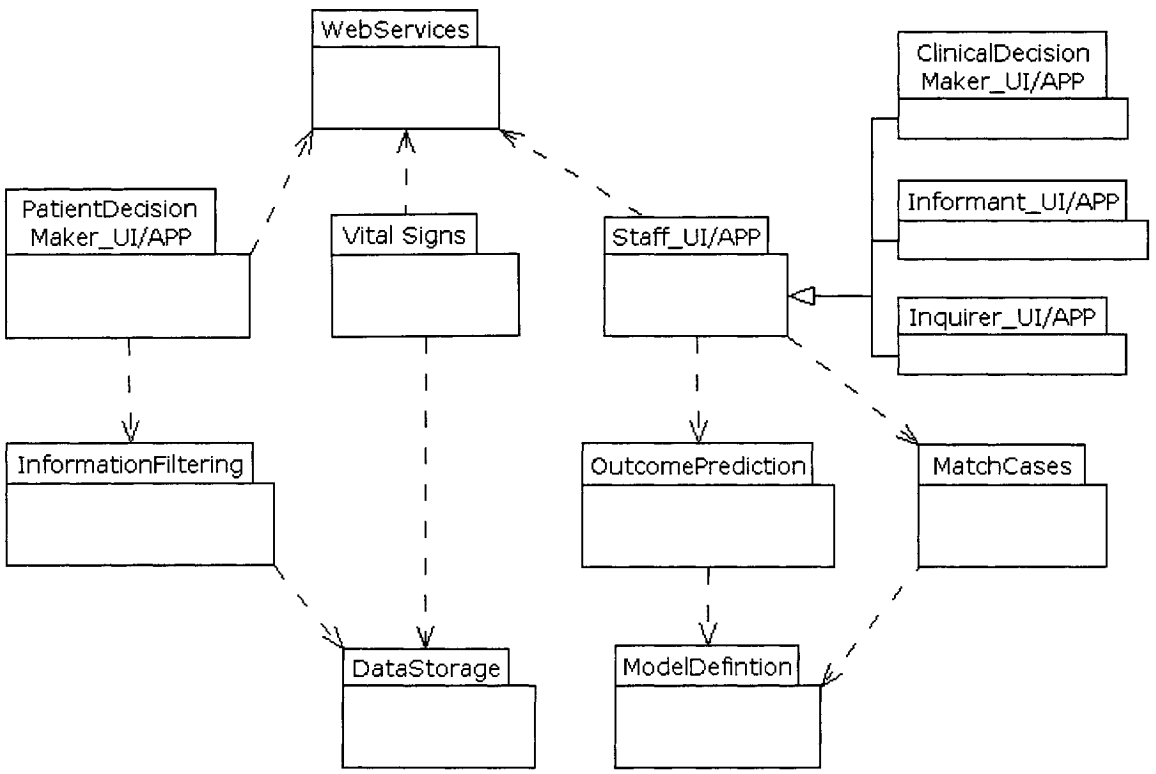

Figure 3-3: Package Diagram: MIRG CHEO NICU CDSS

Package Descriptions:

1. Access and Control

a. **Web Services**: Security and remote access control and delivery of all clinical decision support system (CDSS) applications and services: authentication, confidentiality, and non-repudiation.

2. External Real Time Interactions

   a. **Patient Decision Maker UI/APP**: User interface and associated applications to support patient decision maker (PDM) interaction through PADS.

   b. **Staff UI/APP**: User interface and applications to support hospital staff interactions through PADS.

   c. **Vital Signs**: Automatic retrieval and frequency of storage analysis of physiological parameters from clinical devices and other hospital information system participants.

3. Processing

   a. **Information Filtering**: Filtering of information provided to patient decision maker (PDM). Filtering driven by what information should ethically and legally be provided to PDM.

   b. **Outcome Prediction**: Prediction of clinical or administrative outcomes when requested by staff or delivered by automatic medical alerts. Artificial neural network (ANN) and k-nearest neighbour (k-NN) prediction operations.

   c. **Match Cases**: Match the target active case with closest matching patients in hospital database for evaluation by the clinical decision maker (CDM).

4. Base Level Information

a. **Data Storage**: Automatic data warehousing of all relevant information collected by hospital staff and on-line medical devices.

b. **Model Definition**: Definition of prediction models and all corresponding specifications.

To better show some of the interactions between the various components in the CDSS, a UML 2.0 Conceptual Class diagram is shown in Figure 3-4. Fundamental first pass requirement attributes and functions are shown for each class.

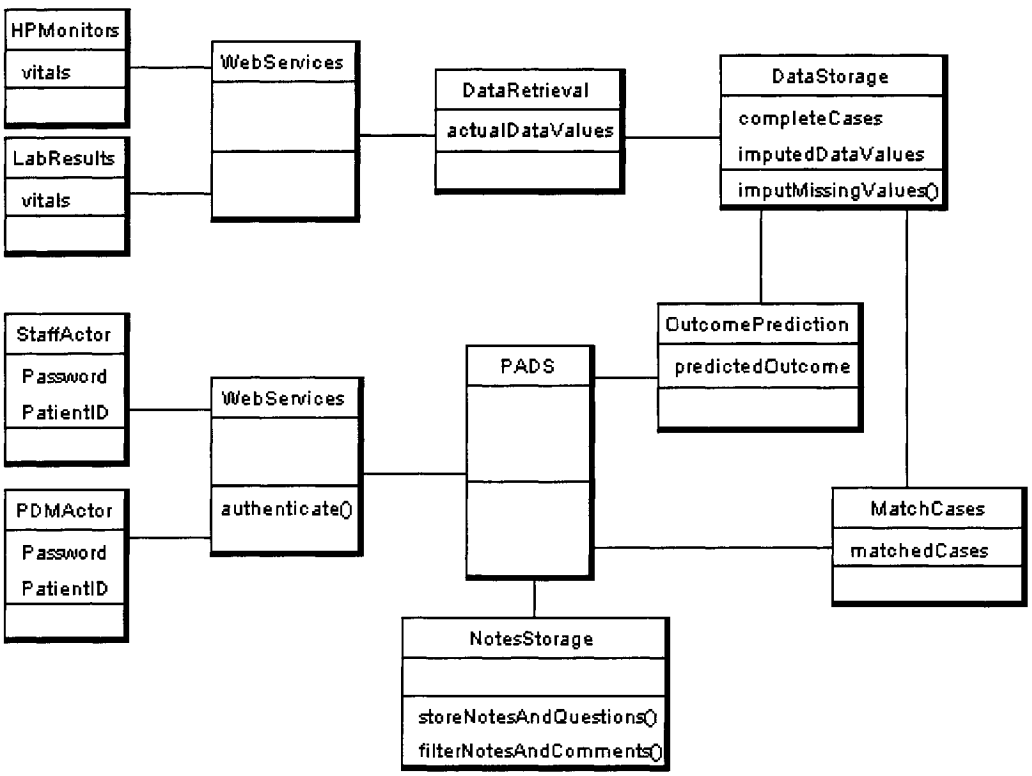

Figure 3-4: Conceptual Class Diagram: MIRG CHEO NICU CDSS.

The core of the work in this thesis is to assist in the creation of prediction models; therefore focus is placed on defining classes and activities for the **Model Definition** package of Figure 3-3 corresponding to the **Outcome Prediction** class box in Figure 3-4.

3.2.1 Model Definition Package Design

The **Model Definition** package of Figure 3-3 contains all of the researcher tools and processes that allow MIRG to create prediction models for clinical outcomes. The overall goal is simple – create an effective and robust prediction model. In the course of that greater goal, a number of smaller steps are taken. Figure 3.3 illustrates the use cases needed to complete these steps.

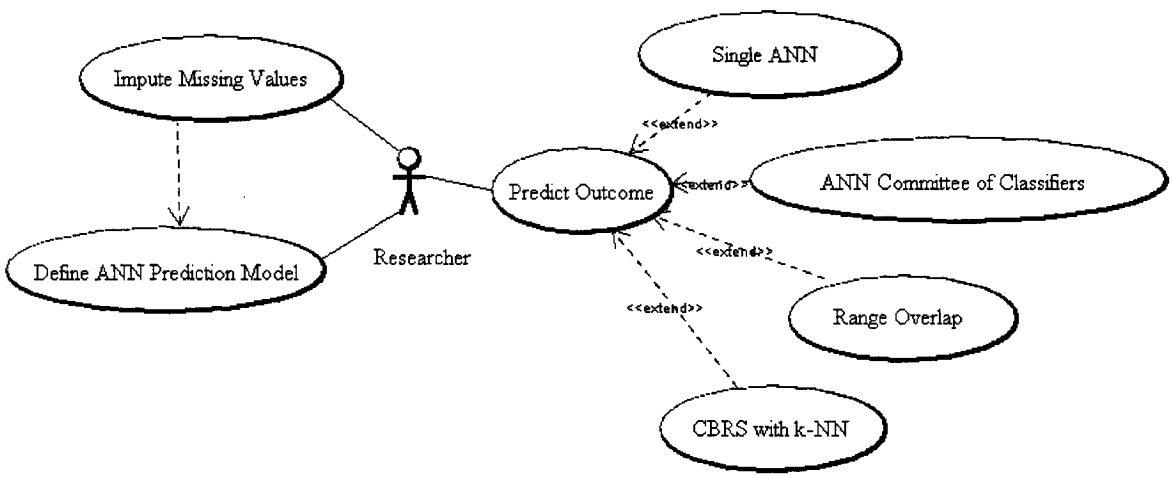

Figure 3-5: Researcher Use Cases: Prediction Model Definition package.

The Use Case diagram in Figure 3-5 has one actor (Researcher) and three main use cases.

Description of Actors:

1. **Researcher**: A member of MIRG working towards defining a prediction model.

Description of Use Cases:

1. **Impute Missing Data**

    a. Use a subset of data containing already complete cases.

    b. **Define ANN Prediction Model**

    c. Transform artificial neural network (ANN) connection weights to define the relative importance of each input node (variable). Transform the relative importance to a k-nearest neighbours (k-NN) format weight.

    d. Run Case Based Reasoner System(CBRS): Use the average calculated from the complete cases to impute values into the remainder of the original dataset that has missing values.

Prerequisite: Have a data set that contains some records with no missing values in the desired input variables and some records with missing values in the desired input variables.

2. **Define ANN Prediction Model**

    a. Set up and run Artificial Neural Network Research Framework (ANN RFW).

    b. Transform artificial neural network (ANN) connection weights to define the relative importance of each input node (variable).

    c. Remove lowest weighted input variable.

    d. Repeat a-c until degradation of results. The remaining input nodes define the minimum dataset.

Prerequisite: Dataset has only complete values in desired input variables (i.e. no missing values).

Alternative: Minimum dataset previously obtained and defined: Do not remove any input variables. Use the original set of input variables and run ANN RFW once.

3. **Predict Outcome**
    a. Attempt single artificial neural network (ANN) prediction.
    b. Attempt ANN committee of classifiers prediction.
    c. Attempt Case Based Reasoner System (CBRS) prediction.
    d. Attempt range overlap prediction from best of single or committee ANNs.
    e. Choose best appropriate (categorical or continuous) prediction method.

3.3 Outcome Prediction Model Definition Process (OPMDP)

Over the last few years, a number of applications have been upgraded or introduced to the Medical Information-technology Research Group (MIRG). Although group members had specialized knowledge of individual tools, there was a general lack of knowledge pertaining to combining these tools into a complete and understandable research cycle. Further complicating the situation, a number of these newly developed tools operate in a rotating sequence. To address these problems this work proposed the formalized Outcome Prediction Model Definition Process (OPMDP), the design of which is presented in UML 2.0 Activity diagrams, described in detail in the following section. The main process path for the Outcome Prediction Model Definition Process is shown in Figure 3-6. Any bubble in Figure 3-6 with a '*' preceding the label indicates a sub-activity whose activity diagram can be found in section Appendix C.

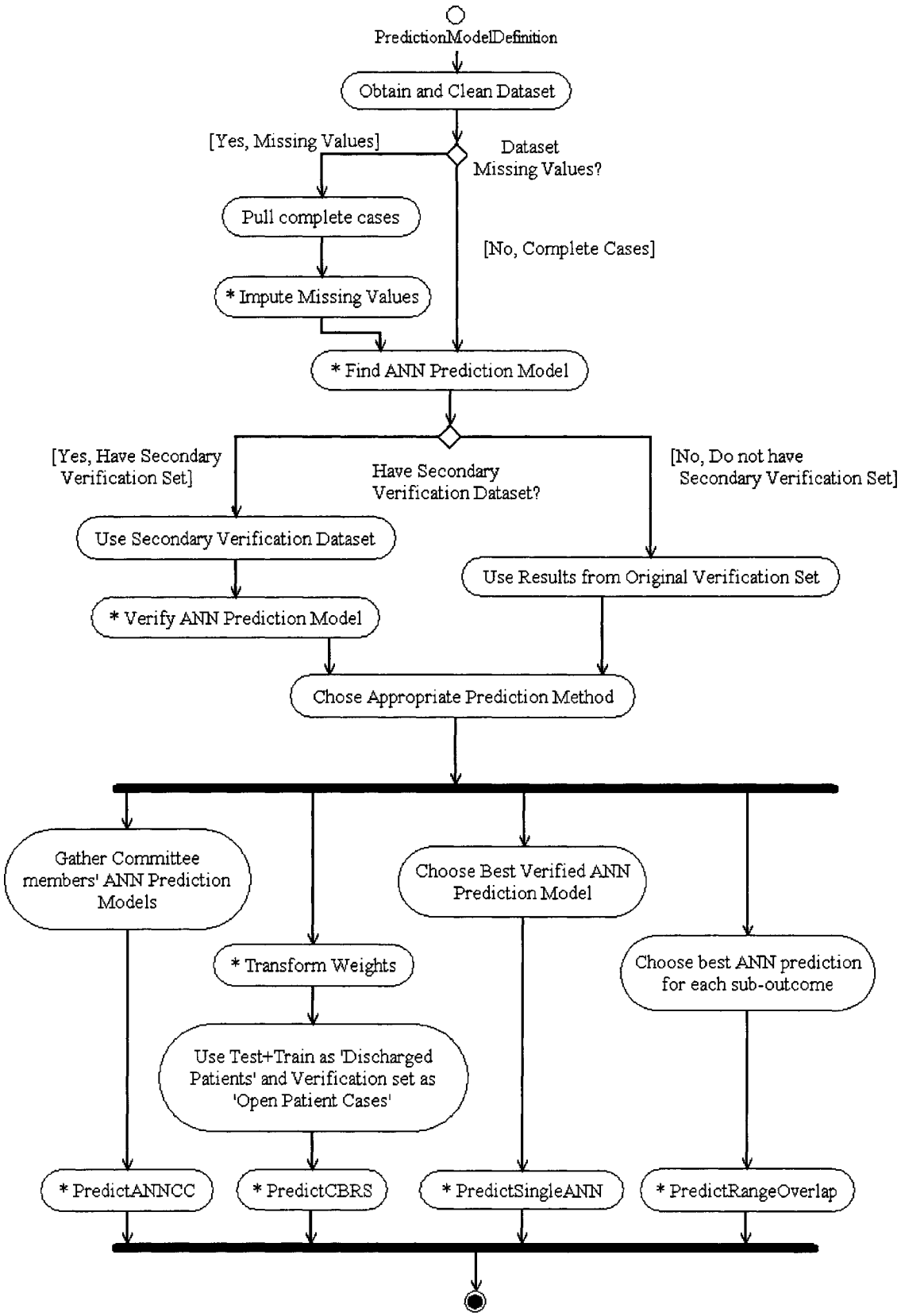

Figure 3-6: Activity Diagram: Outcome Prediction Model Definition Process.

The goal within MIRG is eventually to have all the elements of the process aligned together within a single software package. The functional integration presented in this thesis is a first step towards that goal. A conceptual class diagram for the complete research system is presented in UML 2.0 format shown in Figure 3-7. For design purposes, this will be shown as class diagrams although the system is currently not implemented as an object oriented software package.

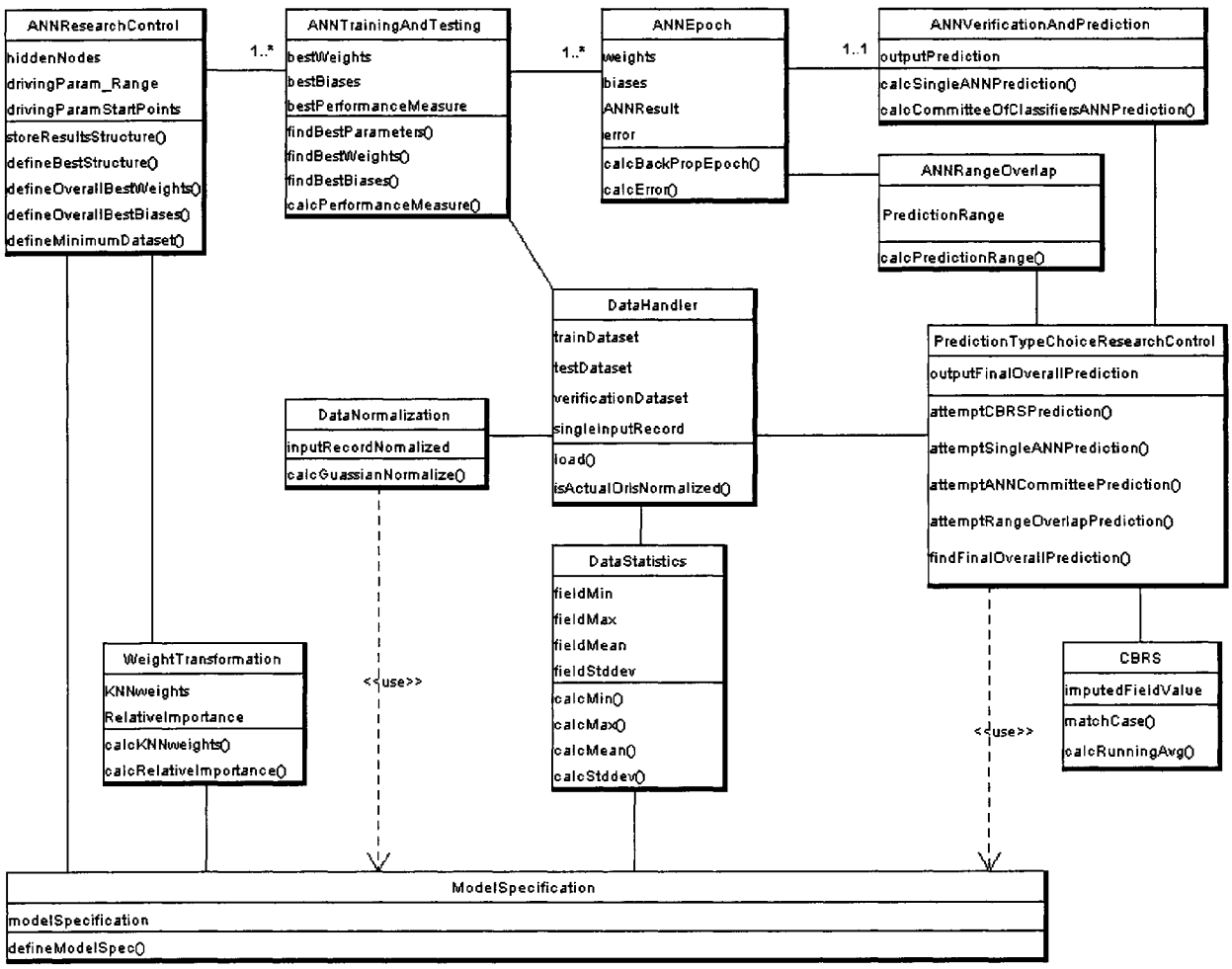

Figure 3-7: Conceptual Class Diagram: Outcome Prediction Model Definition.

The starting point for all prediction model defining in this system is training and testing an artificial neural network (ANN). The **ANNResearchControl** class found in the upper left-hand corner of the conceptual class diagram represents the functions presented in the next section. It is the research framework wrapped around the existing ANN application.

3.4 Artificial Neural Network Research Framework (ANN RFW) Design

In working towards the MIRG goal of creating prediction models for medical outcomes, a number of issues have been encountered. The first series of issues were encountered with limitations of the existing ANN application. The second series of issues grew from the introduction of new tools and associated new prediction theories and an associated need for the definition of research cycle process, or an Outcome Prediction Model Definition Process (OPMDP).

The existing ANN application used by MIRG is written in MatLab. The ANN core is a multilayer perceptron (MLP) using a feed forward back propagation learning algorithm with a weight elimination option. Nine parameters are used to drive the learning: learn rate, learn rate increment and decrement multiplication factors, momentum and error ratio for standard operation and the additional lambda weight decay factor, lambda increment and decrement multiplication factors and weight scale for weight elimination operation. The parameter values are selected using a method of sensitivity analysis where one parameter's values are changed using a divide and conquer algorithm while the other

eight are held constant. The nine-parameter arrangement with the best results for the given stopping criteria's performance measure is retained as the 'best'. [23]

The research framework does not change the core ANN learning. The ANN RFW is intended to provide greater flexibility and functionality to the research capabilities of the existing application.

3.4.1 Limitations of the Existing ANN Application

There were two main limitations with the existing ANN application. One limitation was operational and included slow speed, user intensive operation and limited functionality. The second limitation stemmed from an ANN's inherent inability to deal well with continuous outcomes.

3.4.2 Paring Down the Existing ANN Application

Solutions to the operational issues of the ANN application were implemented in MatLab by first paring down and then expanding the existing ANN application. The first problem to be solved was that of slow operating speed: working from the principle that no additions to the tool would be effective if the operational time could not be lessened. To combat the slow speed, the existing tool was refactored to remove unnecessary calculations and program paths, as well as removing the drawing of unused or redundant figures. When refactoring was completed, the tool ran in considerably less time (due mostly to less plotting). Although bulk metrics were not recorded at this point, it was observed that a number of experiments that took two to three days to complete with the

existing ANN application now were completed in one to two hours with the pared down version. A graph of the time taken to run experiments after the inclusion of the additions mentioned in the next sections is shown in Figure 3-9 on page 71.

After the paring down of the existing ANN, fewer output files were created, saving researcher analysis time as well as storage space. More concise and readable output files were also created. The reduced operational time taken made it possible to then add functionality and wrap a more extensive ANN research environment around the base tool. The ANN Research Framework (ANN RFW) is discussed in the next sections.

3.4.3 Increasing Functionality: Artificial Neural Network Research Framework (ANN RFW)

With the existing ANN application, only one ANN architecture or structure (number of hidden layers and hidden nodes) could be attempted at a time. The structures were selected at the discretion of the researcher. It was very rare for more than three different structures to be attempted for a specific outcome researched by an individual. This did not necessarily provide a high level of confidence in the correctness of the chosen number of hidden nodes.

During runs for that one structure, the researcher was required to extract the results for each of the nine driving parameters and choose the best parameter, manually transferring the best parameter to the next run. A verification dataset had not been implemented in the existing ANN application or the MIRG prediction model definition process to date.

In this work, additions have been made to the ANN application to deal with these three needed improvements, creating the ANN Research Framework (ANN RFW).

The first addition allows the software to automatically progress from one parameter run to the next, carrying forward the best parameter values. Researcher interaction is removed from the intermediate steps. See section 3.4.3.1.

The second addition allows the tool to attempt as many structures as is desired by the researcher, instead of only a single structure. The default is to loop through no hidden layers, then one hidden layer with one hidden node, progressing through to one hidden layer with $2n + 1$ hidden nodes where 'n' is the number of input variables. See section 3.4.3.2.

The third addition verifies the best performing model of each structure on data not trained or tested on. This is the only addition that requires an increase in the researcher's participation. The use of a verification data set was added, requiring the researcher to either split the original data set into three independent sections, instead of two as was done previously or, if a second database (i.e. one from a different facility or collection year) is available, the data can be set up with the testing and training cases from one database and the verification from another. The verification feature takes ten predefined verification sets and tests them on the single best epoch after all the training and testing is completed for that particular structure. See section 3.4.3.3.

A high level flow chart of the operation of the ANN RFW from the operational start point main_autoANN_Sequential.m is shown in Figure 3-8. Flow charts and brief descriptions for other important files can be found in section A.1 in the appendix.

