Imputation of Missing Values by Integrating Artificial Neural Networks