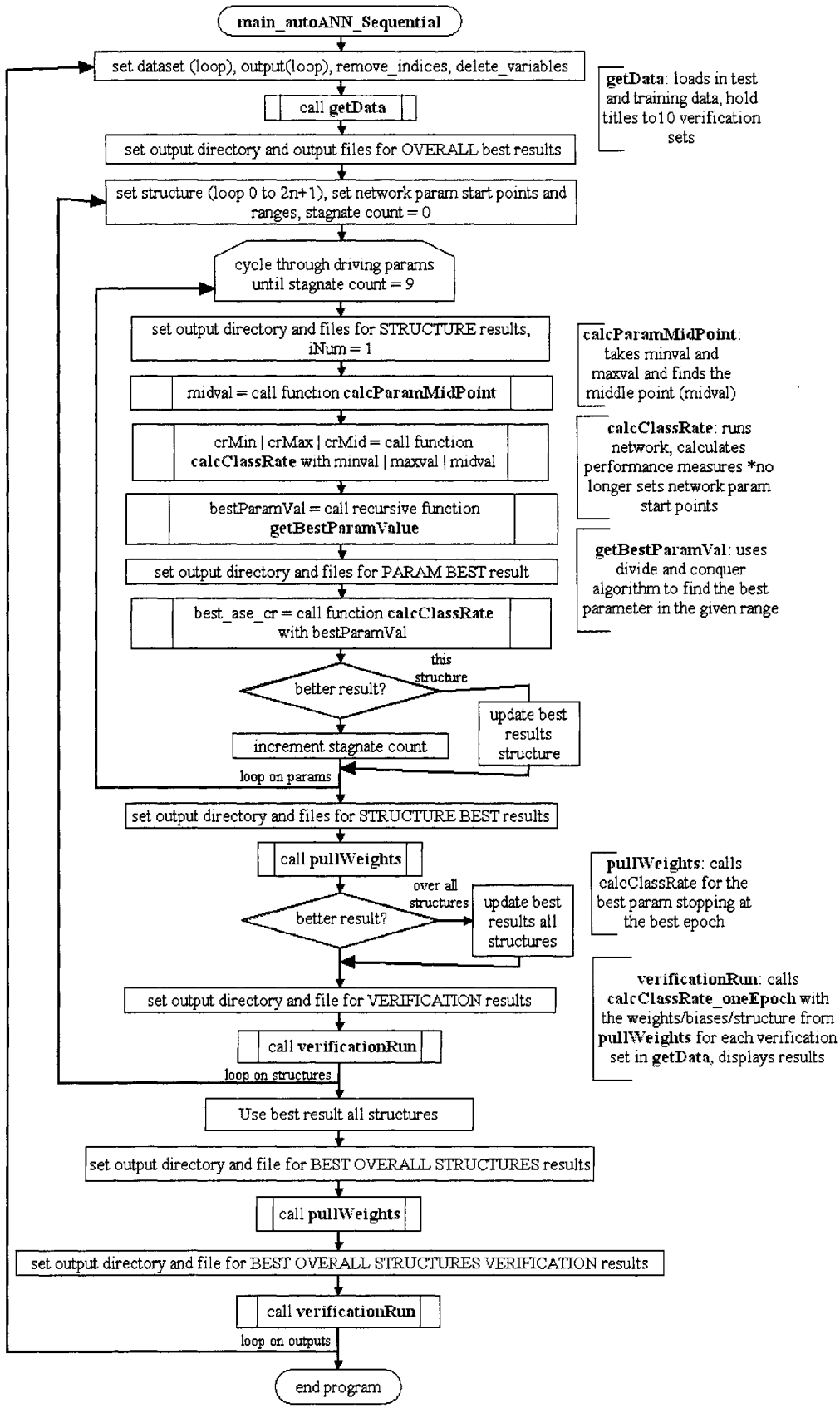

Figure 3-8: Flow Chart: ANN RFW: main_autoANN_Sequential.m.

3.4.3.1 Addition 1 - Automated Parameter Update

The multilayer perceptron artificial neural network (MLP ANN) trained with a back propagation algorithm implementing weight-elimination uses nine driving parameters to define how the weights and biases are changed from epoch to epoch as a function of the sum-squared error calculated over all trained cases. The values for these parameters, working together in different combinations create very different results in the learning of the ANN.

In the existing ANN research cycle, each of the nine driving parameters shown in Table 3-1 was given a default start point and an operating range in which to function. Various options within that operating range were attempted using a divide and conquer algorithm (see section A.3 of the appendix) to reach the value that yielded the best ANN results for a specific configuration of the other eight parameters. The newly found best result was compared to previous best results and carried forward to the next attempt using two different procedures: Best-in-Group Choice (see section 3.4.3.1.2.1) and Sequential Choice (see section 3.4.3.1.2.2).

3.4.3.1.1 Ranges and Start Points for each Parameter

The default parameters and the ranges for each parameter were defined based on values that have been found to be appropriate by the Medical Information-technology Research Group (MIRG) researchers on a number of different outcomes in various medical

databases. [18, 19, 27, 63] The ranges were then expanded to allow for previously unseen possibilities. The default parameter ranges and start point values set in the software can be found in Table 3-1. The researcher can alter the parameter range to encompass a broader or more restrictive series of options if desired. With the frequent updates to the default values made by the ANN RFW, the initial parameters are generally updated early in the iterations attempted and choosing good start points is not of as great concern as with the previous ANN tool.

Table 3-1: ANN RFW Default Parameter Range Minimum, Maximum and Start Points

Parameter	Start Point	Minimum	Maximum
Learn Rate	0.0005	0.00005	0.01
Learn Rate Increment	1.001	0.75	1.25
Learn Rate Decrement	0.999	0.75	1.25
Lambda	0.0001	0.00001	0.0001
Lambda Increment	1.001	0.75	1.25
Lambda Decrement	0.999	0.0001	1.25
Weight Scale	0.01	0.0001	0.999
Momentum	0.5	0.0001	0.999
Error Ratio	1.02	1.0001	1.5

3.4.3.1.2 Carry Forward: Best Values for Parameters

Two methods of choosing how to carry forward the best parameter are presented. The first method is a group choice, which sets the nine parameters' start point values and finds the best of each of the nine parameters individually, then updates using the best one of the group. The second is a subsequent choice where the starting points are set, the best is found for one parameter and, if the result improves on the previous best result, the start point is updated before the next parameter is attempted.

3.4.3.1.2.1 Best-in-Group Choice

In the Best-in-Group Choice method, a start point value is set for each of the nine parameters. The new best value for each of the nine parameters is found using the other eight start point values and a divide and conquer algorithm. The results of the nine trials are compared and the one with the best final results of the group is chosen. If the results of the best of the nine is better than the previous best (the first attempt will be the best by default), that one parameter of the nine had its start point value updated while the other eight do not change. Another set of nine attempts is then run with the eight old and one new parameter value.

The existing ANN tool provided the nine parameter trial results and MIRG researchers used the best-in-group method when parameters had to be transferred by hand from run to run. An experiment running from a single set of start points was performed on all nine parameters. The best result from that set of nine was chosen by hand and that single parameter was updated. A new experiment was then run with the eight old and one new parameter values.

To remove this manual step, a more user-friendly version of parameter updating was implemented in the ANN RFW, it is called the Subsequent Choice.

3.4.3.1.2.2 Subsequent Choice

In the Subsequent Choice method, the start points are set and a single parameter divide and conquer run is performed. If the results from that attempt are better than the previous

best, the parameter value is carried forward as the new start point for the next parameter's attempt. The best parameter is carried forward until a stagnant point is reached. A stagnant point occurs when the best result of the current parameter trial is not better than the previous best. The start point values are not updated at a stagnant point, as the result is not better than the previous best. After finding nine consecutive stagnant points (one attempt for each of the nine parameter's with the most recent update), the structure is complete and the next network structure is tried.

The parameters are ordered as follows: learn rate, learn rate increment factor, learn rate decrement factor, lambda, lambda increment factor, lambda decrement factor, weight scale, momentum and error ratio. The parameter trials begin with learn rate and progress through from one parameter to the next, wrapping around from error ratio back to learn rate, until nine consecutive stagnate points are found.

3.4.3.2 Addition 2 - Network Structure Trials

ANNs are often used to mine data when the relationships between the input variables are unknown. [59] The ANN operates as a black box that discovers the relationships by way of altering connection weights between nodes. An ANN with no hidden layer represents a linear relationship between the input nodes whereas using a hidden layer allows for a non-linear relationship.

Any number of hidden layers, each with any number of hidden nodes, can be used. The configuration that yields the best results for any given output and input combination is

not easily found. The ANN software used by MIRG has the option of single layer (no hidden layer) or double layer networks (one hidden layer). The question then becomes: Will one hidden layer or no hidden layer provide the best results? and then, How many hidden nodes should be used in the hidden layer?

Much work has been done to discover how many hidden nodes are required in the single hidden layer of a multilayer perceptron, ranging from the log of the number of training samples [75], to the square root of the number of input nodes [56]. Kolmogorov proposed a superposition theory that was later adapted to ANNs. Using the Kolmogorov superposition theory, the maximum number of hidden nodes necessary would be no larger than two times the number of inputs nodes plus one (2n+1) [4]. The default upper bound for the number of hidden nodes in one hidden layer of 2n+1 was used for this work.

In the ANN RFW, the default is to perform an exhaustive search to find the results for all network structures including single layer and spanning across double layer with one hidden node to double layer with 2n+1 hidden nodes, where n is the number of input nodes. The automated parameter update process described in section 3.4.3.1 is performed on each network structure. The researcher has the option of performing experiments on the full set of structure options or choosing any single structure or a sequential sub-set of structures. Over the course of the structure trials, the best tested structure is tracked and noted at the end of the results as the best structure for this input-output combination.

3.4.3.3 Addition 3 - Confirmation of the Results: The Verification Dataset

The ANN is trained with a combination of two datasets: the training and the test set. The resulting structure, weights and biases yielded by this training must be tested on a third set of data – the verification set. The verification set contains no cases that have been used to train the ANN. Good performance on a previously unseen dataset indicates the generalized nature of the ANN prediction model. If the results of training are good and the results of the verification sets are poor, the model has most likely over-trained and effectively memorized the cases in the training data. An inappropriately trained network will not be useful for predicting results for cases outside the scope of the training and test datasets.

The verification set is presented to the prediction model developed by the ANN – a single epoch using the weights, biases and network structure corresponding to the best point in the parameter trials. When choosing the prediction model to use beyond the research lab, it is important to choose one that verifies well on data that has not been previously presented to the ANN. The verification sets can be created using a secluded section of the original dataset or from a different source, such as a database taken from a different year or from a different facility than where the training and test set database was obtained.

3.4.4 ANN RFW Operational Solution Results

The ANN RFW provided solutions for the three operational limitations of the existing ANN which were: slow speed, user intensive operation and limited functionality. Operational speed metrics for the ANN RFW were taken from nine experiments. Eight of the experiments were run with nine input variables and approximately 20000 cases, each on a different outcome. A ninth was run with a dataset containing 30 input variables and approximately 5000 cases on a single outcome. A series of experiments with the existing ANN tool that would produce final results for a single structure would take four to six weeks to perform. As seen in Figure 3-9, the ANN RFW performed an experiment to the completion of a structure in less than two hours (for no hidden layer) to just under 18 hours (one hidden layer with 19 hidden nodes). That is a substantial reduction in operational time.

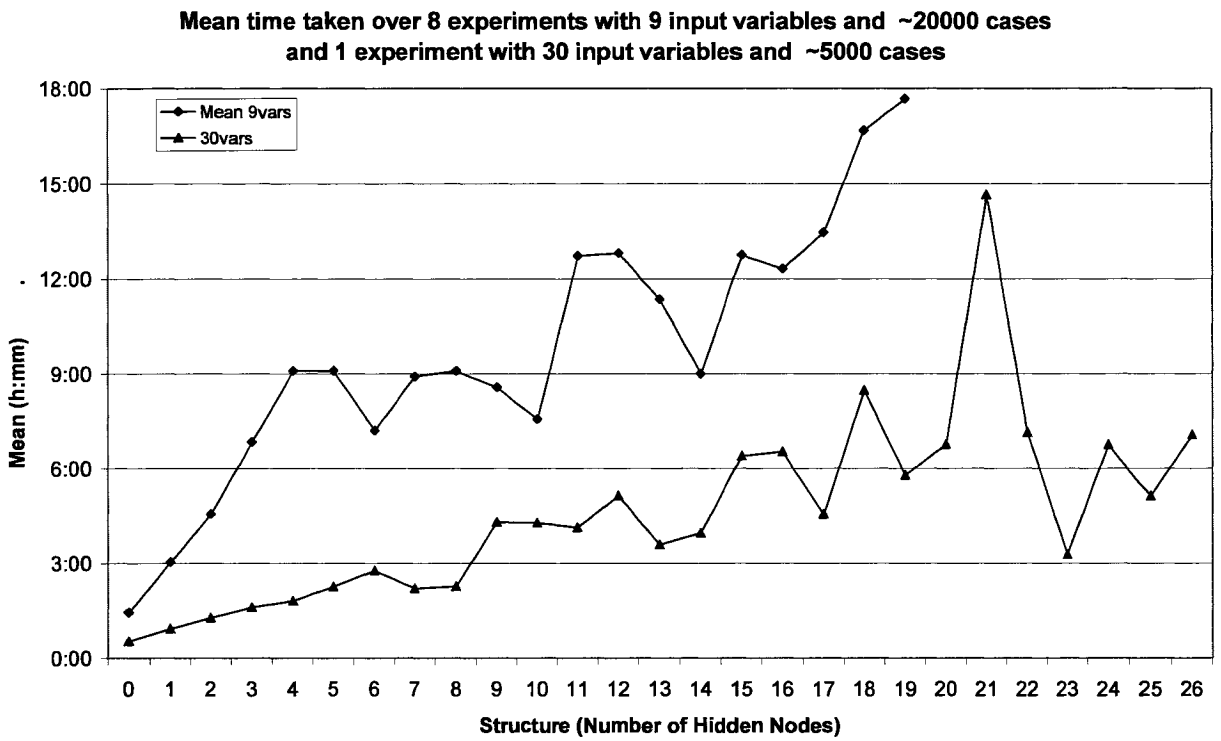

Figure 3-9: Operational speed of ANN RFW.

The addition of an automated parameter update drastically reduced the user participation in running experiments. The addition of sequential multiple structure runs as well as the introduction of the use of a verification dataset have increased the functionality of the ANN research application. To better define and predict ANN results for categorical outcomes and categorical sub-outcomes for continuous outcomes, a committee of classifiers application was introduced.

3.5 Committee of Classifiers Verification Tool (CCVT)

The purpose of the ANN Committee of Classifiers Verification Tool (CCVT) is two-fold. It is a simple application built in MatLab to assist researchers in verifying an ANN

prediction model as well as providing the ability to set-up a committee of classifiers of ANNs. The software can be set up with any number of weight, bias and structures combinations from the ANN RFW results. The tool calculates the predicted output of each case presented to it for each weight arrangement provided (each member of the committee). The overall result for each case is then calculated for a committee of classifiers.

Verification of a prediction model was not being done with the existing ANN tool. The Artificial Neural Network Research Framework (ANN RFW) verifies the model following the final training and testing cycle for each structure. Even with this addition, there still did not exist a method of verifying a model with another database after the research cycle had completed and the model had been defined. The Committee of Classifiers Verification Tool (CCVT) allows for any number of databases to be tested at anytime after the prediction model is defined.

The CCVT exhibits flexibility in the number of prediction models that can be verified at a time. A single prediction model can be set up or a number of different weight configurations for a single outcome representing a series of different prediction models can be verified. Along with this multi-model verification, a committee of classifiers can be defined.

The CCVT provides a plain text, comma delimited, case-by-case result file. The input variables, followed by the individual committee member results, followed by the final

committee result, are written to the output file for each case in the input file. Unlike the ANN RFW, which provides only final statistics for the dataset as a whole, the CCVT presents the output for each case. Knowing individual case results is important when clinical trials are performed. It is important to know which cases the prediction model predicted correctly or incorrectly when compared with which cases the physicians predicted correctly or incorrectly [63].

The input file must be in the same format as for the ANN RFW: a space or comma delimited text file with input fields, followed by output fields. The difference is that the CCVT is set up to accept either normalized or actual value inputs variables.

The Committee of Classifiers Verification Tool (CCVT) currently requires manual intervention. The ANN prediction model specifications required by the CCVT are:

- ANN weights and biases,
- input variables and their order to align with the weights and biases,
- input variable statistics (minimum, maximum, mean, and standard deviation of each input variable)
- ANN internal structure (number of hidden layers and nodes)

When discussing defining the ANN prediction model in this work, the ANN weights, ANN biases and ANN internal structure will be noted. It will be assumed that the input variables, input variable order and their statistics are also included. In the future, the ANN prediction model information will be automatically pulled from a model

specification database. For now, the normalization information for each field in the input set, the weights and biases for each prediction model, as well as how many prediction models will be tested, must be entered by the user.

3.6 Case Based Reasoning System (CBRS) using a K-Nearest Neighbours (k-NN) matching algorithm

The second limitation encountered with the artificial neural network (ANN) is inherent in the stochastic nature of the ANNs results. Many outcomes are continuous in nature (e.g. hours on a ventilator, hours of nursing care). The ANN's binary result causes problems when attempting to predict continuous outcomes. To address the continuous outcome problem, another type of prediction method was introduced that had not previously been used for specific outcome prediction by MIRG: a case based reasoner (CBR) in the form of a k-nearest neighbour (k-NN) matching algorithm.

A CBR had previously been used to match entire cases for inspection by physicians in an adult intensive care unit (ICU) and to generate warnings if any of the 10 closest matched case patients had died [37]. A CBR matching tool had also previously been used to match closest cases and impute missing values in the input fields of clinical databases [20]. It was also suggested that a database could be extended using the imputation process so that it could be successfully merged with another database that contained different input variables. [6] The attempt here will be to use the same sequence of steps used to impute missing input values to attempt to impute missing *outputs*. The k-NN's ability to deal with continuous outcomes by way of matching closest cases and then

imputing the missing value with the mean of that value from the closest matching cases makes it possible to predict continuous outcomes with more precision than is possible with ANN prediction which only predicts categorical outcomes.

The Case Based Reasoning System (CBRS) was written by two fourth year students of MIRG to match a single case to a database of existing cases using a weighted k-nearest neighbours (k-NN) algorithm while ignoring missing values. The first CBRS was written in Java using an ODBC (Open DataBase Connectivity) connection to link to a multi-table Access database. [16] The original purpose of the CBRS was to replace an out-of-box case based reasoner (Haley Corporation Easy Reasoner) that was integrated into a Visual Basic software package and used by Ennett as part of a hybrid process of an ANN and a CBR to impute missing input values into medical databases [18]. The CBRS was first written to fill a missing value with a random choice of either the minimum or maximum of the variable. [16] For this work, the CBRS was revised to match a series of cases in a database instead of matching only one at a time, as well as to impute the missing values with the mean of the closest matching cases values.

3.7 Transforming ANN Weights to k-NN format

With separate tools being part of a single system process, transfer of information from one to the other is inevitable. With the MIRG prediction model definition process, the links between software consist mostly of data files being converted from one format to another. Another information transfer pertains to the transformation of the connection weights from the ANN prediction model to a form useable by the k-NN algorithm.

To transform weights from the artificial neural network (ANN) format to the k-nearest neighbours (k-NN) format, a number of steps are involved. The first is to create a relative weight for each input from the array of connection weights provided by the ANN. The relative weights method described in section 2.3.4.1.2 can be used to create relative weights (importance) of each input. The k-NN accepts weights as whole numbers from 0 to 100 with 0 being an input that is to be ignored, such as an administrative field, and 100 being the most important input field. Linear scaling is used to transform the relative weights calculated from the ANN to the k-NN scaled weights.

A problem arises when a continuous outcome is encountered. The stochastic nature of ANN outputs requires that continuous outcomes be split into separate sub-outcomes. Each sub-outcome could have a different level of relative importance for each input variable. The question is then – How to define a single weight set from a series of weight sets?

For most continuous variable sub-outcomes, the weights should align. If the weights do not align, then the continuous outcome may need to be split into separate sections. For example, if the outcome hours on a ventilator is being examined sub-outcomes of greater than 0 hours, 4 hours, 8 hours, 12 hours and 24 hours may be used. If the relative importance of the variables is different for 0 and 4 hours than for the rest of the sub-outcomes, perhaps hours on a ventilator should be presented as two outcomes: a short-term ventilation of 4 hours or less and a long term ventilation of more than four hours.

Domain experts should be consulted when inconsistencies such as this arise as to an appropriate course of action and what would be a useful set of outcomes to display to the end-user.

When relative weights are found that align across a series of sub-outcomes the mean can be calculated for each input and then linearly scaled with the maximum to 100. The resulting weights are used for the continuous outcome. The final relative weights are used in the k-NN matching yielding the closest cases. The mean value of the desired outcome is calculated and used as the predicted outcome.

4 Experiment Construction for Verification of Tools and Processes

This chapter describes the databases and outcomes chosen to illustrate the effectiveness of the applications and processes described in this work. Three outcomes were chosen from a database that is familiar to the Medical Information-technology Research Group (MIRG): the Canadian Neonatal Network (CNN) database collected in 1996-97. As a secondary verification dataset, the Evidence-based Practice Identification & Change (EPIC) for 2002, an extension to the CNN database, was chosen. To illustrate a categorical result, Mortality of an infant (MORT) was chosen. To illustrate continuous results, two outcomes were chosen: length of stay (LOS) in the Neonatal Intensive Care Unit (NICU) and hours on a ventilator (VENT).

4.1 The Canadian Neonatal Network (CNN) and Evidence-based Practice Identification and Change (EPIC) Databases

Two databases were used in this work: The Canadian Neonatal Network (CNN) database and the CNN extension database Evidence-based Practice Identification and Change 2002 (EPIC). The CNN database is a compilation of data from 17 NICUs across Canada collected between January 8, 1996 and October 31, 1997. The CNN dataset contains data collected for 20488 admissions on day 1, 3, 14 and 28 of stay in one of the 17 NICUs that were participating in the study. [52, 65] The EPIC database contained 59 cases from premature infants resident at CHEO in 2002. [63]

Data collected for the Canadian Neonatal Network (CNN) database is highly regulated through the instructions provided by The Canadian Neonatal Network Abstractor's Manual [11]. The data is collected by a research assistant who enters all data on a daily basis directly from the Neonatal Intensive Care Unit (NICU) system. The direct entry on site removes the possibility of paper and keyboard transcription errors. All data is cleaned upon receipt at the coordinating centre i.e. removal of out-of-range entries. The validation cycle includes verifying a random selection of 5% of the cases; all centres participating have had excellent verification results. [34]

The CNN database has been used by MIRG to predict infant mortality, number of hours on a ventilator and length of stay in the NICU. The CNN database consists of many clinical and administrative fields. This work uses the Score for Neonatal Acute Physiology version 2 with Perinatal Extension (SNAPPE-II) prediction score variables. The nine SNAPPE-II variables are a subset of the Score for Neonatal Acute Physiology (SNAP) listed in Table D-1. SNAP values are collected in the first 12 hours of life. SNAPPE-II was devised by the CNN from the 37 SNAP variables using linear regression to predict infant mortality in the first 28 days of life. [64] The SNAPPE-II variables are listed in below.

- Lowest Blood Pressure in mmHg, (LBLOODP)
- Lowest Serum pH, (LSERUM)
- Lowest Urine in cc, (LURINE)
- Lowest Fraction of Inspired Oxygen, (PO2FIO2R)

- Lowest Temperature in Fahrenheit, (LTEMPF)
- Small for Gestational Age, (SGA)
- APGAR score at 5 minutes, (APGAR5),
- Birth Weight in grams, (BTHWT)
- Presence of Seizures, (SEIZURE)

The CNN dataset used in this work consisted of cases with complete values in all nine SNAPPE-II input variables as well as the outputs mortality, ventilation hours and length of stay. The final dataset consisted of 19377 cases. The original CNN dataset contained 5088 cases with complete values in the SNAPPE-II variables mentioned but was filled by Ennett. [18] The EPIC dataset was used as a verification dataset of 59 cases for infant mortality (MORT) and duration of ventilation (VENT) and 24 cases for length of stay (LOS). All input data used is from Day 1 of the infant's stay in the NICU.

A different set for training, testing and verification were created for each of the outcomes (MORT, VENT and LOS). Sub-outcomes were chosen for both of the continuous outcomes. VENT was split into VENTgt0 (0 hours), VENTgt1 (24 hours) and VENTgt2 (48 hours) with all ventilations being greater than the number in days. LOS was split into LOSgt7, LOSgt14, LOSgt21 and LOSgt28 with all stays being greater than the number in days.

The dataset was split separately for each of the three outcomes into a training, test and verification set. The verification set comprised 1/3 of the data (6459 cases). The

remaining 2/3 of the cases were split into 2/3 as the training set (8162 cases) 1/3 as the test set (4306 cases). Due to the high imbalance of the MORT outcome, resampling was performed.

Resampling is often performed on datasets with a high imbalance in the categorical outcome to artificially increase the prevalence of the less frequent result of the binary outcome. Some resampling methods require the removal of cases with the more frequent case until a desirable balance is reached. The danger with this method is that information can be lost with the random removal of cases.[19]

The method of re-sampling performed here was to choose cases with a positive outcome (e.g. patients who die) randomly and make copies of them within the dataset. The re-sampling creates multiple copies of cases with a positive result making it easier for the artificial neural network (ANN) to learn how to correctly predict the positive cases. For the mortality (MORT) outcome, cases with a positive outcome (e.g. death) were randomly resampled back into the data until a data split of 10% positive cases was reached. Upon further research, after the experiments were completed, it was discovered that an increased artificial set of 15% [19] or possibly as high as 20% [22, 21] would potentially lead to a higher effectiveness. The total number of cases in each dataset is shown in Table E-1, Table E-2 and Table E-3. These tables also show the A Priori statistics for each of the outcomes and sub-outcomes considered. A Priori statistics refer to the percent of positive and negative cases in the data set prior to data mining.

The verification dataset was broken into five sets of 1000 cases and five sets of 1500 cases. Each was built from a random selection of cases from the verification set with possible duplication of cases. Each of the 10 created verification datasets had the same proportion of positive to negative result cases as the original overall verification dataset. An 11^{th} verification set was created as a secondary verification set using the Evidence-based Practice Identification & Change (EPIC) database.

The verification datasets created from the Canadian Neonatal Network (CNN) database were used for the Artificial Neural Network Research Framework (ANN RFW). Experiments run to verify the ANN prediction model found with the ANN RFW using the CNN dataset on the EPIC dataset were done using the committee of classifiers and verification tool (CCVT). The EPIC dataset was also used in the Range Overlap and mean imputation Case Based Reasoner System (CBRS) continuous value predictions.

5 Results

This chapter summarizes the results from experiments run using the Canadian Neonatal Network (CNN) database to verify the Outcome Prediction Model Definition Process (OPMDP) defined in Chapter 3.2.1. The results from the proof-of-concept applications described in Section 3.4.3 (Artificial Neural Network Research Framework (ANN RFW)), Section 3.5 (Committee of Classifiers Verification Tool (CCVT)) and Section 3.6 (Case Based Reasoner System(CBRS)) are also presented in this chapter. The results of the relative weights calculations for each of the three chosen outcomes is also presented.

Results from the CBRS imputation using the mean of closest matching cases and range overlap solutions with the Evidence-based Practice Identification & Change (EPIC) dataset are then presented. Results from the EPIC dataset within this work are compared to results from work previously done by the Medical Information-Technology Research Group (MIRG) with the EPIC database using the nine SNAPPE-II variables as inputs and mortality (MORT), duration of ventilation (VENT) and length of stay (LOS) as outputs.

5.1 Artificial Neural Network Research Framework (ANN RFW) Results

The artificial neural network research frame work (ANN RFW) performed experiments for 20 structures on each of the eight outcomes. The final best results of all 20 structures as measured using the logarithmic sensitivity index as the performance measure are

presented in graph format. The results from the training, test and the average of the 10 verification datasets are shown. The five best structures are highlighted with enlarged shapes; triangles for training, circles for testing and squares for verification.

Results showing the training set performing best, then the test set performing slightly worse and the verification sets a little worse again is the expected result of an artificial neural network that has trained well but not over-trained. Provided the verification results are not substantially lower than the train and test, the progression of slightly worse for each curve is acceptable and shows a network that was not over-trained. If the verification set results are substantially lower than the training results, it is most likely indicative of an over-trained network. An over-trained network has poor generalizability and has effectively memorized the training (and possibly test set) and is not useful for predicting new cases.

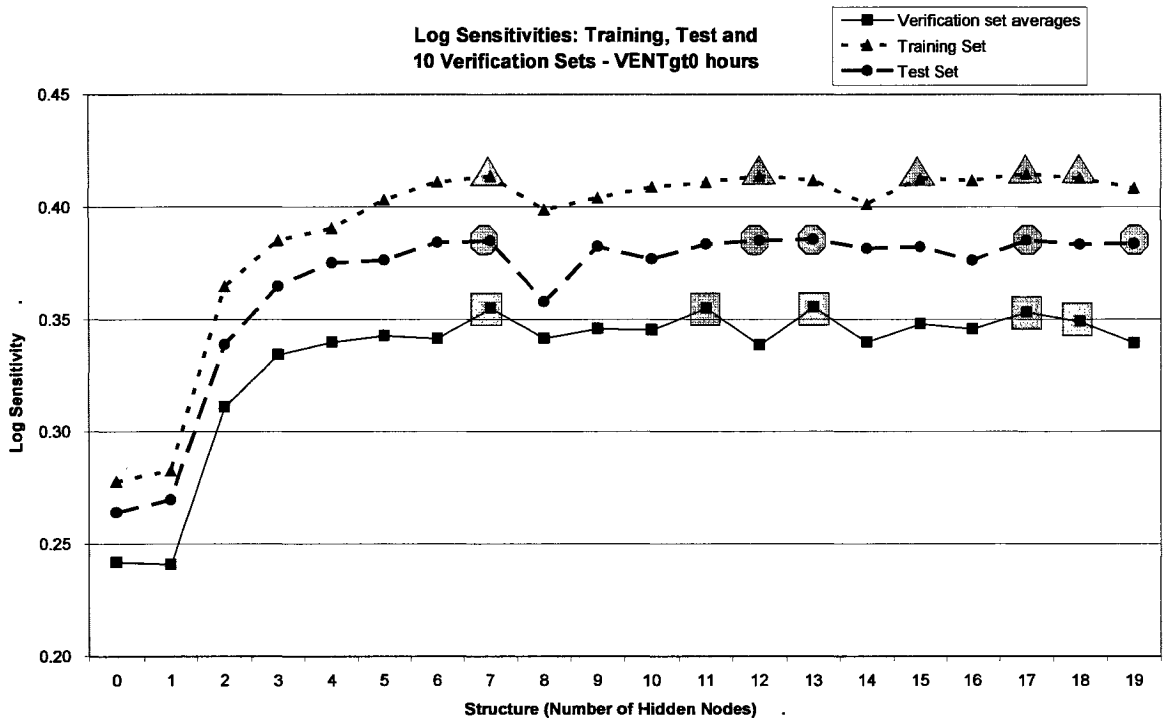

Figure 5-1: Log Sensitivities of 20 attempted structures for VENTgt0 hours (Ventilation Yes or No). Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

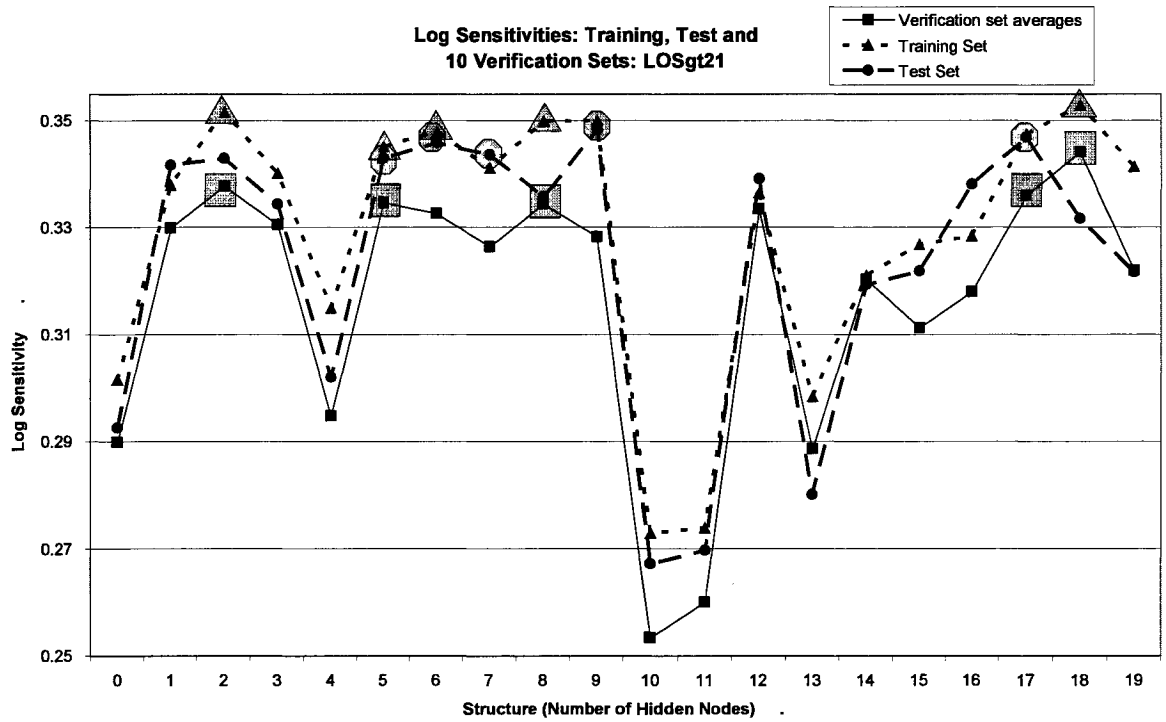

Figure 5-2: Log Sensitivities of 20 attempted structures for LOSgt21 days. Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

Figure 5-1 shows a result that is fairly consistent across all attempted structures. Figure 5-2 shows an example where the results change drastically as the number of hidden nodes change. The results of structure 12 in Figure 5-1 shows an example of possible overtraining, where the network trained and tested in the top five but verified much lower.

Because the relationship between the variables with respect to the outcome is unknown and therefore the complexity of the relationship is unknown, the number of hidden nodes required to appropriately define that relationship is best found by inspection of the *verified* results. The five best performing results were chosen to be carried to the next step in the process: the committee of classifiers.

Graphs of MORT, VENTgt1, VENTgt2, LOSgt7, LOSgt14 and LOSgt28 can be found in section F.1 of the appendix. Duration of ventilation outcomes of greater than 12 days (VENTgt12) and greater than 24 days (VENTgt24) were also prepared and used in the subsequent sections. VENTgt12 and VENTgt24 were resampled from 7% and 4% positives to 13% and 10% respectively. As with the MORT outcome, these should have been resampled to 20% to be effective [22, 21].

5.2 Committee of Classifiers Verification Tool (CCVT) Results: Verification of the Five Single Best and a Five Member Committee of Classifiers

The five best results found in the previous section were used to define a committee of classifiers for each of the outcomes and their sub-outcomes. The 10 verification sets created from the CNN database were used to test the committee concept. The average of the 10 verification sets' results, as well as the results from the training and testing sets (representing the training and testing log sensitivity of the prediction model) for a successful and an unsuccessful committee are presented.

The expected results for a committee of classifiers, where the committee performs as good as or better than the individual members, is illustrated by the VENTgt1 outcome shown in Figure 5-3. An example of a result where the committee as a whole did not perform better than each of the committee members is illustrated by the VENTgt24 outcome shown in Figure 5-4.

MORT, VENTgt0, VENTgt1, VENTgt2, LOSgt7, LOSgt14 had committees that performed better than their individual members. VENTgt12, VENTgt24, LOSgt21 and LOSgt28 days did not have a committee result that performed as good as or better than each of the committee members. It is interesting to note that the short term outcomes all performed well in a committee with Day 1 data while, the long term outcomes returned better results in individual ANNs.

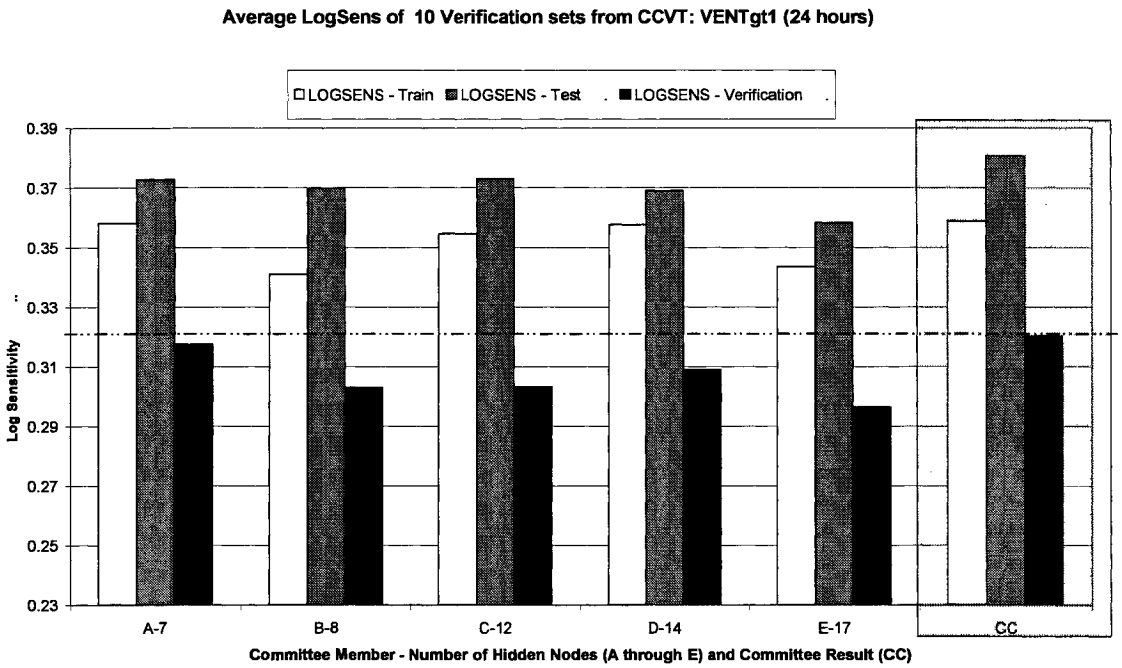

Figure 5-3: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for VENTgt1 (24 hours).

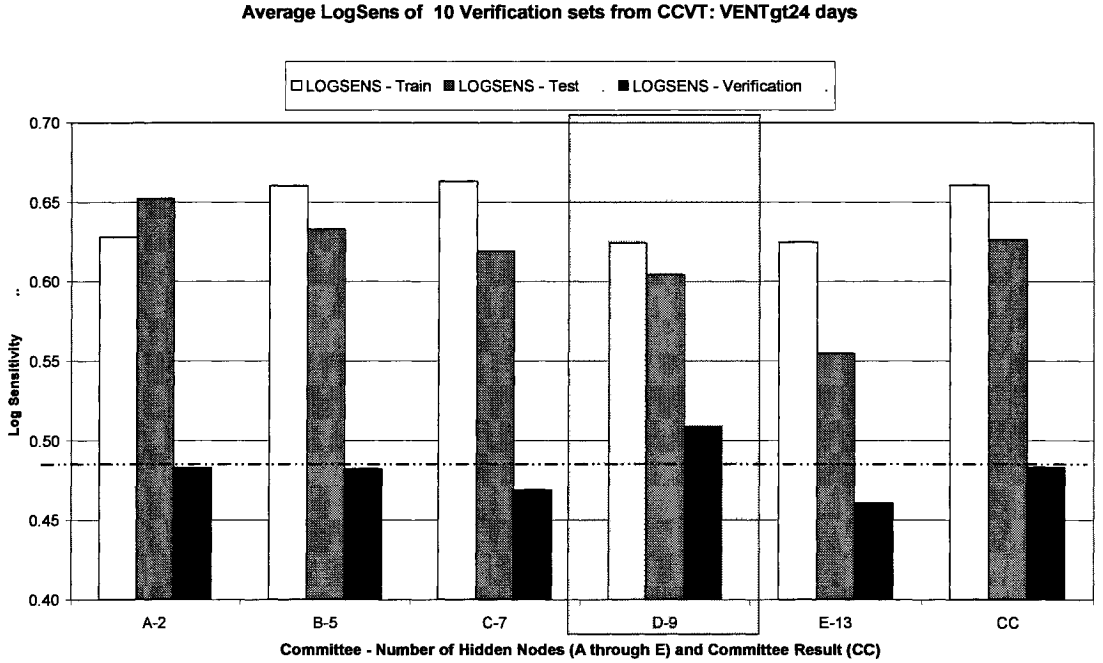

Figure 5-4: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for VENTgt24 days.

For the remainder of this work the best verified result will be used for each outcome. In the outcomes that the committee performed best, the committee results will be calculated. In outcomes that had a single ANN perform best, results from that single ANN will be calculated.

Graphs of MORT, VENTgt0, VENTgt2, VENTgt12, LOSgt7, LOSgt14, LOSgt21 and LOSgt28 can be found in section F.2 of the appendix.

5.3 Relative Weight Calculation Results

When removing variables to define a minimum dataset and in order to define the most and least important variables when defining indicators, the relative importance of each input variable must be found. The case based reasoner system's (CBRS's) k-nearest neighbours (k-NN) matching algorithm also requires relative weights to be predefined for each input. The relative weight method described in section 2.3.4.1 was used to define the relative importance of the five best verified structure results for each outcome. The five relative values were then averaged for each input variable.

To create a single weight for each input for the continuous outcomes that had been divided into sub-outcomes, the average relative weights of each input for the five best structures were taken and then scaled from 0 to 100. The resulting averages for each sub-outcome for VENT can be found in Figure 5-5 and for LOS in Figure 5-6.

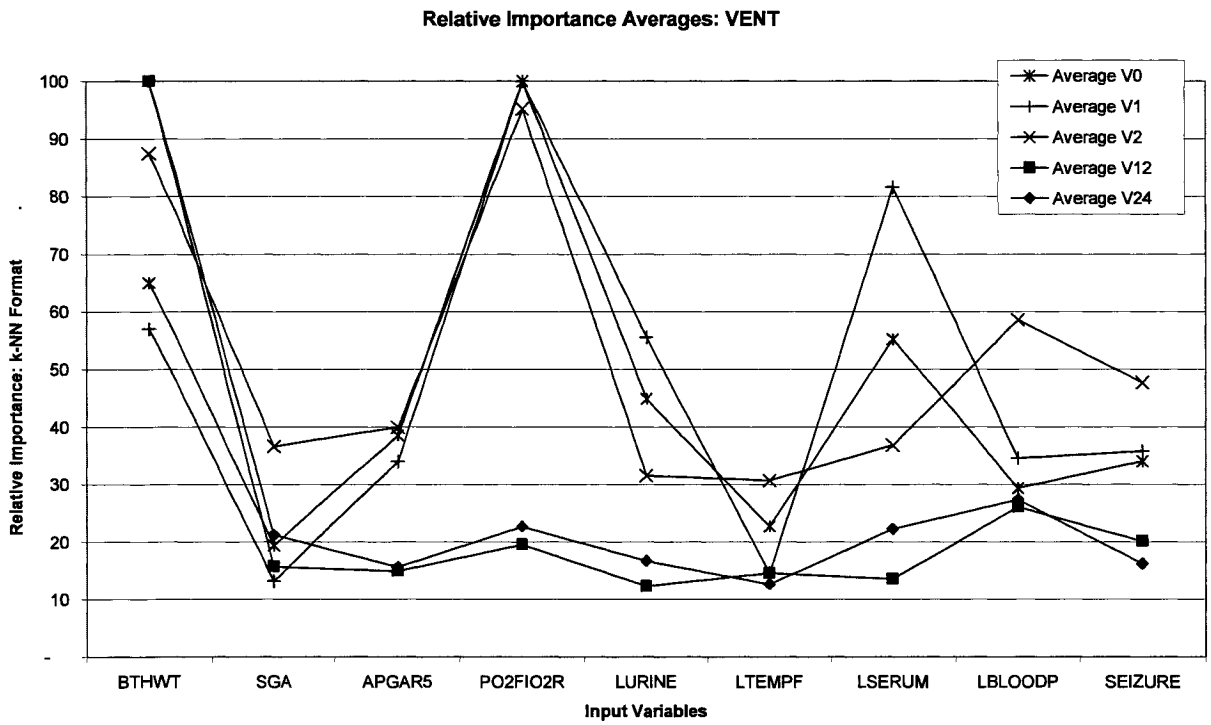

Figure 5-5: Relative Weights for VENT sub-outcomes.

The relative weights of the input variables for the VENT sub-outcomes change as the duration changes. There appears to be a division of VENT into short term (VENTgt0, 1 and 2 days) and long term durations (VENTgt12 and 24 days). Experiments of durations of ventilation between 2 days and 12 days should be conducted to see if there is a gradual trend from short to long term durations or if there is a sharp division. A sharp change in the weights would indicate a definite division of the VENT outcome into a short term and long term continuous sub-divisions.

The division of short term and long term ventilation durations mirrors what is seen in clinical practice. Infants entering the NICU are put on ventilation for a number of reasons but tend to stay on ventilation in only certain situations. [72] The short term

duration results indicate that the fraction of inspired oxygen (PO2FIO2R) along with birth weight (BTHWT) are the most import factors for placing an infant on ventilation but that the BTHWT is the most important factor in having an infant kept on artificial ventilation for an extended period of time. Once most specific duration divisions are found from data mining, investigations into the similarity or difference of the length of duration set by clinicians and as defined by data mining can be performed.

All data used in this work is from the first 12 hours after admission, or 'admission data'. The nearness of the data acquisition time to the short term duration values possibly indicates that the weights for the short term values may be the most relevant and should be used when predicting ventilation with Day 1 data. Work to discover if the relative weights of long term durations or, if found to exist, midrange durations should be performed using time varying data from days after Day 1 in the NICU.

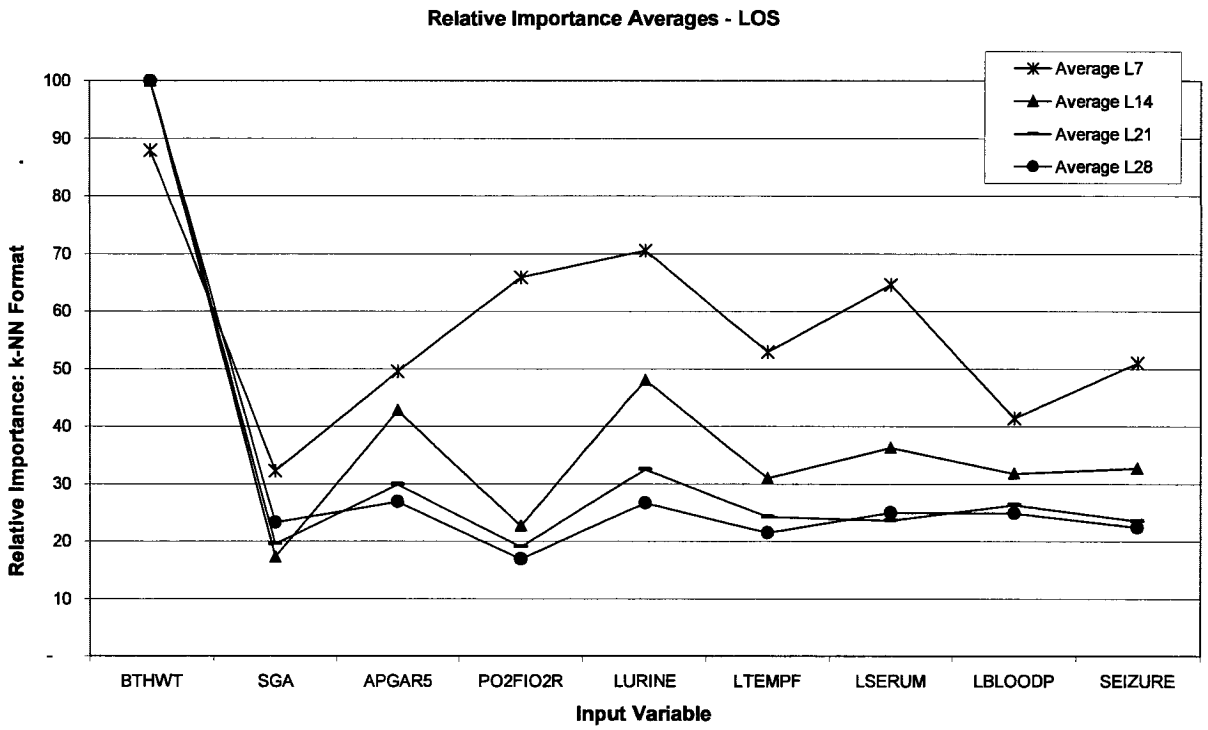

Figure 5-6: Relative Weights for LOS sub-outcomes.

The relative weights for the LOS outcome, as shown in Figure 5-6, change similarly as the days of stay progress. All inputs except BTHWT become less and less important as we attempt to predict a longer and longer stay in the NICU. It appears as though BTHWT is more important for predicting a longer term stay than it is for predicting a shorter term stay.

Rescaling the resulting overall average relative weight values from 0 to 100, the k-NN adjusted weights were found for MORT, VENT and LOS as seen in Figure 5-7 and Table 5-1. A second average of VENT using short term outcomes only (VENTgt0. VENTgt1 and VENTgt2: i.e. for 48 hours or less) was taken and is also presented in Figure 5-7 and Table 5-1.

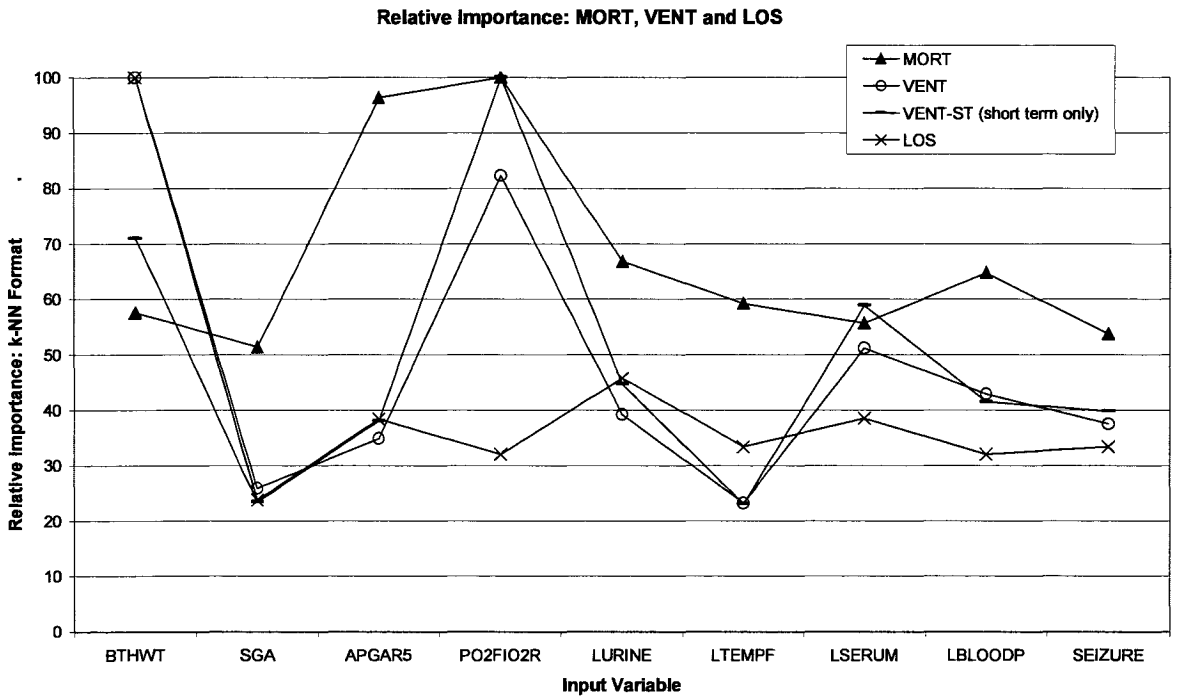

Figure 5-7: Relative Importance, k-NN Format Weights for MORT, VENT and LOS

As seen in Figure 5-7, the relative importance of the nine input variables over the three outcomes does not align. Most notably, the most important variable for the VENT and LOS outcomes, birth weight (BTHWT), is the least important for MORT, while the most important for MORT and one of the most important for VENT, fraction of inspired oxygen (PO2FIO2R), is one of the very least important for LOS.

Table 5-1: k-NN Adjusted Weights from MORT, VENT and LOS

	BTHWT	SGA	APGAR5	PO2FIO2R	LURINE	LTEMPF	LSERUM	LBLOODP	SEIZURE
MORT	58	51	96	100	67	59	56	65	54
VENT	100	26	35	82	39	23	51	43	38
VENT short term only	71	23	38	100	45	23	59	42	40
LOS	100	24	38	32	46	33	39	32	33

The difference in the importance of the variables leads to the conclusion that when working with a large dataset and developing a prediction model, a minimum dataset must be created for each individual outcome. Subsequently, prediction models should not be used to predict anything other than their primarily intended outcome. The difference also leads again to the question: How does one define the weights of a set of variables when matching entire patient cases for inspection and imputing missing values into a database?

For the remainder of this work, the VENT outcome will be approached using two relative weight configurations; the average of all five VENT sub-outcomes used in this work (VENT-ALL) and the average of the short term values only (VENT-ST).

5.4 Case Based Reasoning System (CBRS) Results

The Canadian Neonatal Network's Evidence-based Practice Identification & Change 2002 (EPIC) database was used to verify the Case Based Reasoning System (CBRS) mean imputation predictions. The weights displayed in Table 5-1 in the previous section were used for each of the outcomes. The VENT outcome was predicted using both the short-term duration (VENT-ST) and all-encompassing (VENT-ALL) sets of weights. The results for VENT-ST, VENT-ALL and LOS are shown in Table 5-2. There were 59 available cases for VENT and 24 for LOS. The difference between each imputed value and the actual value was calculated in number of weeks. The number of patient cases in the EPIC dataset that had an imputed value incorrect by a specific number of weeks is shown in Table 5-2.

Table 5-2: CBRS Results for VENT-ST, VENT-ALL and LOS.

	VENT - ST	VENT - ALL	LOS
Correct	4	7	0
Within a week	42	39	3
Between one and two weeks	10	10	3
Between two and three weeks	2	1	3
More than three weeks	1	2	15

Prediction of MORT with the CBRS was attempted as a categorical result. Recalling that to predict a categorical result by calculation of a mean requires an odd number of cases to ensure that no ties occur. The method is effectively a majority rules voting prediction. As the continuous results were calculated with 10 matched cases, the first attempt at predicting MORT was using 11 cases. A second attempt was performed using five cases. The five matched cases performed better than the 11 matched cases. Table 5-3 shows the results for the mean imputation majority rules prediction for MORT with 11 cases and with five cases. The true positives (TP), false negatives (FN), false positives (FP) and true negatives (TN) are shown for each. Both methods predicted the same number of negatives correctly (49) but the method using five closest cases predicted two positives correctly, while the method using 11 closest cases predicted only 1 out of the eight positives correctly.

Table 5-3: CBRS Mean Imputation Results for MORT.

	MORT11	MORT5
TP	1	2
FN	7	6
FP	2	2
TN	49	49

Overall, the mean imputation VENT results were mostly within a week. Given that this is an outcome that may very well be presented as a range, this is encouraging. LOS on

the other hand was most frequently at least a month wrong. The mortality results were also not particularly encouraging with only two out of the eight positive (death) cases predicted correctly.

5.5 Range Overlap Results

Range overlap calculations were performed on the EPIC database cases. The first set of results is from the calculations performed on ANN predictions done for each of the sub-outcomes. A second set of calculations were performed on the results from the CBRS imputations. The continuous results from the CBRS were categorized into the appropriate range and then compared with an overlap. The range overlap results for VENT and LOS outcomes calculated from the CCVT and CBRS results are shown in Table 5-4. As before, there were 59 cases used for VENT and 24 used for LOS.

Table 5-4: Range Overlap Method Results from CBRS VENT-ST, VENT-ALL and CCVT VENT.

	CBRS VENT - ST	CBRS VENT - ALL	CCVT VENT	CBRS LOS	CCVT LOS
Correct Range	29	30	24	5	10
One Range Over	13	12	23	8	8
Two Ranges Over	15	15	10	4	4
Three Ranges Over	2	2	2	7	2

The ranges for VENT were not well conceived for this particular type of prediction. The three bracketed ranges consisted of two that were one day long while, the last was 10 days long. The VENT outcome and the LOS outcome will most likely be presented in ranges of at least one week possibly longer. Combining those cases that predict in the correct range and one range over gives a good idea of what would be in a two week range

(from correct to 13 days wrong) for LOS. This further enumeration of results gives the correct classification rates shown in Table 5-5. Results were calculated from the results of two tools: (1) the CBRS imputed values that were then categorized into ranges and from (2) the CCVT results of either the single ANN or committee of classifier that was discovered to have performed the best on the 10 verification sets from the CNN dataset presented in section 5.2.

Table 5-5: Classification Rates for Range Overlap Method Results

	CBRS VENT - ST	CBRS VENT - ALL	CCVT VENT	CBRS LOS	CCVT LOS
CR	71%	71%	80%	54%	75%

VENT sub-outcomes must be created in a more useful and/or evenly divided fashion before any further conclusions on the benefit of this particular type of prediction presentation are made. The results from the committees and single ANN results (from the Committee of Classifiers Verification Tool (CCVT)) were better that the results from the imputation of an outcome results (from the Case Based Reasoner System(CBRS) mean imputation).

5.6 Comparison of Results with Previous EPIC dataset Predictions

The results in this section from this work were chosen from the results of the committee of classifiers define as better than all of the members in section 5.2 for MORT and VENTgt1 (24 hours) while the individual best performing committee member prediction model was used to classify the LOSgt28 outcome. The Canadian Neonatal Network

(CNN) database and the nine SNAPPE-II inputs have been used previously by the Medical Information-technology Research Group (MIRG) to predict MORT, VENTge24 and LOSgt28 by Qi [63]. The difference of the 'gt' (greater than) and 'ge' (greater than or equal) affects the manner in which the data was split for training and therefore the prediction model created.

Table 5-6: CCVT Results for EPIC MORT, VENTgt1 and LOSgt28

	CCVT MORT	CCVT VENTgt1 (24 hours)	CCVT LOSgt28
TN	48	7	1
FP	3	7	17
FN	3	8	1
TP	5	37	5

The results seen from the Committee of Classifiers Verification Tool (CCVT) for MORT are better than for VENT and LOS. The nine input variables used to train the ANNs were from the SNAPPE-II sub-set of the 37 variable SNAP. Recall that the SNAPPE-II variables were chosen from the SNAP as those best able to predict MORT so, the mort prediction being the best of the three stands to reason. A prediction model should be designed from the SNAP variable set for predicting VENT as well as LOS. Results from the CCVT and those taken from Qi [63] are shown below.

Table 5-7: CCVT Results and Previous Results from Qi. [63, Table 5-13, Table 5-14 and Table 5-14]

	Mortality	Mortality	Ventilation > 24 hours	Ventilation >= 24 hours	LOS > 28 days	LOS > 28 days
	CCVT	Qi [63]	CCVT	Qi [63]	CCVT	Qi [63]
Sensitivity	62.5%	25.0%	82.2%	40.0%	26.3%	63.2%
Specificity	94.1%	98.0%	50.0%	100.0%	9.1%	63.6%
CR	89.8%	88.1%	74.6%	54.2%	25.0%	63.3%
CP	86.4%	86.4%	76.3%	76.3%	75.0%	63.3%

The ventilation outcome was not set up in the same fashion between this work and Qi's work. The difference of categorizing ventilation = 24 hours in different categories did not affect the division of the EPIC dataset but it would have changed the data that was being trained and tested on the ANN.

The classification rate was higher for MORT and for VENT24 in this work than found by Qi. The classification rate found for the LOS outcome was considerably lower in this work than in Qi's. Predicting the LOS outcome with the EPIC dataset required removing a number of patient cases as the final discharge or death date was unknown. The 30 cases used for LOS by Qi were not identified and only 24 could be found for this work. It is not know if they are the same cases. The difference in cases could account for the difference in the LOS outcome result as this is not a large dataset where individual cases have little impact on overall results.

6 Discussion of Results

6.1 Conclusions

The design and implementation of the Artificial Neural Network Research Framework (ANN RFW) provided considerably faster operation than the existing ANN application. With the reduced speed and increased functionality that provided automated parameter updates and the performance of sequential experiments with increasing numbers of hidden nodes, a greater quantity of results were seen for an outcome.

The previously unseen quantity of results along with the use of a verification dataset extended confidence in the results. When an outcome is trained on two or three structures, it is impossible to tell if the numbers of hidden nodes used were the best options. The ANN RFW performing an experiment on an exclusive set of hidden nodes from no hidden layer to one hidden layer with $2n+1$ hidden nodes, where n is the number of input variables. A simple committee of classifiers was able to increase prediction ability for a number of outcomes and defining a better committee with diverse members should be explored.

The reduced operational time of the ANN RFW allows researchers to perform experiments on a larger quantity of outcomes. This has made it feasible to perform experiments on a larger number of sub-outcomes for a continuous outcome allowing for a more complete study of the outcome.

The introduction of relative weight equations made it possible to extract weights correctly from a three-layer ANN. The benefit of that is seen in that ANNs are used most often on problems that are non-linear in their solutions and therefore require a hidden layer and hidden nodes. A method of extracting relative importance from the three layer network gives some insight into the internal operations of the black-box.

The ability to extract weights from a three-layer ANN and the ability to perform experiments on an increased number of outcomes have made it possible to discovery the change in relative importance of input variables for continuous outcomes, such as duration of ventilation and length of stay in an NICU, as longer durations are predicted. The change in relative importance may indicate a division of an outcome into short term and long term.

The relative weights calculated for each of the three main outcomes were seen to be different and it was concluded that every outcome must have its own prediction model defined with its own input variables and corresponding weights. This was illustrated by the mortality outcome performing the best on input variables chosen for prediction with mortality over duration of ventilation and length of stay outcomes.

6.2 Contributions to Knowledge

1. Limitations of weight extraction and relative weight calculation models for multilayer perceptron artificial neural networks (MLP ANNs) found in the literature were

discovered and corrected: A modified relative weight calculation method was created to expand the functionality of an existing method to include the types of networks used frequently with medical data: non-linear MLP ANNs with one hidden layer where the hidden to output layer connection weights are non-similar.

2. Through investigations of the relative weights extracted using the modified model, three important points were demonstrated:
    a. Short-term, long-term and possibly mid-range-term divisions in a continuous outcome may be defined using the differences in relative weights of input variables of an ANN,
    b. Continuous outcomes may need to be treated as separate duration-based outcomes (short-term, long-term and any existing mid-range-terms) for prediction purposes due to different relative weights and different prediction models, and
    c. Different outcomes, beyond those sub-outcomes within a single continuous outcome, may have different input variable weights requiring that every outcome to be predicted be given its own prediction model.

3. A method of predicting continuous outcomes using an ANN plus CBR (artificial neural network plus a case based reasoner) collaboration in a similar fashion to imputing missing values in a database was attempted. The results were interesting to discover even though they were not yet deemed favourable.

4. Design and implementation of an Artificial Neural Network Research Framework (ANN RFW) that has improved automation and increased functionality led to the successful introduction of a committee of classifiers to increase the generalisation prediction ability of ANNs and a majority rules range overlap solution to broaden the outcome presentation ability of ANNs.

6.3 Future work

1 Clinical Outcome Prediction Research:

1.1 Train artificial neural networks (ANNs) with an ensemble method (bagging, boosting or clustering) to enable a committee of classifiers that have members of more diverse training origin subsets and therefore enable better final predictions.

1.2 Further investigation of the possible clinical significance of finding different weight values associated with the input variables for different sub-outcomes of the same continuous outcome: how to use these different relative weights to define a single set of relative weights for the continuous outcome or how to divide the continuous outcome.

1.3 Further investigation into the different weight values associated with continuous outcomes' sub-outcomes on time-varying data from real-time data acquisition.

1.4 Further investigation into the inputs and output(s) needed to define an ANN prediction model and therefore relative weights for both imputing missing values into a patient case and matching an entire patient case for inspection.

1.5 Creation of prediction models for duration of ventilation (VENT) and length of stay (LOS) from the full Score for Neonatal Acute Physiology (SNAP) variable set. Investigating if the new prediction models perform better than the mortality (MORT) based Score for Neonatal Acute Physiology version 2 with the Perinatal Extension (SNAPPE-II) variables.

2 Software development:

2.1 Refining and updating the Medical Information-technology Research Group (MIRG) Clinical decision support system (CDSS) for the Children's Hospital of Eastern Ontario's Neonatal Intensive Care Unit (CHEO's NICU):

2.1.1 Creation of a prediction model specification database.

2.1.2 Integration of prediction tools with prediction model specification database.

2.1.3 Integration of prediction tools with user interface and secure remote access control (PArent Decision Support (PADS) + web services).

2.2 Artificial Neural Network Research Framework (ANN RFW):

2.2.1 Expand the divide and conquer section to include a dynamic range for the parameters. If a parameter hits the top or bottom of the defined range then reset the range boundary. Design so that the function can be disabled if desired by the researcher.

2.2.2 Enable a user-defined number of verification sets to be pulled at random from a single verification data set reducing the user intensive data creation.

2.2.3 After all structures have been attempted, automatically choose the best one or the best set of *verified* structures, calculate relative weights and automatically remove least important input(s). Re-run using a single structure or set of structures until a minimum dataset has been reached.

2.3 CBRS with k-NN:

2.3.1 Attempt imputation calculations that are more aligned with data structure than an average. i.e. a value that is statistically matched with the skew or distribution of the data in question.

References

1.Aurelle D, Lek S, Giraudel JL, and Berrebi P. "Microsatellites and artificial neural networks: tools for the discrimination between natural an hatchery brown trout (Salmo trutta, L.) in Atlantic populations" Ecological Modeling (1999): 313-324.

2.Bach J. "A RFW for Good Enough Testing", Computer: Innovative technology for computer professionals Oct 1998: 124-126.

3.Back AD and Trappenberg TP. "Selecting Inputs for Modeling Using Normalized Higher Order Statistics and Independent Component Analysis" IEEE Transactions on Neural Networks May 2001: 612-17.

4.Beiu V. "A Novel Highly Reliable Low-Power Nano Architecture When von Neumann Augments Kologorov" Proc of the 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors (2004).

5.Bermak A, and Martinez D. "A Compact 3-D VLSI Classifier Using Bagging Threshold Network Ensembles" IEEE Transactions on Neural Networks Sept 2003: 1097-1109.

6.Cately C. Personal Communication – MIRG Research Meeting, April 2005.

7.Catley C and Frize M. "A Prototype XML-Based Implementation of an Integrated 'Intelligent' Neonatal Intensive Care Unit" Proc of the 4th IEEE Conference on Information Technology Applications in Biomedicine (2003): 323-25.

8.Catley C, Frize M, Walker RC, and Petriu DC. "Predicting Preterm Birth Using Artificial Neural Networks" Proc of the 8th IEEE Symposium on Computer-Based Medical Systems (2005): 103-8.

9.Catley C, Frize M, Petriu DC, Walker CR, and Yang L. "Towards a Web Services Infrastructure for Perinatal, Obstetrical, and Neonatal Clinical decision Support" Proc of the IEEE Conference on Engineering in Medicine and Biology (2004): 3334-37.

10. Catley C, Frize M, Walker CR, and St-Germain L. "Integrating Clinical Alerts into an XML-based Health Care Framework for the Neonatal Intensive Care Unit" Proc of the IEEE Conference on Engineering in Medicine and Biology (2003): 1276-79.

11. Canadian Neonatal Network, Bavinton, H. ed. The Canadian Neonatal Network Abstractor's Manual. Last updated 31 Jan 2005. <http://www.canadianneonatalnetwork.org/form/CNN%20Manual%20NEW%20DATABASE%20July%202004.pdf>. Accessed 02 Sept 2005.

12. Chapados N, and Bengio Y. "Input Decay: Simple and Effective Soft Variable Selection" IEEE Transactions on Neural Networks (2001): 1233-37.

13. Chawla N, Moore TE, Bowyer KW, Hall LO, Springer C, and Kegelmeyer P. "Bagging is A Small-Data-Set Phenomenon" IEEE Transactions on Neural Networks (2001): II-684 to II-689.

14. Chen D, and Cheng X. "Majority Vote Based on Weak Classifiers" Proc of the 5th International Conference of Signal Processing vol 3 (2000): 1560-62.

15. Chun SH, and Park YJ. "Dynamic adaptive ensemble case-based reasoning: application to stock market prediction" Expert Systems with Applications (2005): 435-43.

16. Cotea C, and Jiwani S. A Case-Based Reasoning (CBR) System for the Neonatal Intensive Care unit (NICU). Bachelor of Science in Engineering Thesis. University of Ottawa, Ottawa, ON, Canada. 2003.

17. Cunningham P, Carney P, and Jacob S. "Stability problems with artificial neural networks and the ensemble solution" Artificial Intelligence in Medicine (2000): 217-25.

18. Ennett CM. Imputation of Missing Values by Integrating Artificial Neural Networks and Case-based Reasoning. PhD in Electrical Engineering Thesis, Carleton University, Ottawa, ON, Canada. 2003.

19. Ennet CM. Coronary artery surgery mortality prediction using artificial neural networks. Master of Applied Science in electrical Engineering Thesis, School of Information Technology and Engineering, University of Ottawa, Ottawa, ON, Canada. 1999.

20. Ennett CM, and Frize M. "Validation of a Hybrid Approach for Imputing Missing Data" Proc of the IEEE Conference on Engineering in Medicine and Biology (2003): 1268-71.

21. Ennett CM, and Frize M. "Weight-Elimination Neural Networks Applied to Coronary Surgery Mortality Predcition" IEEE Transactions on Information Technology in Biomedicine (2003): 86-92.

22. Ennett CM, and Frize M. "Selective Sampling to Overcome Skewed a priori Probabilities with Neural Networks" Proc of the American Medical Informatics Association Symposium (2000): 225-29.

23. Ennett CM, Frize M, and Charette E. "Improvement and Automation of ANNs to Estimate Medical Outcomes" Medical Engineering Physics (2004): 321-28.

24. Ennett CM, Frize M, and Scales N. "Evaluation of the Logarithmic-Sensitivity Index as a Neural Network Stopping Criterion for Rare Outcomes" Proc of the IEEE Conference on Information Technology Applied to Biomedicine (2003): 207-10.

25. Ennett CM, Frize M, and Walker CR. "Neural Network Weight Extraction for Strategic and Clinical Decision-Making" Proc of the IEEE Conference on Engineering in Medicine and Biology Society (2003).

26. Ennett CM, Frize M, and Walker CR. "Influence of Missing Values on Artificial Neural Network Performance" Medinfo pt 1 (2001): 449-53.

27. Erdebil Y. Modeling Severe ATV Injuries Using Artificial Neural Networks. Master of Applied Science in Electrical Engineering Thesis. University of Ottawa, Ottawa, ON, Canada. 2005.

28. Fausett L. Fundamentals of Neural Networks. Englewood Cliffs, NJ: Prentice-Hall, 1994

29. Fowler M. and Scott K. UML Distilled: Second Edition. Reading, MA: Addison-Wesley, 2000.

30. Frize M, Taylor KB, Nickerson BG, Solven FG, Bor H. "A knowledge-based system for the intensive care unit" Proc of the IEEE Conference on Engineering in Medicine and Biology Society (1993): 677.

31. Frize M, Ennett CM, Stevenson M, and Trigg HC. "Clinical decision support systems for intensive care units: using artificial neural networks" Medical Engineering and Physics (2001): 217-225.

32. Frize M, Ibrahim D, Seker H, Walker RC, Odetayo MO, Petrovic D, Naguib RNG. "Predicting Clinical Outcomes for Newborns Using Two Artificial Intelligence Approaches" Proc of the 26th IEEE Conference on Engineering in Medicine and Biology (2004): 3202-05.

33. Frize M, Yang L, Walker RC, Oapos CAM. "Conceptual Framework of Knowledge Management for Ethical Decision-Making Support in

Neonatal Intensive Care" IEEE Transactions on Information Technology in Biomedicine Jun 2005: 205-15.

34. Frize M. Personal Communication. E-mail 20 Oct 2005 9:55 ET.

35. Frize M, Ennett CM, and Charette E. "Automated Optimization of Neural Networks in Estimating Medical Outcomes" Proc of the Conference on Information Technology Applied to Biomedicine & International Telemedical Information Society (2000): 168-73.

36. Frize M, Ennet CM, and Hebert P. "Improving the Efficiency and Effectiveness of Artificial neural Networks As Clinical Decision-Support Systems" MEDINFO (2001): 574.

37. Frize M, and Walker R. "Clinical Decision-Support Systems for Intensive Care Units Using Case-Based Reasoning" Medical Engineering and Physics (2000): 671-77.

38. Frize M, Walker CR, and Ennett CM. "Development of an Evidence-Based Ethical Decision-Making Tool for Neonatal Intensive Care Medicine" Proc of the Conference on Information Technology Applied to Biomedicine & International Telemedical Information Society (2003).

39. Garson GD. "Interpreting neural-network connection weights" AI Expert April 1991: 46-51.

40. Gilly D. UNIX In a Nutshell: System V Edition. Cambridge, MA: O'Rielly & Associates, Inc., 1992.

41. Goh TC. "Back-propagation neural networks for modeling complex systems" Artificial Intelligence in Engineering (1995): 143-51.

42. Gopinath P, and Reddy NP. "Toward Intelligent Web Monitoring: Performance of Committee Neural Networks vs Single Neural Network" Proc of the Conference on Information Technology Applied to Biomedicine & International Telemedical Information Society (2000): 179-82.

43. Hanley JA, and McNeil BJ. "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve" Diagnostic Radiology Apr 1982: 29-36.

44. Haykin S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ: Prentice Hall, 1999.

45. He Z, Xu X, and Deng S. " A cluster ensemble method for clustering categorical data" Information Fusion (2005): 143-51.

46. Igelnik B, Pao YH, LeClair SR, and Shen CY. "The Ensemble Approach to Neural-Network Learning and Generalization" IEEE Transactions on Neural Networks Jan 1999:19-30.

47. Islam MM, Yao X, and Murase K. "A Constructive Algorithm for Training Cooperative Neural Network Ensembles" IEEE Transactions of Neural Networks July 2003: 820-34.

48. Ji C, and Ma S. "Combinations of Weak Classifiers" IEEE Transactions on Neural Networks Jan 1997: 32- 42.

49. Kordylewski K, Graupe D, and Liu K. "A Novel Large-Memory Neural Network as an Aid in Medical Diagnosis Applications" IEEE Transactions on Information Technology in Biomedicine Sept 2001: 202-09.

50. LeBaron B, and Weigend AS. "A Bootstrap Evaluation of the Effect of Data Splitting on Financial Time Series" IEEE Transactions on Neural Networks Jan 1998: 213-20.

51. Lee Y, Song HK, and Kim MW. "An Efficient Hidden Node Reduction Technique for Multilayer Perceptrons" IEEE Transactions on Neural Networks (1991): 1937-42.

52. Lee SK, McMillan D, Ohlsson A, Pendray M, Synnes A, Whyte R, Chien LY, Sale J, the Canadian Neonatal Network. "Variations in practice and outcomes of the Canadian NICU Network: 1996-1997." Pediatrics (2000): 1070-9.

53. Leung FHF, Lam HK, Ling SH, and Tam PKS. "Tuning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm" IEEE Transactions on Neural Networks Jan 2003: 79-88.

54. Limsombunchai V, Gan C, and Minsoo L. "House Price Prediction: Hedonic Price Model vs. Artificial Neural Network" American Journal of Applied Sciences (2004): 193-201.

55. Lisboa PJG, and Mehri-Dehanvi AR. "Sensitivity Methods for Variable Selection Using the MLP" IEEE Transactions on Neural Networks (1996): 330-38.

56. Masters T. Practical Neural Network Recipes in C++. San Diego, CA: Academic Press, 1993.

57. Metz CE. "Basic Principles of ROC Analysis" Seminars in Nuclear Medicine Oct 1978: 283-298.

58. Mitchell T, Machine Learning. Hill McGraw, 1997.

59. Olden JD, and Jackson DA. "Illuminating the 'black box': a randomization approach for understanding variable contributions in artificial neural networks" Ecological Modeling (2002): 135-150.

60. Pan A. MIRG ANN Guide. NSERC Summer Student Report. 2002.

61. Pattichis CS, Christos N, and Middleton LT. "Neural Network Models in EMG Diagnosis" Transactions on Biomedical Engineering May 1995:486-96.

62. Penny W, and Frost D. "Neural networks in clinical medicine." Medical Decision Making (1996): 386-398.

63. Qi L. Network Tool for Neonatal Intensive Care Units. Master of Science in Information and Systems Science Thesis. Carleton University, Ottawa, ON, Canada. May 2005.

64. Richardson DK, Gray JE, McCormick MC, Workman K, and Goldmann DA. "Score for Neonatal Acute Physiology–A Physiologic Severity Index for Neonatal Intensive Care" Pediatrics (1993): 617-623.

65. Richardson DK, Corcoran JD, Escobar GJ and Lee SK. "SNAP-II and SNAPPE-II: Simplified newborn illness severity and mortality risk scores." Pediatrics (2001): 92-100.

66. Santos-Garcia G, and Varela G. "Prediction of postoperative morbidity after lung resection using an artificial neural network ensemble" Artificial Intelligence in Medicine (2004): 61-69.

67. Sohn SY, and Lee SH. "Data Fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea" Safety Science (2003): 1-14.

68. Tan X, Chen S, Yuan A, and Zhang X. "Recognizing Partially Occluded, Expression Variant Faces From Single Training Image per Person With SOM and Soft k-NN Ensemble" IEEE Transactions on Neural Networks July 2005: 875-86.

69. "The Area Under an ROC curve" Tape, T. G. Interpreting Diagnostic Tests: University of Nebraska Medical Center. <http://gim.unmc.edu/dxtests/roc3.htm> Accessed Sept 2005.

70. Tsymbal A, Cunningham P, Pechenizkiy M, and Puuronen S. "Search Strategies for Ensemble Feature Selection in Medical Diagnostics" Proc of the IEEE Symposium on Computer-Based Medical Systems (2003): 124.

71. Tsymbal A, Pechenizkiy M, and Cunningham P. "Diversity in search strategies for ensemble feature selection" Information Fusion (2005): 83-98.

72. Walker CR. Personal Communication – MIRG Research Meeting, August 2005.

73. Walker CR, and Frize M. "Are Artificial Neural Networks "Ready to Use" for Decision Making in the Neonatal Intensive Care Unit?" Pediatric Research (2004): 6-8.

74. Walker CR, and Frize M. "Use of And Artificial Neural Network (ANN) to Assess Duration of Ventilation in Neonatal Intensive Care Unit (NICU) Patients" Pediatric Academic Societies Poster (2000).

75. Wanas N, Audo G, Kamel MS, and Karray F. "On the Optimal Number of Hidden Nodes in a Neural Network" IEEE Transactions on Neural Networks (1998): 918-21.

76. Wang W, Richards G, and Rea S. "Hybrid Data Mining Ensemble for Predicting Osteoporosis Risk" Proc of the IEEE Conference on Engineering in Medicine and Biology (2005). Abstract.

77. Weigend AS, Rumelhart DE, and Huberman. "Back-propagation, weight-elimination and time series prediction" Proc of the Connectionist Models Summer School. (1990): 105-116.

78. Weigand AS, Rumelhart DE, and Huberman. "Generalization by Weight-Elimination applied to Currency Exhange Rate Prediction" Proc of the IEEE International Conference on Neural Networks (1991): 2374-79.

79. Weisstein EW. "Hyperbolic Tangent." From MathWorld--A Wolfram Web Resource. <http://mathworld.wolfram.com/HyperbolicTangent.html>. Accessed 14 Sept 2005.

80. West D, Mangiameli P, Rampal R, and West V. "Ensemble strategies for a medical diagnostic decision support system: A breast cancer diagnosis

application" European Journal of Operational Ressearch: Computing, Artificial Intelligence and Information Technology (2005): 532-51.

81. Witten IH, and Frank E. Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations. San Francisco: Morgan Kaufmann Publishers, 2000.

82. Wolf L, and Bileschi S. "Combining Variable Selection with Dimensionality Reduction" Proc of the IEEE Conference on Computer Vision and Pattern Recognition (2005): 801-06.

83. Yang L. Pilot Usability Study of UI Prototype for Collaborative Adaptive Decision Support in Neonatal Intensive Care Unit. Masters of Applied Science Electrical Engineering Thesis, 2004. University of Ottawa. Ottawa, Ontario, Canada.

84. Yang L, Frize M, and Walker R. "Towards Ethical Decision Support and Knowledge Management in Neonatal Intensive Care" Proc of the IEEE Conference on Engineering in Medicine and Biology Society (2004): 3420-23.

85. Zhou Z, and Jiang Y. "Medical Diagnosis with C4.5 Rule Preceded by Artificial Neural Network Ensemble" IEEE Transactions on Information Technology in Biomedicine March 2003: 37-42.

Appendix A. Artificial Neural Network Research Framework (ANN RFW) Design Diagrams

This Appendix contains the flow charts and a hierarchy diagram of the ANN RFW. Some of the descriptions of the flow charts give a brief description of what was updated in the individual files for the creation of the ANN RFW from the existing ANN application. Following the charts are sections describing the stopping criteria and divide and conquer algorithm used in the software.

A.1. Flow Charts and Block Diagrams

This section contains flow charts for many of the functions and subroutines that comprise the ANN RFW. Not all MatLab files have been elaborated in a flow chart. Some that are important have been described briefly in comments along side of other flow charts where they are called. A legend for the flow chart symbols is given in Figure A-1.

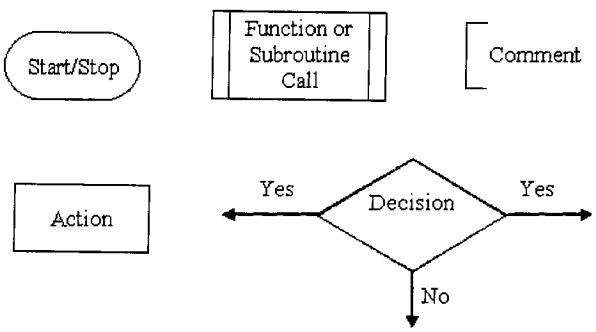

Figure A-1: Legend for Flow Chart Symbols.

The initiation point of the software is **main_autoANN_Sequential** shown in Figure A-2.

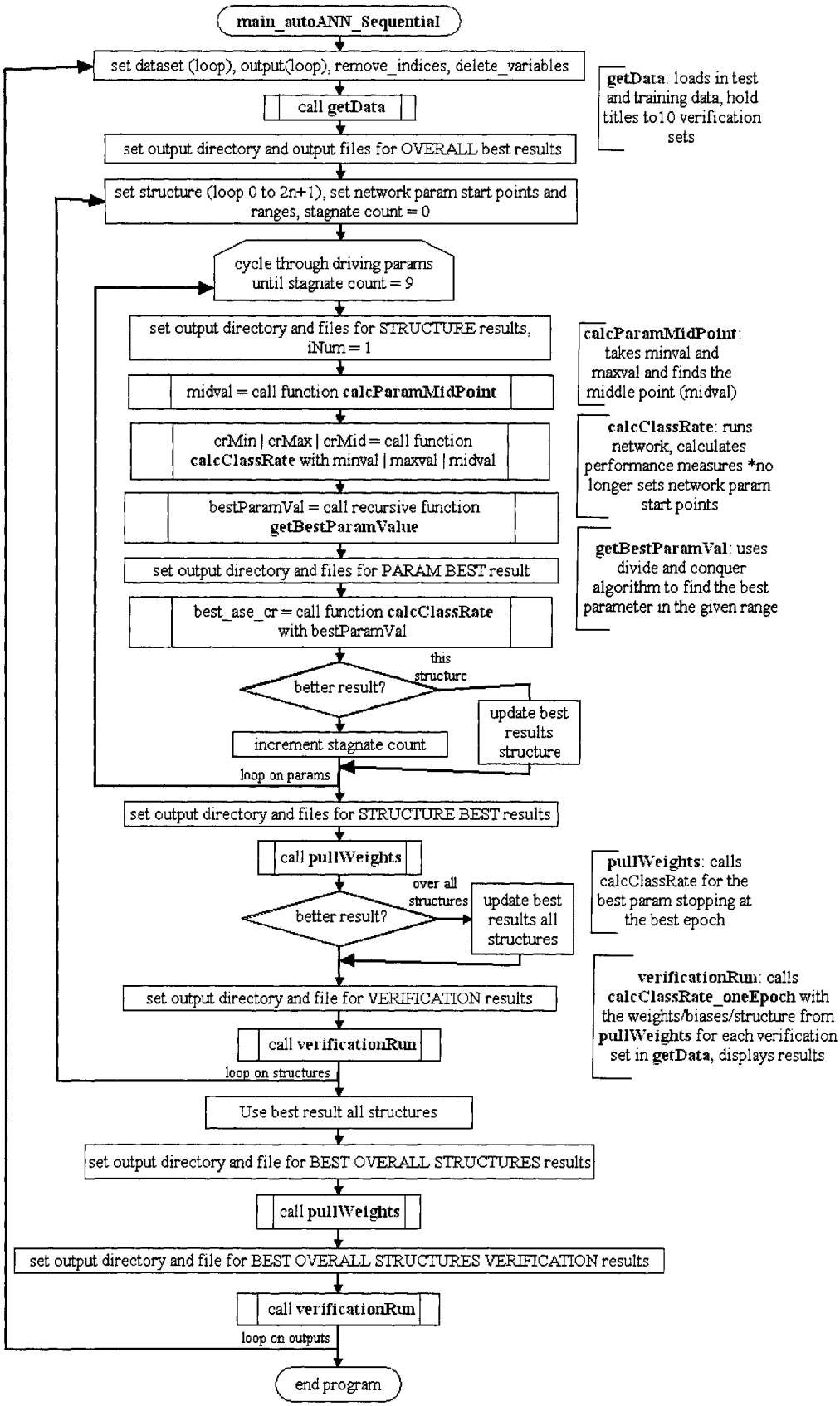

Figure A-2: Flow Chart: main_autoANN_Sequential.m.

The function **calcParamMidPoint** takes the minimum point and the maximum point of a range as arguments and returns the midpoint. The tolerance for the smallest division is set for each of the nine driving parameters as 'roundfactor'. If the new range is smaller than allowed by the roundfactor, a flag is returned indicating that no more divide and conquer iterations are possible.

The function **getBestParamValue** uses the minimum, maximum and midpoint value of a range to calculate three sets of results. These are compared and a new range is set with a divide and conquer algorithm as described in section A.3. The parameter value of the best of the three results is returned as the *bestParamVal*.

The function **calcClassRate** calls all the functions and various other .m files that perform the ANN learning and statistic calculations. It returns the final performance measure statistics in an array: *classrate*. A flow chart of **calcClassRate**'s operation is shown in Figure A-3. The default params are no longer set in **calcClassRate**. As a requirement for the automatic parameter update process of the ANN RFW, the default values of the parameters are now set in **main_autoANN_Sequential**.

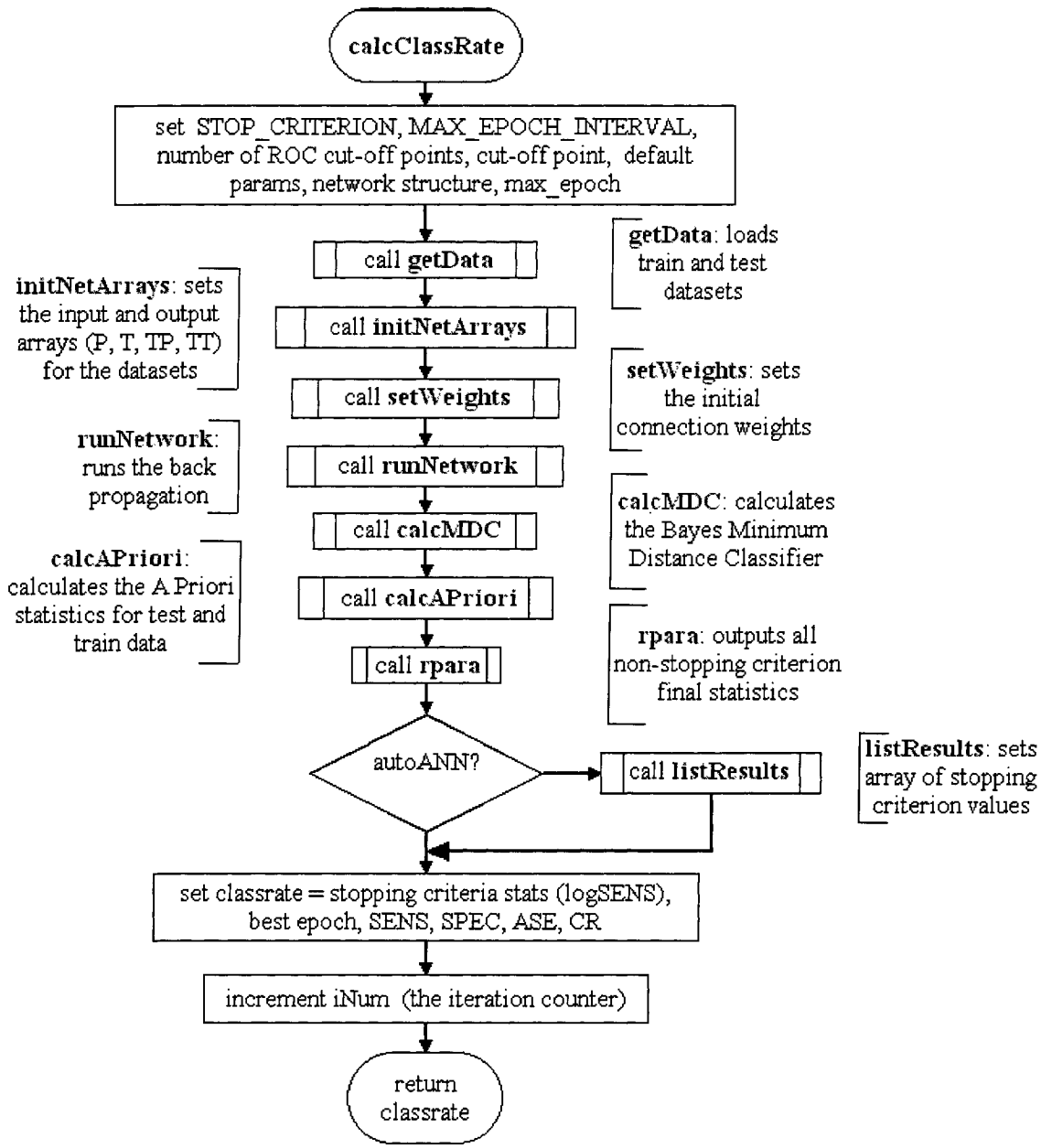

Figure A-3: Flow Chart: calcClassRate.m.

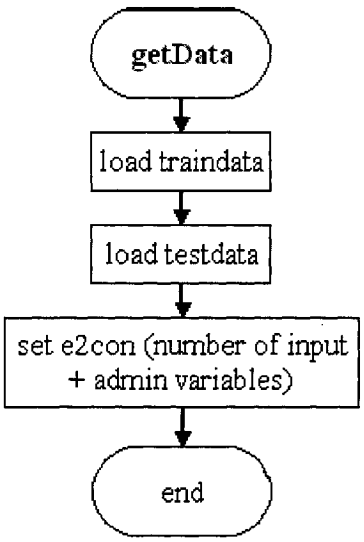

Figure A-4: Flow Chart: getData.m.

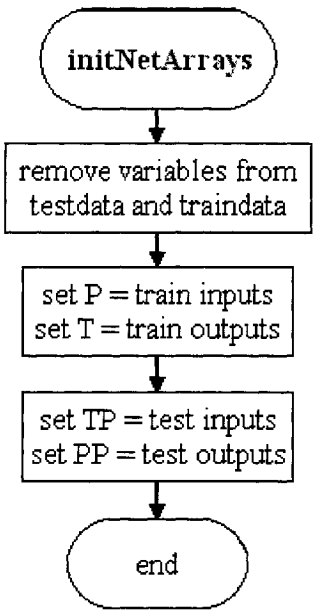

Figure A-5: Flow Chart: initNetArrays.m.

Data is retrieved from comma-delimited plain text file defined in **getData** and prepared in **initNetArrays**. The data is separated into inputs and outputs and the administrative fields are removed from the input variables array (variables T and P) in **initNetArrays**.

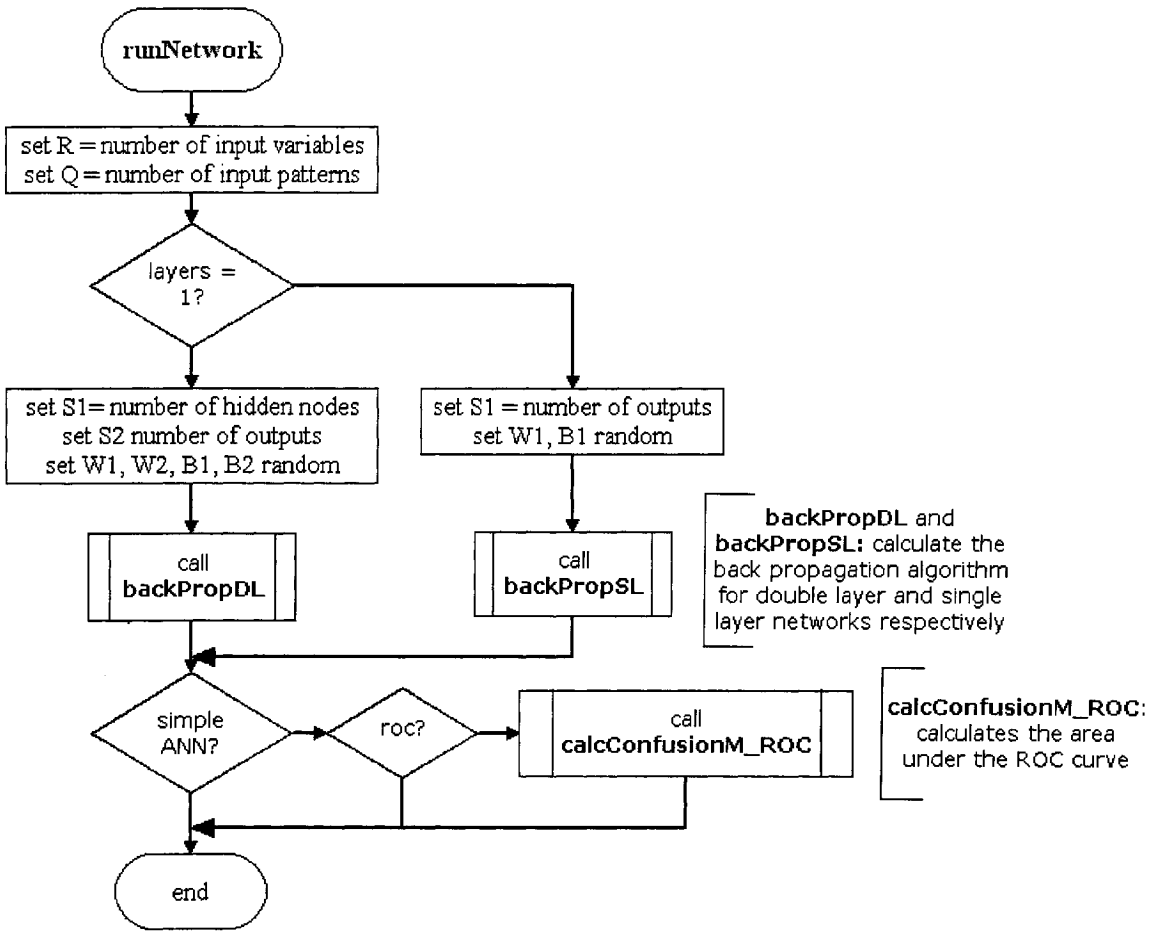

Figure A-6: Flow Chart: runNetwork.m.

runNetwork sets all variables for the ANN structure, input cases and output values required to perform the back propagation. In the ANN RFW **runNetwork** has flags set to call **DL_oneEpoch** and **SL_oneEpoch** which performs a single epoch of the ANN effectively doing the prediction.

backPropSL/backPropDL perform the back propagation learning. Weight-elimination is performed here if it is enabled.

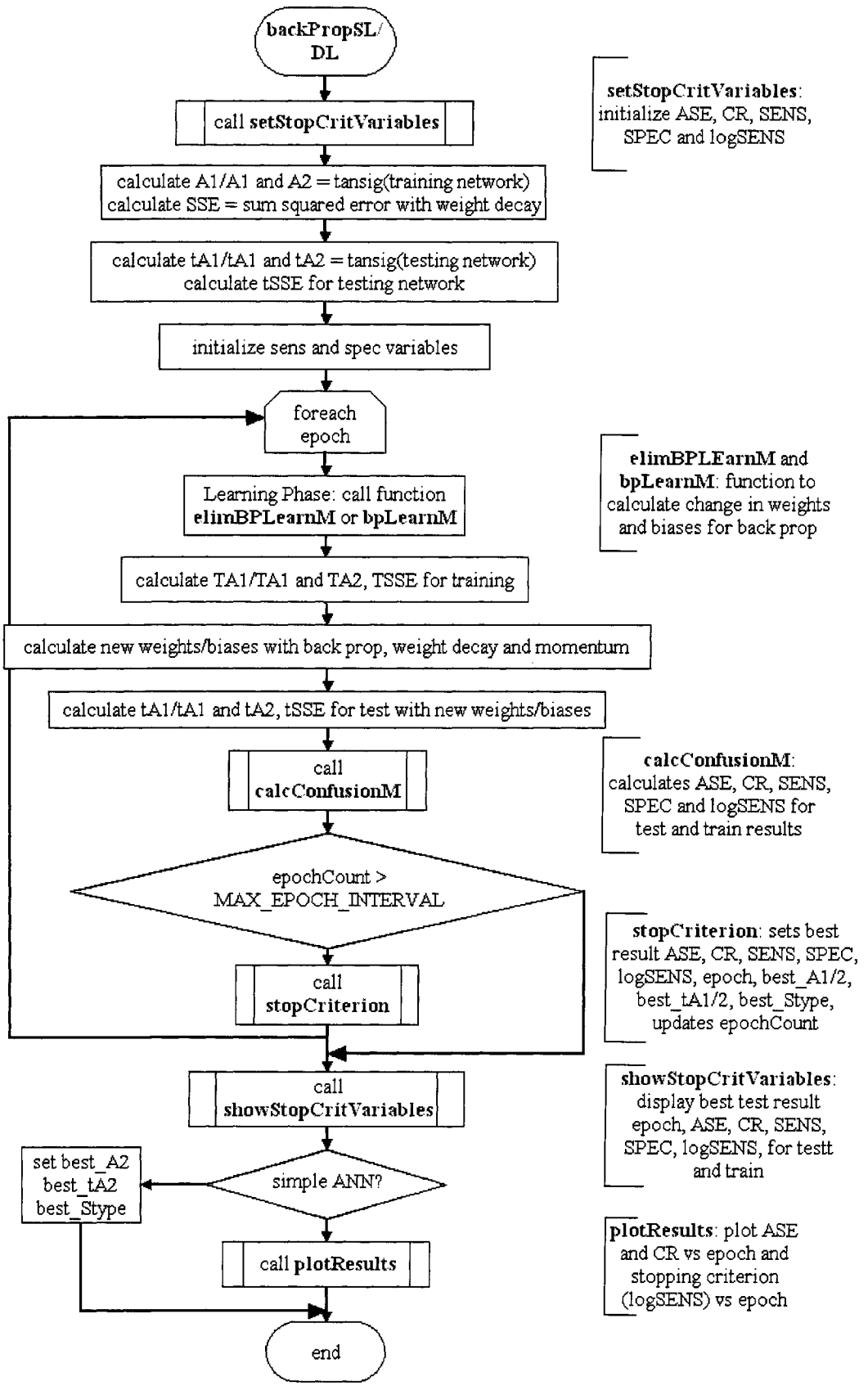

Figure A-7: Flow Chart: backPropDL/SL.m.

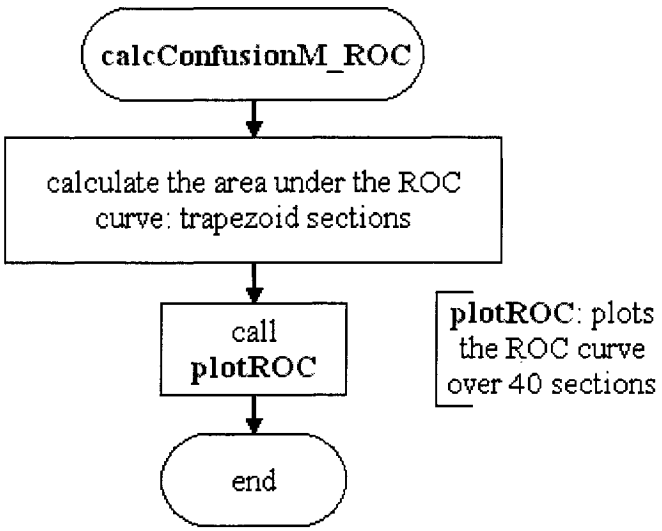

Figure A-8: Flow Chart: calcConfusionM_ROC.m.

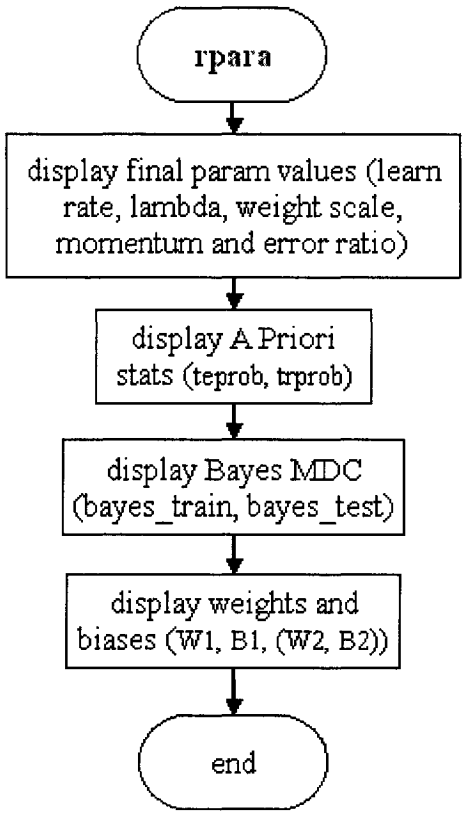

Figure A-9: Flow Chart: rpara.m.

calcConfusionM_ROC calculates the area under to ROC curve. It uses a trapezoid approximation. The number of sections is set in **calcClassRate**. There are 40 sections set with a value of 41 points including end points going from -1 to 1 in intervals of 0.05.

rpara displays many of the non-stopping criteria related final results of the ANN. These include the final non-multiplication factor parameter values after incrementing and decrementing over epochs, a priori stats of the train and test datasets, the minimum distance classifier and, the weights and biases at the final epoch.

A.1.1. ANN RFW File Hierarchy

The updates to the existing ANN tool to develop the ANN RFW were secluded to a few of the files. Major updates were written into only three of the pre-existing files while five new files were added. The higher level flow charts representing most of the files remain the same or not significantly different from those presented in section A.1. A file hierarchy diagram shown in Figure A-10 indicates which files had major changes (includes changes and new file creation), minor changes and those that remain unchanged.

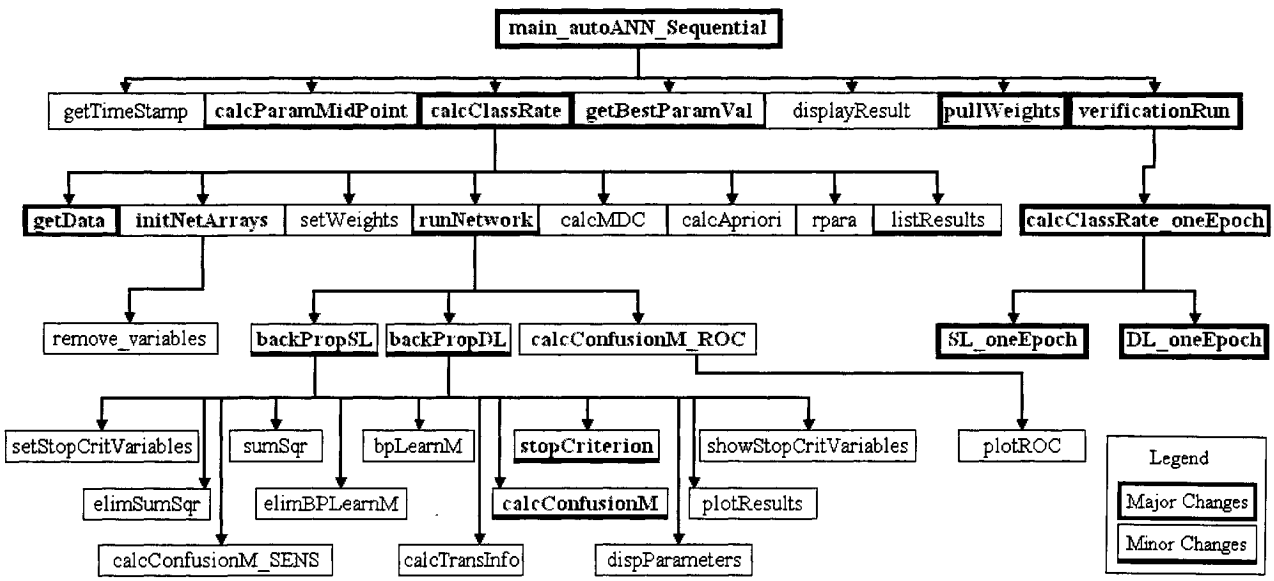

Figure A-10: File Hierarchy: ANN RFW.

Figure A-10 shows a file hierarchy of the ANN RFW. The boxes in bold outline represent files that were updated or created from the existing tool to the expanded ANN RFW. The boxes with a bold underline indicate files that underwent minor changes. Not

all connections have been shown as some files belong to more than one level of the hierarchy. Files have been placed in the topmost level in which they operate.

A.2. Stopping Criterion

A stopping criterion usually centers on a performance measure. Peripheral considerations, beyond the performance measure, help to dictate when an ANN has learned sufficiently. For the MIRG ANN, the combination of a performance measure and various epoch criteria create a multipart stopping criterion.

A.2.1. Performance Measure Criteria

The performance measures that have been used by MIRG are lowest Average Squared Error (ASE), highest Sensitivity, highest Classification Rate and highest Log Sensitivity Index. The ASE criteria is not a good option for categorical results as it calculates a distance error whereas in categorical results from 1 to -1 splitting at 0, with a desired result of -1 a calculated result equal to 0.9 or 0.1 are both equally incorrect. ASE would define the 0.9 as being more incorrect than the 0.1 result. Sensitivity and Classification results do not perform well for instances where datasets are highly skewed in favour of one category. MIRG has chosen the Log Sensitivity Index as the most useful performance measure for clinical databases [24] and it is therefore the default for the ANN RFW.

One of the updates made to the existing ANN tool in this work was to error catch for the inherent failure of the log sensitivity with an ideal result. If sensitivity and specificity both equal one (100% of the results are correct) then the equation blows up to infinity. To avoid the infinite answer the following addition was made to the code: if the argument of the log function is equal to zero then the log sensitivity is automatically set to 5. The value 5 was chosen because the output is set to five decimal places and with a sensitivity of 0.99999 and a specificity of 0.99999 the log sensitivity is 4.69897 and 5 is a small amount larger ensuring that the log sensitivity for the best result was always the highest.

A.2.2. Epoch Criteria: Starting and Stopping Points

There are three specific epoch starting or stopping points that are part of the multipart stopping criteria: one for the start point and two for the stopping point. To give the ANN some time to learn, results were not recorded until 25 epochs had passed [27]. The maximum number of epochs that could be attempted is 2000 to prevent a slowly climbing result to continue indefinitely. The third criteria allows for the run to end before 2000 epochs if the best point has been achieved and has remained the best point for 500 epochs [35].

A.3. Divide and Conquer Algorithm

The divide and conquer algorithm used in the ANN software is more complicated than the standard – "find best min-mid or max-mid pair, reset to that range, repeat". Tracing through the existing ANN code yielded the options shown in Figure A-12 to Figure A-15.

There exist 13 different possible combinations of the results (performance measure) with three parameters (minimum, maximum and midpoint of the current parameter range).

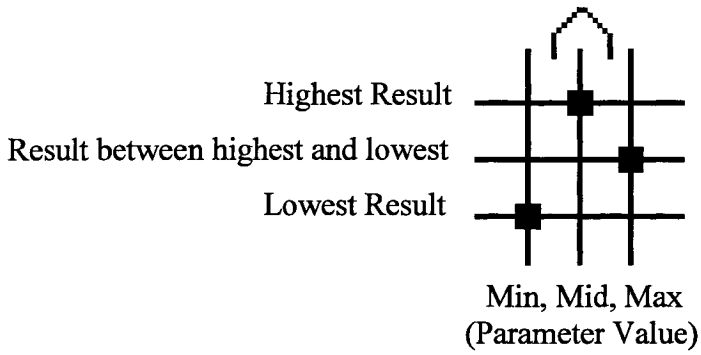

Figure A-11: Divide and Conquer: Example instance.

The example shown in Figure A-11 indicates that:

- the result from the minimum parameter was the lowest,
- the result from the midpoint value of the parameter was the highest
- the result for the maximum parameter value was somewhere in between.

In this instance, both the minimum and maximum points are reset leaving the midpoint untouched. The ^ shape indicates the new range.

The following figures give the 13 instances and the new range or the identification of a terminal point. A terminal point indicates the end of recursion: a final point has been found so the recursive functions return. It is important to note that the standard simple divide and conquer algorithm does not reset the minimum and maximum values while leaving the midpoint the same. The instances shown in Figure A-14 are not typical of a simple divide and conquer algorithm.

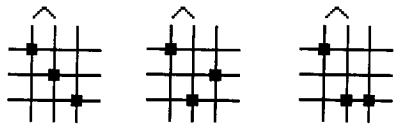

Figure A-12: Divide and Conquer: Minimum point best.

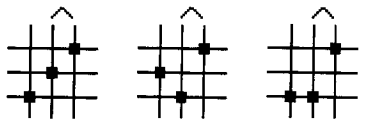

Figure A-13: Divide and Conquer: Maximum point best.

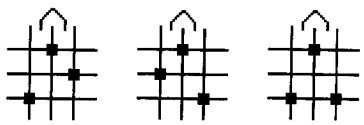

Figure A-14: Divide and Conquer: Midpoint best.

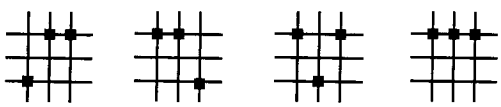

Figure A-15: Divide and Conquer: Terminal points.

Appendix B. Moving from The Haley Corporation's Easy Reasoner to the Case Based Reasoner System (CBRS) k-Nearest Neighbour (k-NN) Application

The Medical Information-technology Research Group (MIRG) Case Based Reasoner System (CBRS) was created to replace an out of box CBR tool, the Haley Corporation Easy Reasoner. The Easy Reasoner is a black box software package that does not permit the researcher access to the algorithms used to match cases. At some point about a decade ago, at the request of previous researchers who have since split off from MIRG, the Easy Reasoner's code was change by Haley Corporation and recompiled. The resulting altered executable was then distributed back to the researchers and used for patient case matching.

In 2002, the author performed an inspection of all available documentation, as well as contacted previous researchers and the Haley Corporation. No documentation as to the changes made was found. Through conversations with Haley Corporation representatives, it was discovered that all support of the tool was discontinued when the altered executable was issued. While attempting to set up the tool for use by MIRG after having not used it for a number of years, it was discovered that when MIRG split off from the original group that the portion of the Easy Reasoner tool required to set up matching for new databases belonged did not go with MIRG. Communications with the previous researchers were necessary to request the creation of the binary driving files

required by the Easy Reasoner for the database used by Ennett throughout experiments performed for work published in [18] and [20].

A search of the tools available in industry at the time, yielded no similarly functioning case matching tools for MIRG. After a single project was completed with the Easy Reasoner, it was decided to build our own matching tool that could also impute missing values as required by the process defined by Ennett [18].

The resulting CBRS gives MIRG independence from the Easy Reasoner's binary file creation requirement and full control and understanding of the matching algorithm used. The CBRS is also fully embeddable into the MIRG CDSS with web services with no licensing issues often encountered with out-of-box tools.

To compare the CBRS with the Easy Reasoner: datasets were built with the same cases used by Ennett: 5102 patient cases with nine input variables representing the cases with complete values in the SNAPP-II variables in the CNN database [18]. The 5102 cases were randomly separated into two sections of 1744 and 3358 cases as defined by Ennett [18]. The separation was performed 12 times to create 12 random configurations of the 5102 cases. The 1744 case set became the Query or Open Patient Cases table. This table contains the cases with missing values. Missing values were artificially inserted into the cases following the proportions aligning with the original CNN database of 20 thousand cases [18, Table 6-1]. The number of cases these percentages amounted to in the 1744 case Query sets used here are shown in Table B-1.

Table B-1: Number of Artificial Missing Values in CBRS Test Datasets

	PO2FIO2R	LURINE	LSERUM	LBLOODP
Missing Values	1066	919	689	286

The weights used for testing the CBRS were obtained from Ennett [18, Table 7-1] and are shown in Table B-2.

Table B-2: k-NN Adjusted Weights for CBRS Test

PO2FIO2R	LURINE	APGAR5	LTEMPF	SGA	LSERUM	BTHWT	LBLOODP	SEIZURE
100	65	43	42	32	28	19	19	19

The results from the CBRS and from the Haley Corporation Easy Reasoner with the ANN defined weights and average value missing value imputation are shown in Table B-3.

Table B-3: CBRS vs Easy Reasoner Results

	PO2FIO2R	LURINE	LSERUM	LBLOODP
CBRS: Mean[a]	104.84%	182.87%	1.12%	15.38%
CBRS: Standard Deviation[a]	71.46%	125.44%	0.07%	1.45%
CBRS: Maximum[a]	3533.87%	7353.00%	9.40%	124.07%
Easy Reasoner: Mean[b]	94.62%	174.23%	1.14%	15.73%
Easy Reasoner: Standard Deviation[b]	146.80%	488.09%	1.18%	16.13%
Easy Reasoner: Maximum[b]	1683.33%	10300.00%	13.26%	113.33%

[a]CBRS: Taken from 10 datasets (12 datasets removing the best and worst)
[b]Easy Reasoner: Hybrid with ANN Weights [18, Table 7-4]

Although the results of the CBRS were not as good as those from the out-of-box application, the greater flexibility, ease of use and knowledge of and control over internal workings of the CBRS tool make it a good choice. Continuing updates will be made to the algorithms within the CBRS to improve the results.

Appendix C. Elaboration on the Outcome Prediction Model Definition Process (OPMDP)

Following is a list of the steps in the Outcome Prediction Model Definition Process (OPMDP).

1. Clean Data. Remove all ambiguous data and outliers.
2. Impute missing values:
    a. Pull cases with no missing values from dataset to create a new dataset of 'complete cases'.
    b. Normalize data and place in plain text files for use by ANN: training, test and verification sets.
    c. Find ANN Prediction Model with all input variables. See Step 4.
    d. Transform ANN Weights to k-NN format. See section 3.7.
        i. Use Extended Garson's Method to find Relative Weights (Importance) of all input variables.
        ii. Scale Relative Weights from 0 to 100 to create k-NN formatted weights.
    e. Setup **actual** value data in Microsoft Access with the 'complete cases' as the Match table (Discharged Patients) and the cases with missing values as the Query table (Open Patient Cases).
    f. Setup and Run CBRS tool.
    g. Create a database including all 'complete cases' and newly filled cases.
3. Normalize data and place in plain text files for use by ANN: training, test and verification sets.
4. Find ANN Prediction Model:
    a. Setup and Run ANN RFW.
    b. Find best structure(s).

c. Remove input variables using extended relative weight method to find the relative importance of input variables.

d. Repeat a-c until the results begin to degrade: creates a minimum dataset. **The ANN prediction model consists of weights (W1,W2), biases (B1,B2), normalization values (minimum, maximum, mean and standard deviation), structure (number of hidden layers and nodes) and inputs (which input variables are used and the order of the variables).**

5. To verify the ANN prediction model on a dataset not previously used (i.e. from a different facility or a different year) use the CCVT (Committee of Classifiers/Verification Tool).
    a. Transfer the ANN prediction model into the appropriate sections of the CCVT code.
    b. Setup data in the correct variable order (to align with the ANN prediction model weights). Data can be either: normalized using the normalization values from the model or as real values, leaving the software to normalize the data. The input must be in plain text space or comma delimited format as for the ANN RFW tool.
6. Using the ANN Prediction Model as a base, four prediction methods are available with MIRG's current tools:
    a. Single ANN prediction.
    b. ANN Committee of Classifiers
    c. ANN Range Overlap
    d. CBRS with k-NN mean of closest matching cases calculation prediction.
7. Single ANN prediction:
    a. A Prototype is set up that uses a single ANN Prediction Model.
    b. Data is entered by hand in actual format into the GUI.
    c. Output is saved to a comma delimited plain text file.
    d. GUI used in the first prototype was in MatLab. The real user interface that the doctors see will be done through web services and tools like PADS.

e. To test a dataset on a Single ANN, it is currently best to use the CCVT with a committee of one (1) member set up.

8. ANN Committee of Classifiers:
    a. Transfer the five (5) top ANN prediction model weights into the CCVT code.
    b. Setup data in the correct order (to align with the ANN prediction model weights). Data can be either: normalized using the normalization values from the model or used as real values, leaving the software to normalize the data. The input must be in plain text format.
    c. The output will give the result for each of the input cases for each of the five (5) members of the committee as well as the committee result.
9. ANN Range Overlap
    a. If the ANN tool is used to predict a continuous outcome an overlap of the individual sub-outcomes can be used to find a more specific result. i.e. instead of a result of LOS>7 days with a single sub-outcome, a result of LOS>7 days and LOS<14 days can be determined by overlapping the sub-outcome results.
    b. Use the Single ANN or Committee of Classifiers results for each sub-outcome and then combine them.
10. CBRS: k-NN mean of closest matching cases calculation outcome prediction:
    a. The ANN weights must be transformed into k-NN format (a whole number from 0 to 100). Use the Extended Garson's Method to define the Relative Importance and then linearly scale the results for the inputs with 100 as the highest.
    b. Data must be in actual value form. Enter the train and test set combined as the 'Discharged Patients'. Use the verification data sets with the outcome field as the missing value to be filled.

C.1. Activity Diagrams for OPMDP

This section contains the UML 2.0 Activity diagrams for the description of the Outcome Prediction Model Definition Process. Is the starting point. All other activity diagrams are referred to beginning with a task from the main activity diagram. A bubble in the diagram with a '*' preceding the label is link to another activity.

These diagrams should be used in conjunction with the Outcome Prediction Model Definition Process described in the last section and the descriptions of applications and equations from the main body of this thesis to develop prediction models.

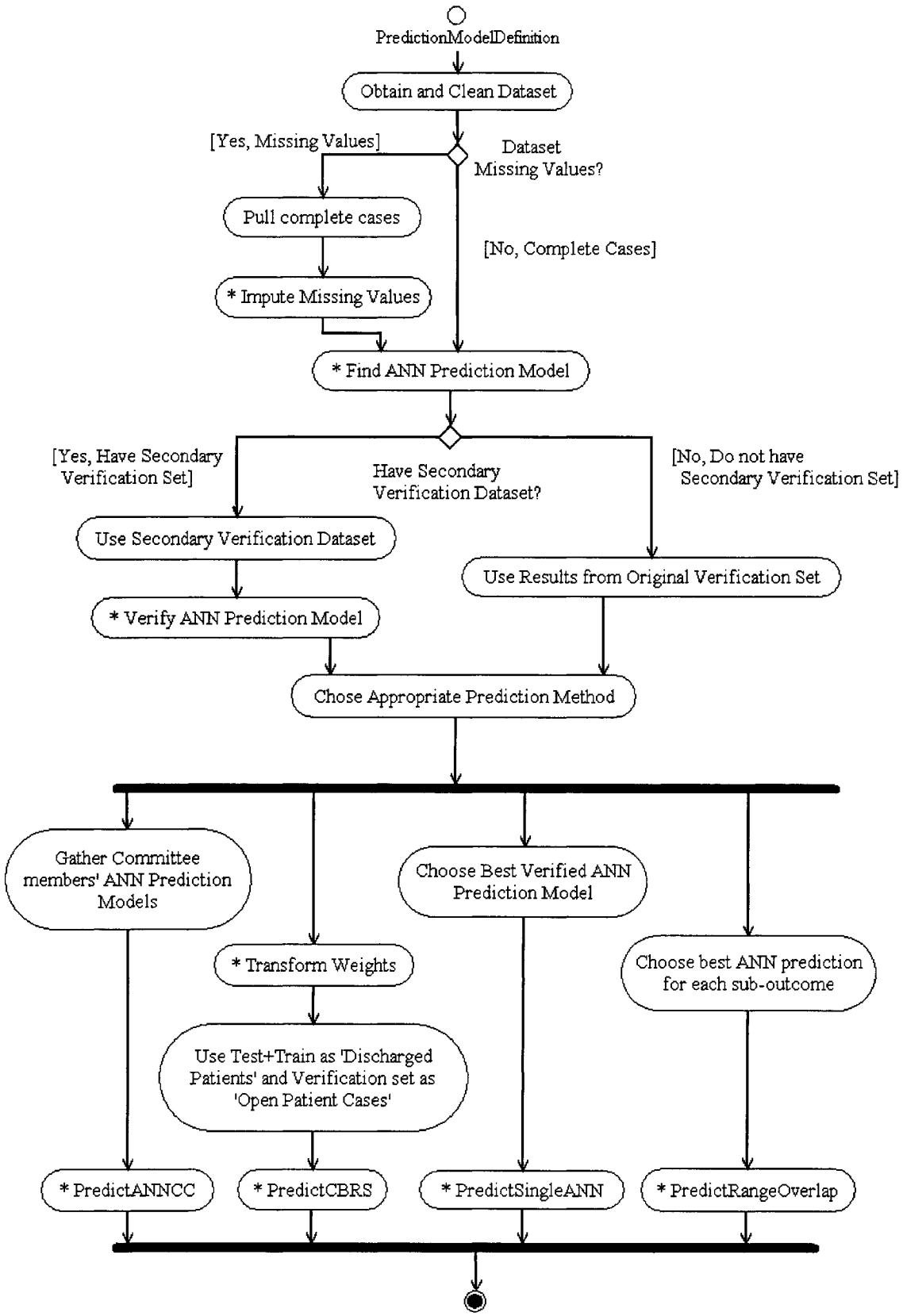

Figure C-1: Activity Diagram: Outcome Prediction Model Definition Process.

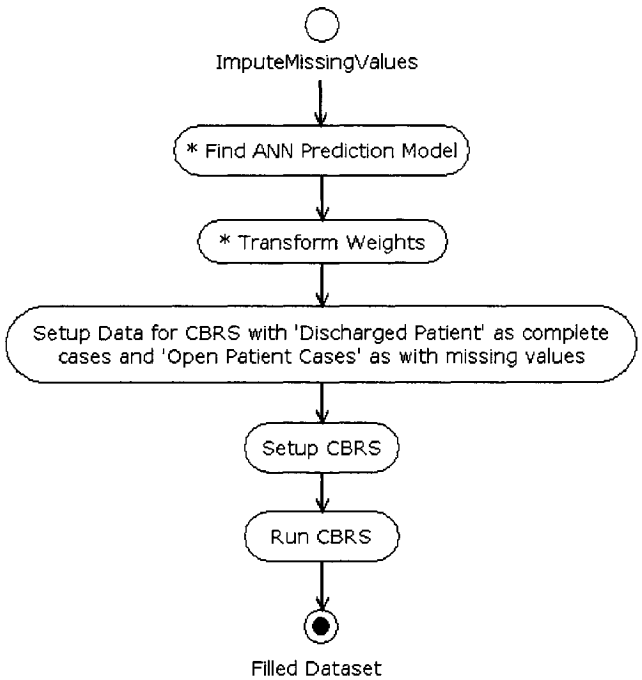

Figure C-2: Activity Diagram: Impute Missing Values.

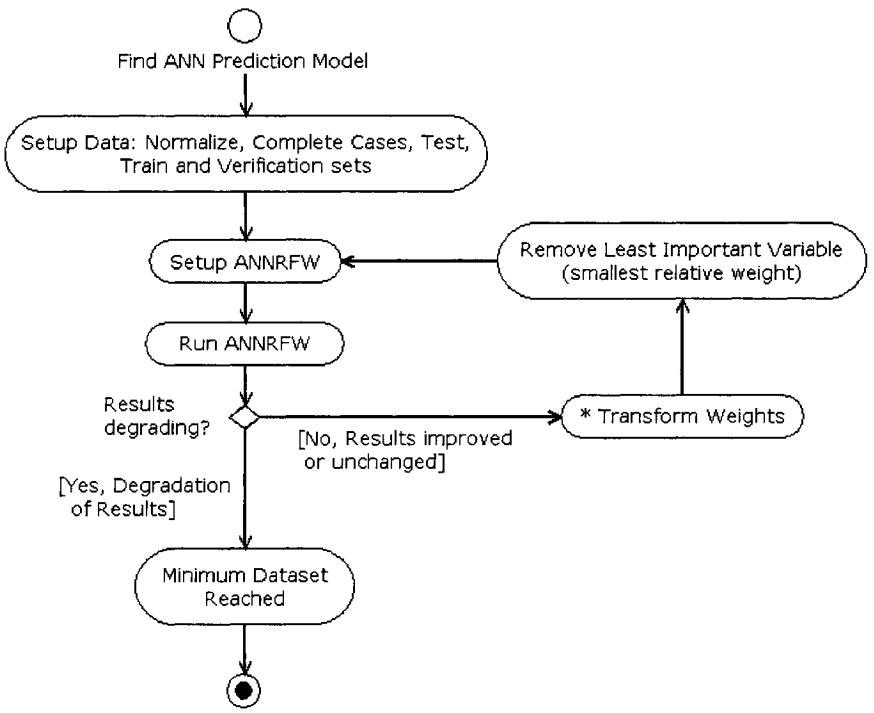

Figure C-3: Activity Diagram: Find ANN Prediction Model.

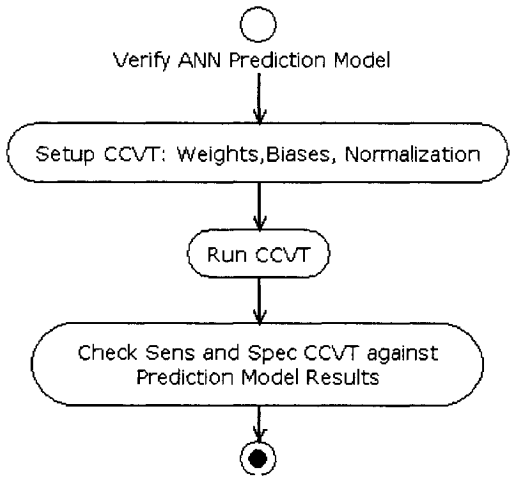

Figure C-4: Activity Diagram: Verify ANN Prediction Model.

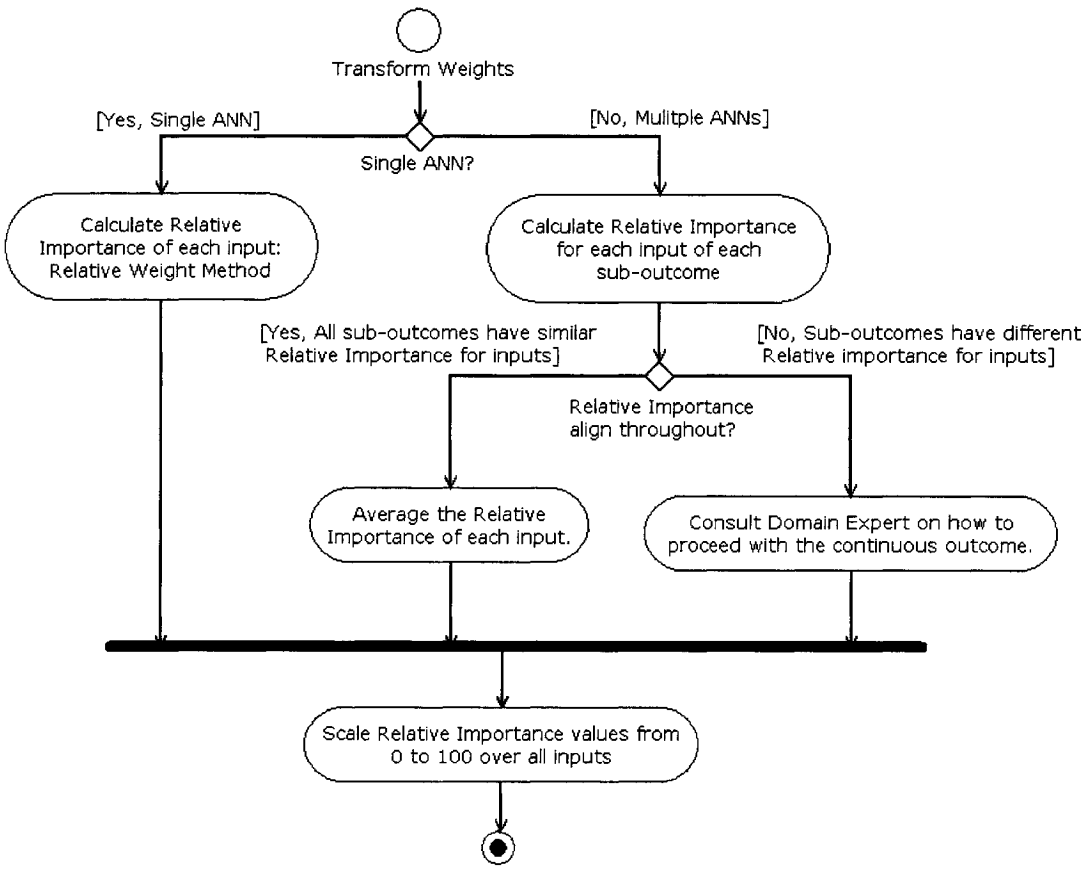

Figure C-5: Activity Diagram: Transform Weights.

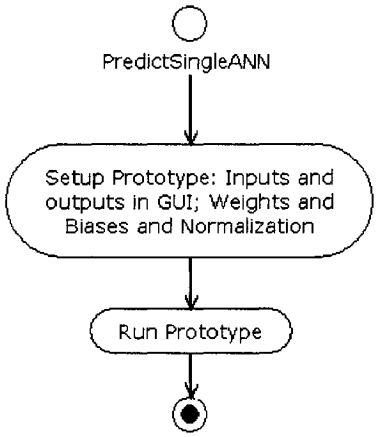

Figure C-6: Activity Diagram: ANN Prediction: Single ANN.

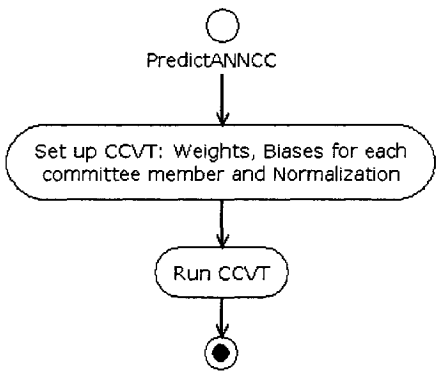

Figure C-7: Activity Diagram: ANN Prediction: Committee of Classifiers.

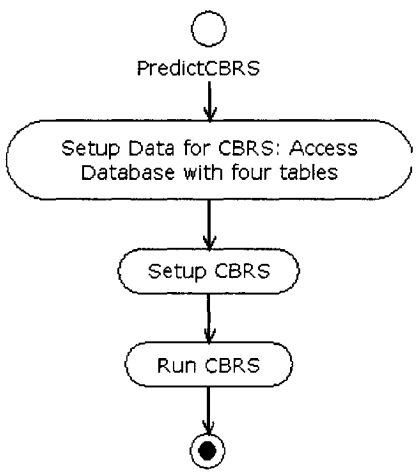

Figure C-8: Activity Diagram: Continuous Outcome Prediction: CBRS.

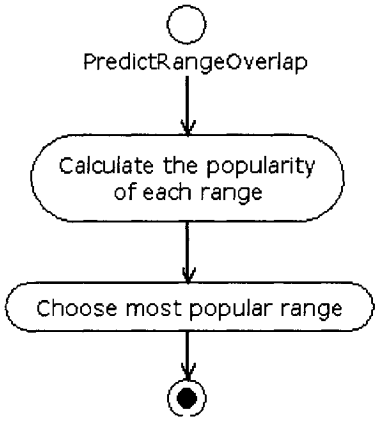

Figure C-9: Activity Diagram: Continuous Outcome Prediction: Range Overlap.

Appendix D. SNAP Scoring System

Table D-1: SNAP Scoring System Variables and Description

Variable		Points scored			
		0	1	3	5
Mean blood pressure	High	<66	66-80	81-100	>100
	Low	>35	30-35	20-29	<20
Heart rate	High	<180	180-200	201-250	>250
	Low	>100	80-100	40-79	<40
Respiratory rate	High	<60	60-100	>100	–
Temperature (°F)	Low	>96	95-96	92-94.9	<92
pO2	Low	>65	50-65	30-50	<30
pO2/FiO2 ratio	Low	>3.5	2.5-3.5	0.3-2.49	<0.3
pCO2	High	<50	50-65	66-90	>90
Oxygenation index	High	<0.07	0.07-0.2	0.21-0.40	>0.40
Hematocrit	High	<66	66-70	>70	–
	Low	>35	30-35	20-29	<20
White blood count	Low	>5.0	2.0-5.0	<2.0	–
Immature/total neutrophil ratio	High	<0.21	≥0.21	–	–
Absolute neutrophil count	Low	>999	500-999	<500	–
Platelet count	Low	>100	30-100	0-29	–
Blood urea nitrogen	High	<40	40-80	>80	
Creatinine	High	<1.2	1.2-2.4	2.5-4.0	>4.0
Urine output	Low	>0.9	0.5-0.9	0.1-0.49	<0.1
Indirect bilirubin	High				
- bili for birth weight>2kg		<15	15-20	>20	–
- bili/kg for birth weight<2kg		<5	5-10	>10	–
Direct bilirubin	High	<2.0	>2.0	–	–
Sodium	High	<150	150-160	160-180	>180
	Low	>130	120-130	<120	–
Potassium	High	<6.6	6.6-7.5	7.6-9.0	>9.0
	Low	>2.9	2.0-2.9	<2.0	–
Total calcium	High	<12	>12	–	–
	Low	>6.9	5.0-6.9	<5.0	–
Ionized calcium	High	<1.4	>1.4	–	–
	Low	>1.0	0.8-1.0	<0.8	–
Glucose	High	<150	150-250	>250	–
	Low	>40	30-40	<30	–
Serum bicarbonate	High	<33	>33	–	–
	Low	>15	11-15	<10	–
Serum pH	Low	>7.30	7.20-7.30	7.10-7.19	<7.10
Presence of seizures		None	Single	Multiple	–
Presence of apnea		none	Response to stimuli	No response to stimuli	Complete apnea
Stool guaiac		Negative	Positive	–	–

* Additional points scored for the Perinatal Extension (SNAPPE) are:

Birth weight ≤ 749 g	30 points
Birth weight 750-999 g	10 points
Apgar < 7 at 5 minutes	10 points
Small for gestational age (<5th percentile)	5 points [Richardson *et al.* 1993a]

Appendix E. CNN Dataset A Priori Statistics

The following tables show the A Priori statistics (the number of positive and negative cases present) for each of the training (Table E-1), testing (Table E-2) and verification main dataset (Table E-3). Values are given for the outcomes and sub-outcomes used in this work.

Table E-1 : A Priori Statistics: Training Datasets.

	Original MORT	Resampled MORT	VENTgt0	VENTgt1	VENTgt2	LOSgt7	LOSgt14	LOSgt21	LOSgt28
Positive Cases	317	922	3245	2773	2115	4178	2617	1880	1477
Negative Cases	8295	8295	5367	5839	6497	4434	5995	6732	7135
Total	8612	9217	8612	8612	8612	8612	8612	8612	8612
Positive Cases Percent	4%	10%	38%	32%	25%	49%	30%	22%	17%
Negative Cases Percent	96%	90%	62%	68%	75%	51%	70%	78%	83%
Total	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table E-2: A Priori Statistics: Test Datasets.

	Original MORT	Resampled MORT	VENTgt0	VENTgt1	VENTgt2	LOSgt7	LOSgt14	LOSgt21	LOSgt28
Positive Cases	158	461	1633	1379	1038	2090	1349	948	738
Negative Cases	4148	4148	2673	2927	3268	2216	2957	3358	3568
Total	4306	4609	4306	4306	4306	4306	4306	4306	4306
Positive Cases Percent	4%	10%	38%	32%	24%	49%	31%	22%	17%
Negative Cases Percent	96%	90%	62%	68%	76%	51%	69%	78%	83%
Total	100%	100%	100%	100%	100%	100%	100%	100%	100%

Table E-3: A Priori Statistics: Verification Datasets.

	Original MORT	Resampled MORT	VENTgt0	VENTgt1	VENTgt2	LOSgt7	LOSgt14	LOSgt21	LOSgt28
Positive Cases	237	691	2413	2057	1672	3109	1969	1417	1107
Negative Cases	6222	6222	4046	4402	4887	3350	4490	5042	5352
Total	6459	6913	6459	6459	6459	6459	6459	6459	6459
Positive Cases Percent	4%	10%	37%	32%	24%	48%	30%	22%	17%
Negative Cases Percent	96%	90%	63%	68%	76%	52%	70%	78%	83%
Total	100%	100%	100%	100%	100%	100%	100%	100%	100%

Appendix F. Charts of Results

The chapter contains the graphs of the results not presented in the main body of this work. There are two sections:

- section F.1 contains results from the artificial neural network frame work (ANN RFW) for those outcomes and sub-outcomes not presented in the main body of the work including MORT, VENTgt1, VENTgt2, LOSgt7, LOSgt14 and LOSgt28, and
- section F.2 contains committee of classifier results from the committee of classifier verification tool (CCVT) for those outcomes and sub-outcomes not presented in the main body of the work including MORT, VENTgt0, VENTgt2, VENTgt12, LOSgt7, LOSgt14, LOSgt21 and LOSgt28..

F.1. Extended Artificial Neural Network Research Framework (ANN RFW) Results

This section contains results from the 20 structures attempted for each of the outcomes and sub-outcomes using the ANN RFW. Results of the training, test and the average of the 10 verification datasets are shown for each structure (number of hidden nodes). The five best results based on the logarithmic sensitivity index performance measure are highlighted with enlarged shapes: triangles for the training set, circles for the test set and squares for the verification set average.

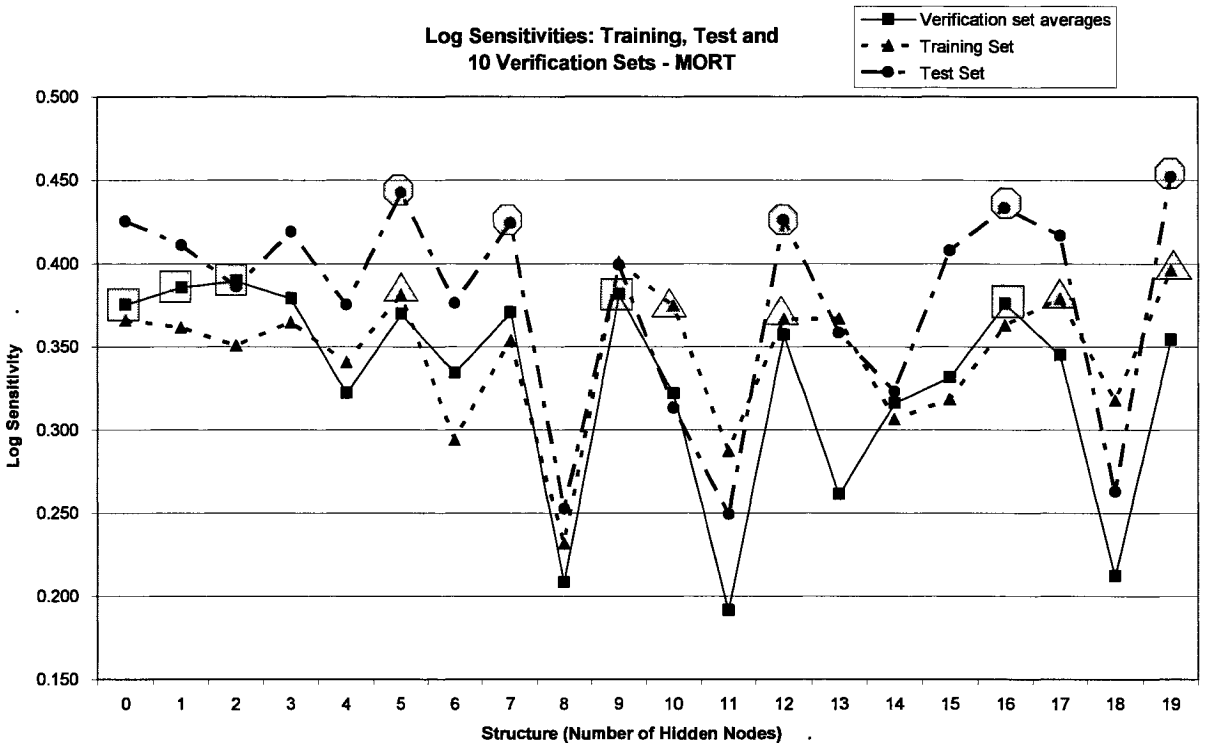

Figure F-1: Log Sensitivities of 20 attempted structures for MORT. Boxes highlight the five best of each Training, Test and 10 Verification sets.

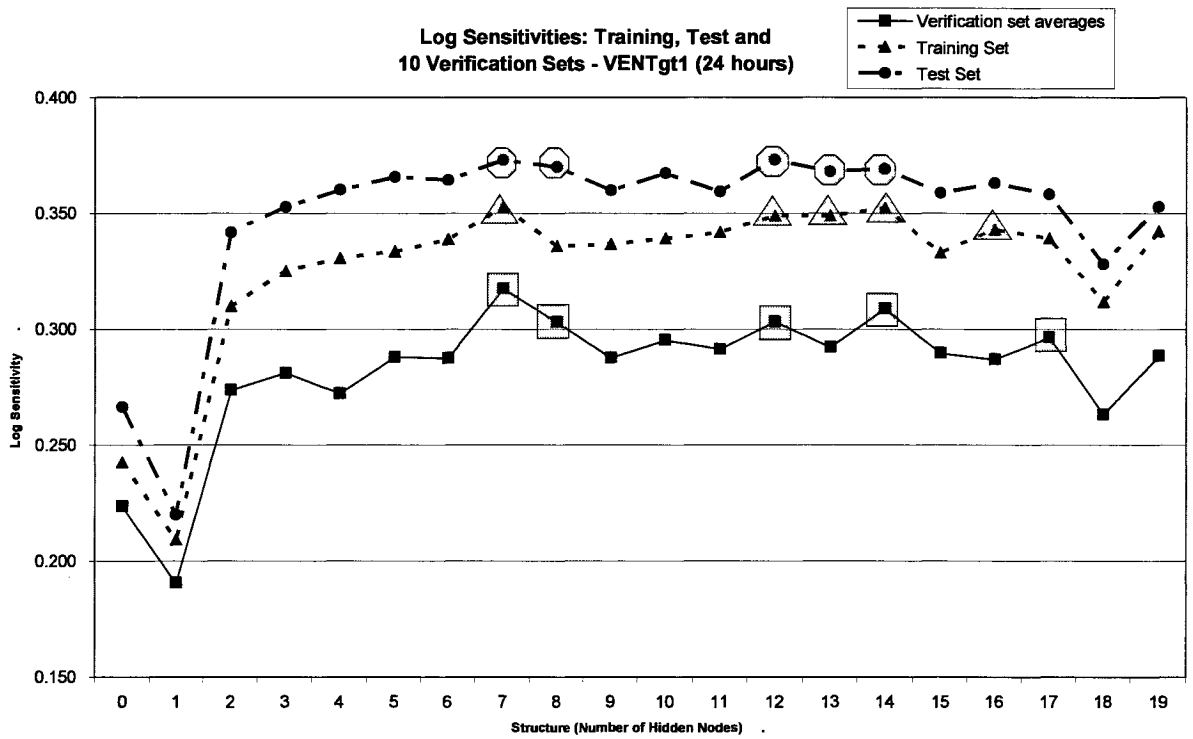

Figure F-2: Log Sensitivities of 20 attempted structures for VENTgt1 (24 hours). Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

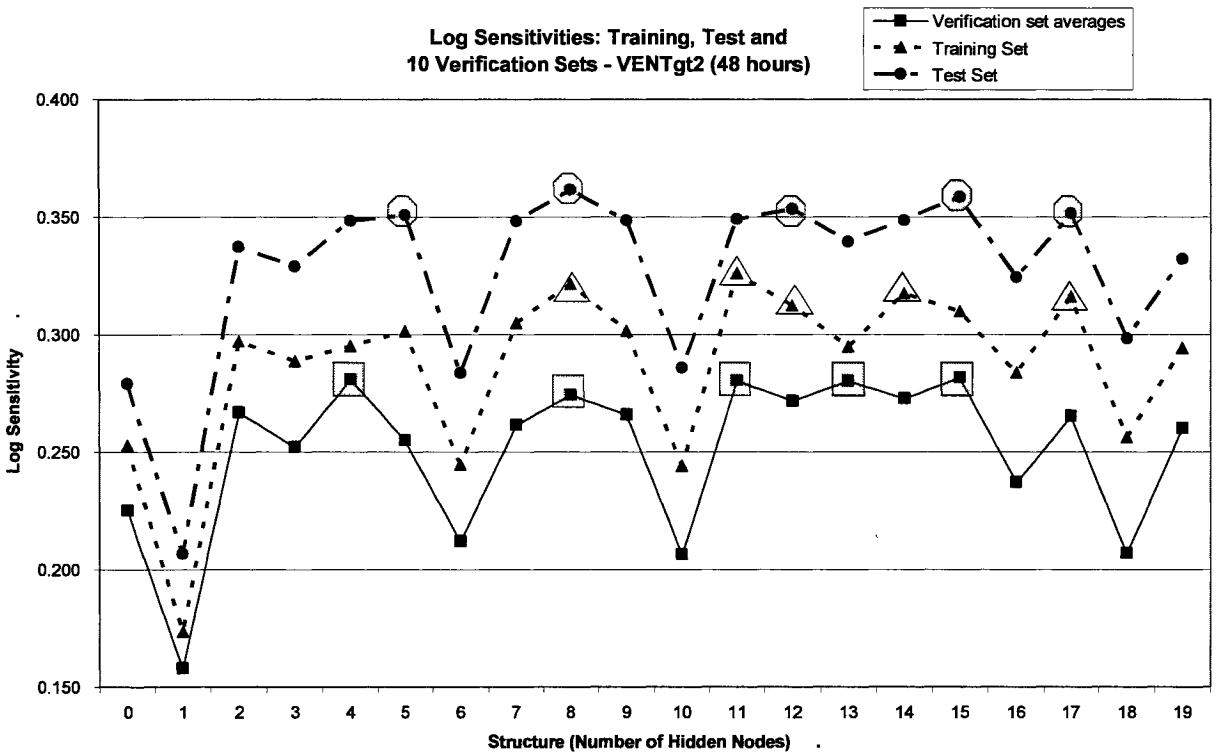

Figure F-3: Log Sensitivities of 20 attempted structures for VENTgt2 (48 hours). Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

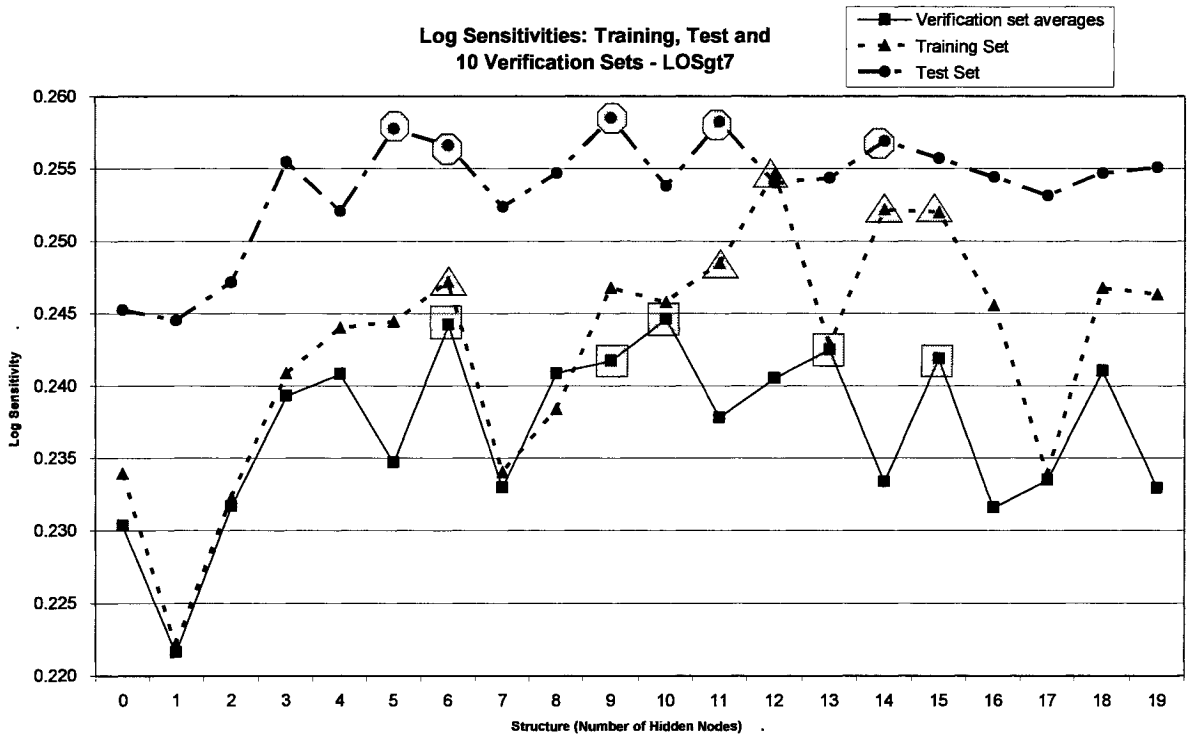

Figure F-4: Log Sensitivities of 20 attempted structures for LOSgt7 days. Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

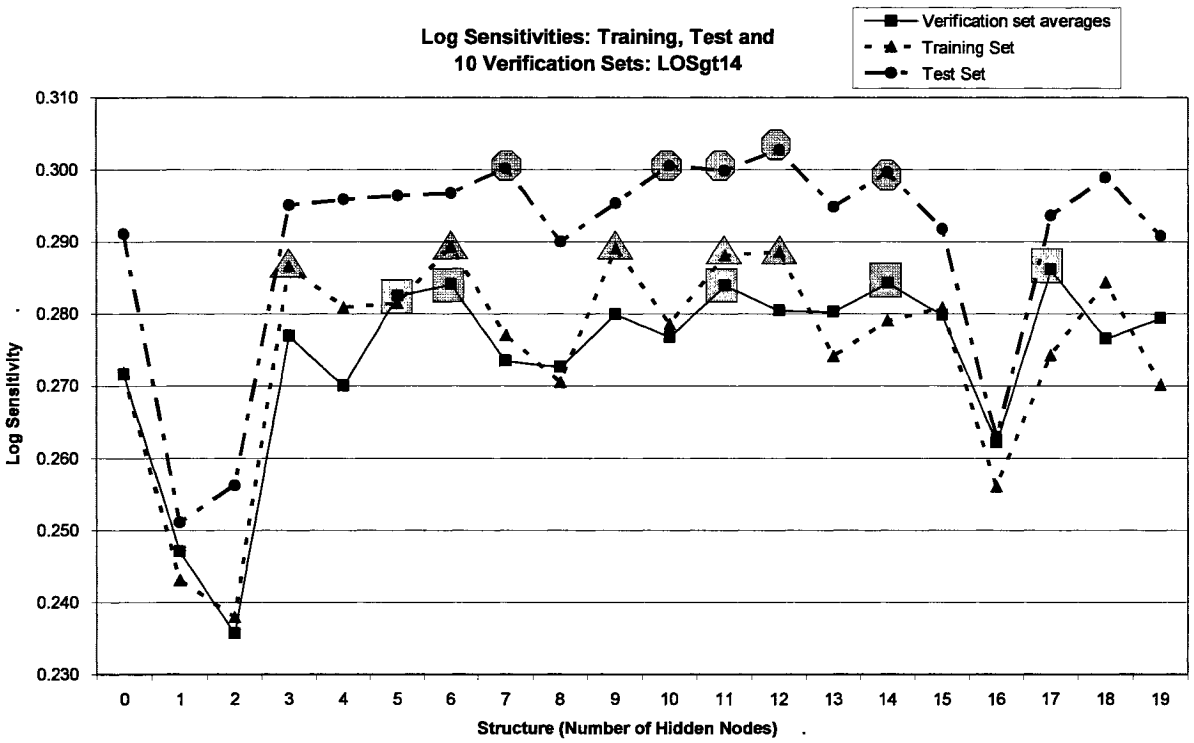

Figure F-5: Log Sensitivities of 20 attempted structures for LOSgt14 days. Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

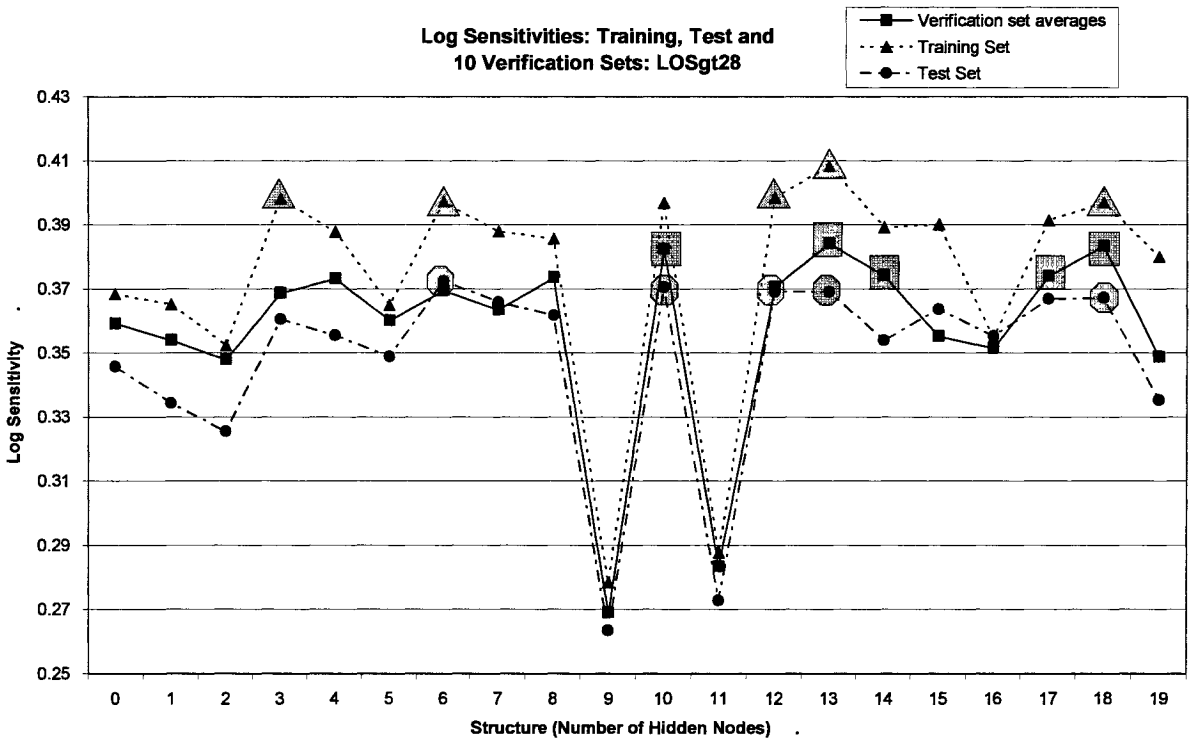

Figure F-6: Log Sensitivities of 20 attempted structures for LOSgt28 days. Boxes highlight the five best of each Training, Test and average of 10 Verification sets.

F.2. Committee of Classifier Results

This section contains results for the five best structures and the committee constructed with the five best structures as its members. The results for the training, test and average of the 10 verification sets are shown using the logarithmic sensitivity index as the performance measure. The best of the group is defined using the result from the verification sets. The best result is highlighted with a shaded box. The dashed line indicates the verification results from the committee.

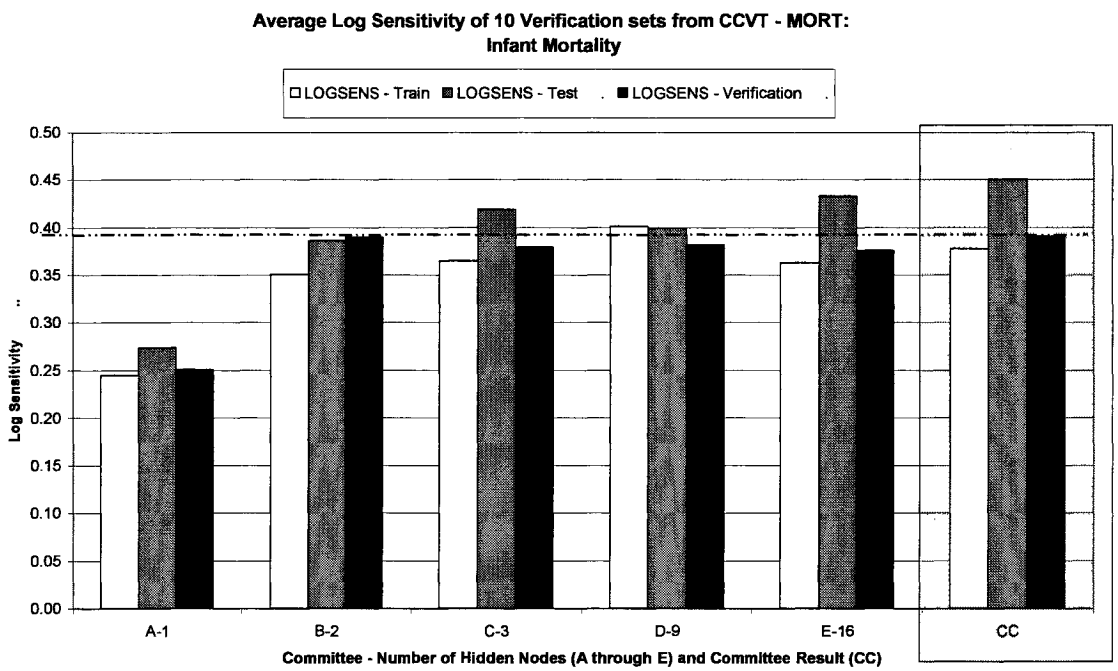

Figure F-7: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for MORT.

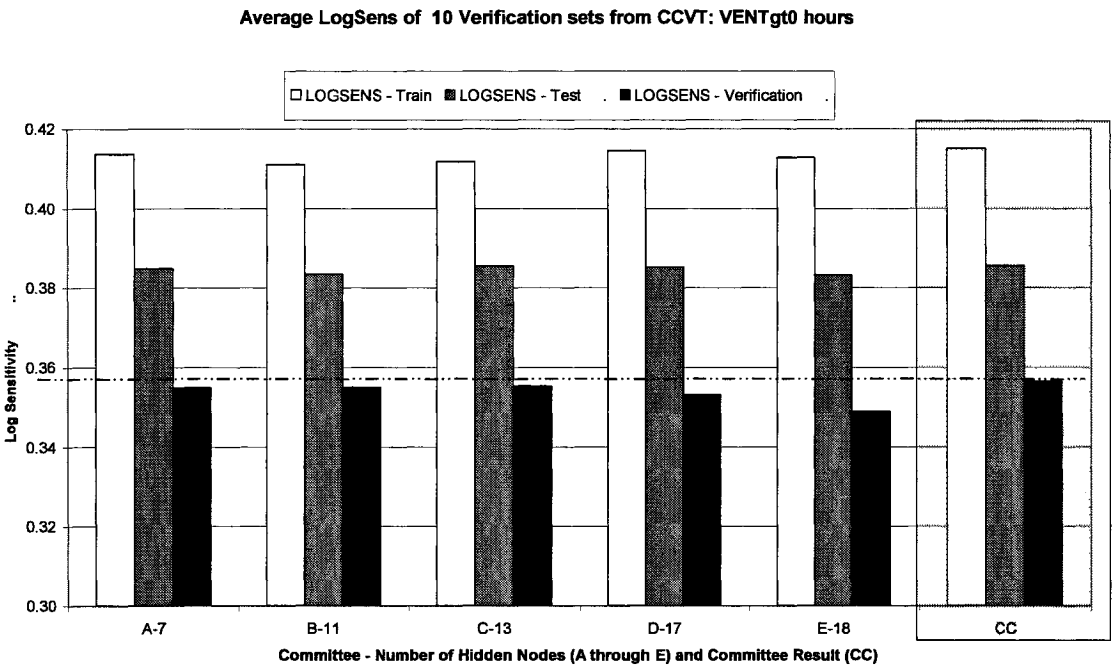

Figure F-8: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for VENTgt0 hours.

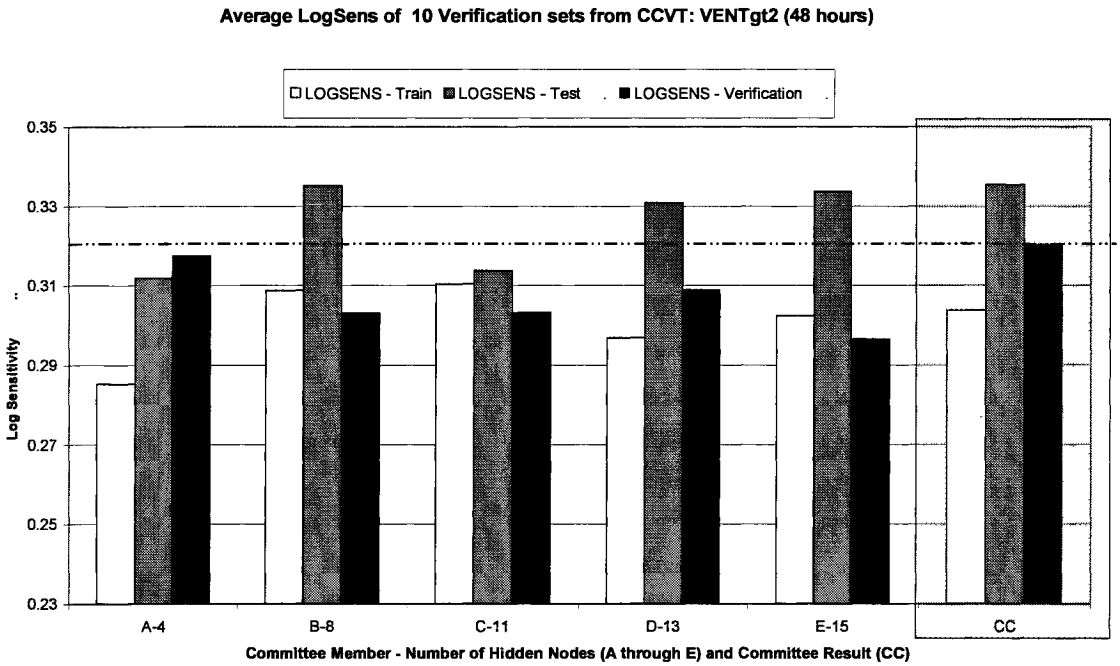

Figure F-9: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for VENTgt2 (48 hours).

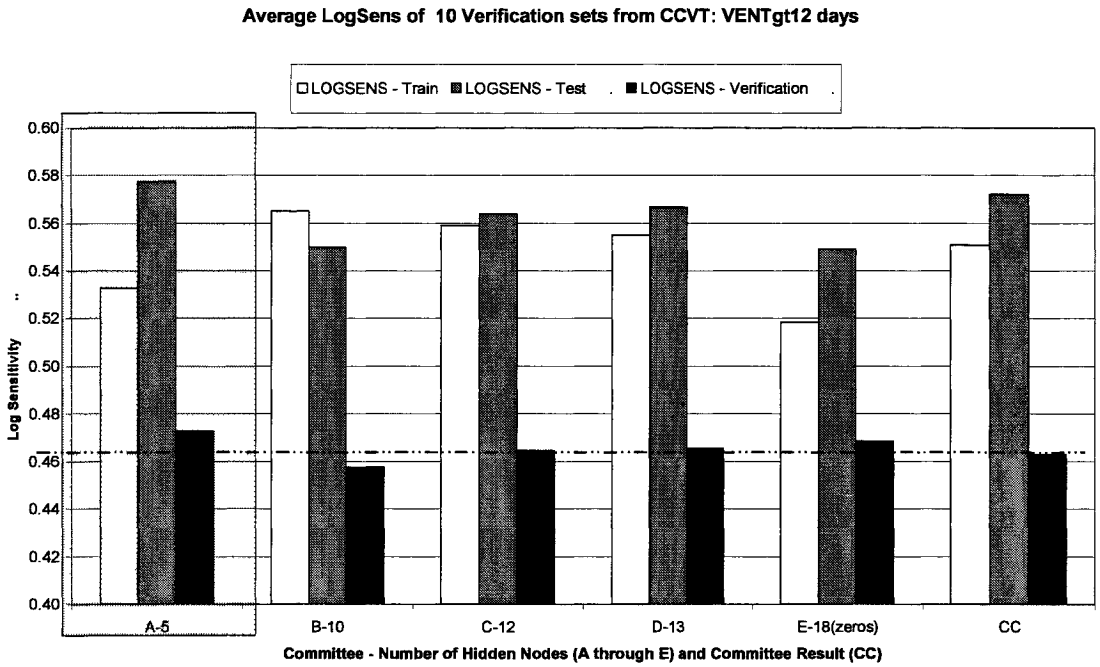

Figure F-: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for VENTgt12 days.

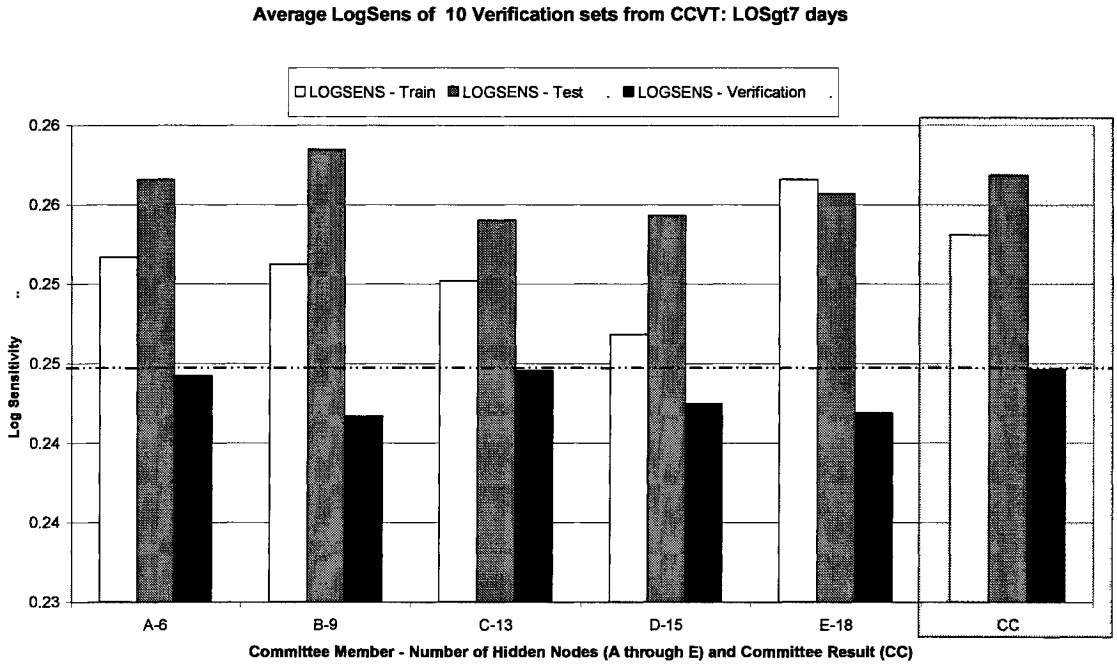

Figure F-10: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for LOSgt7 days.

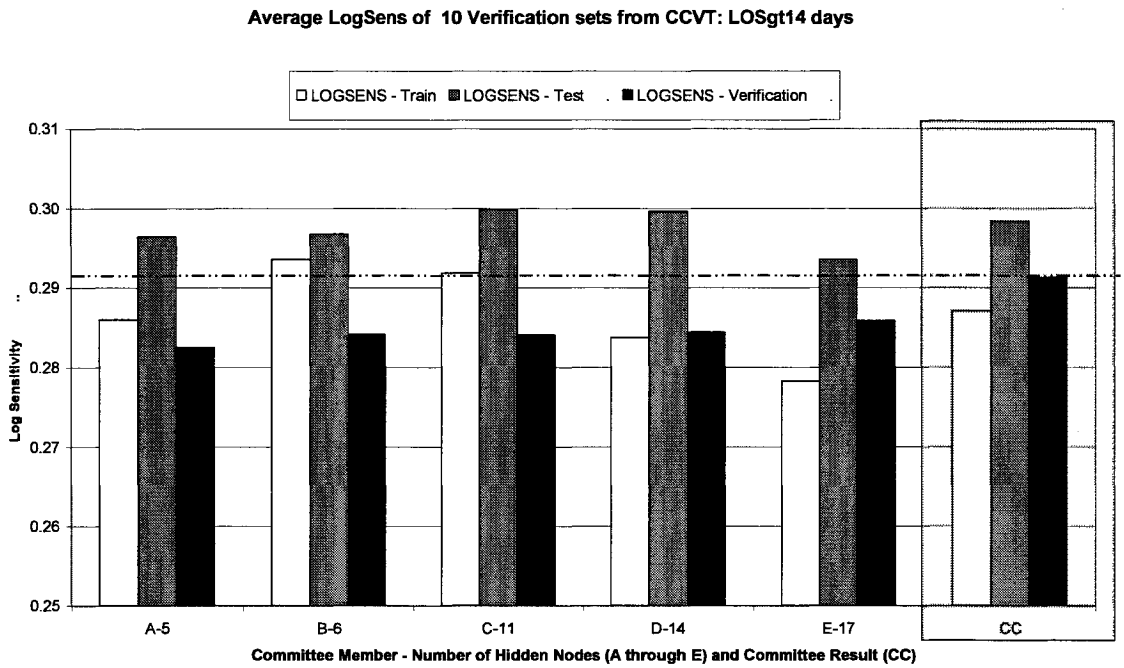

Figure F-11: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for LOSgt14 days.

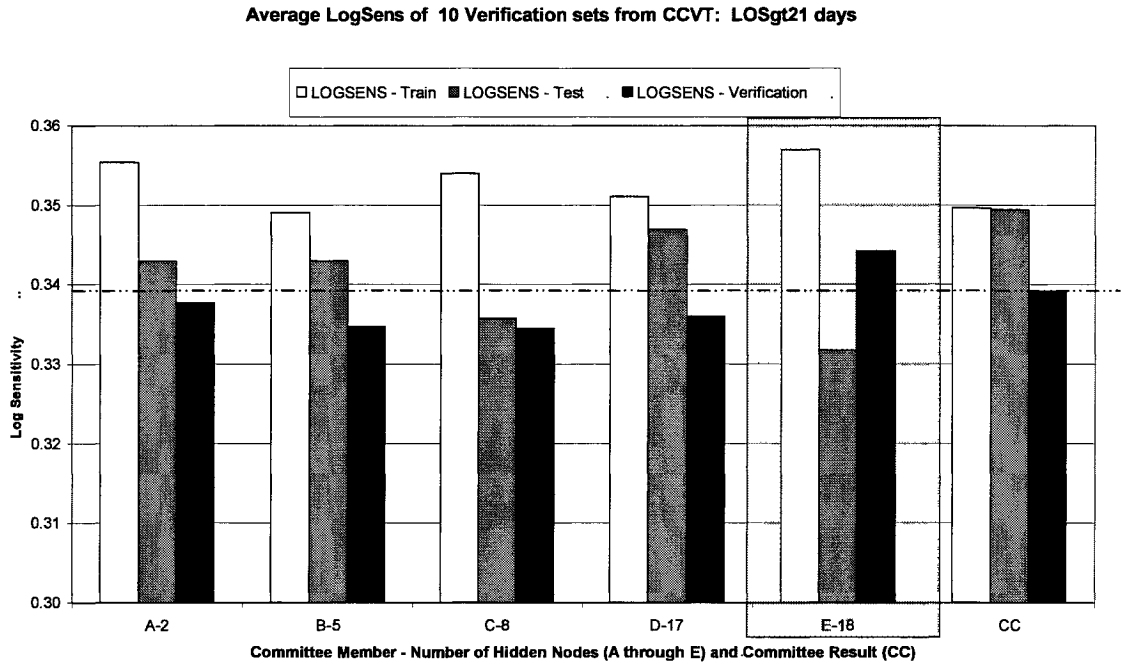

Figure F-12: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for LOSgt21 days.

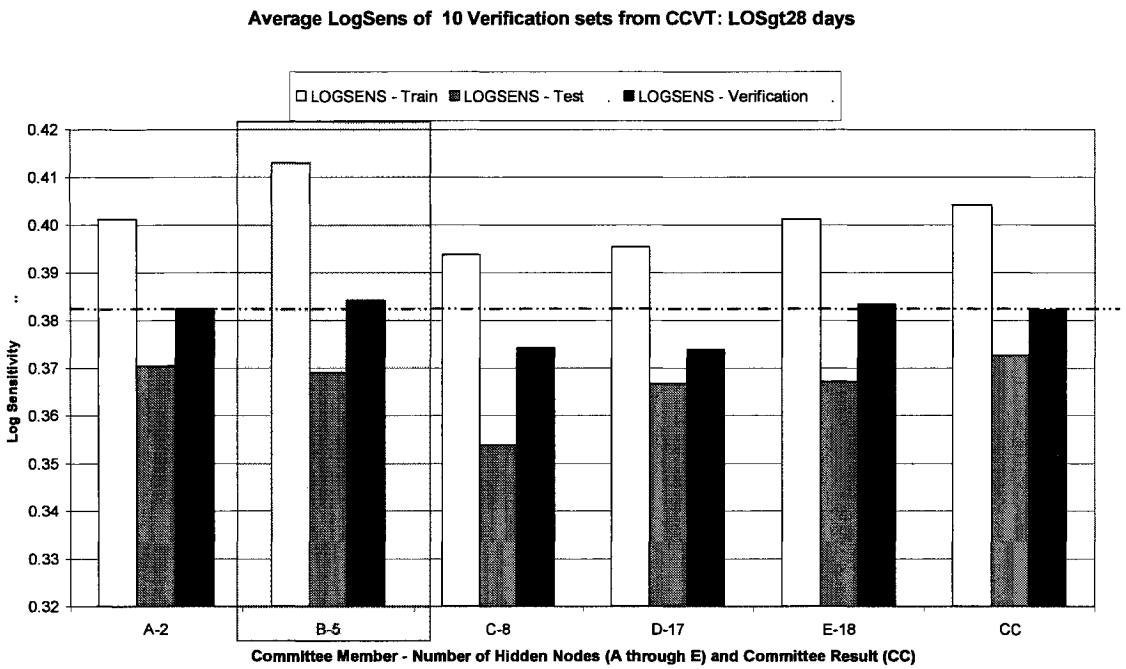

Figure F-13: Average Log Sensitivities for five committee members and committee result using the CNN Verification datasets for LOSgt28 days.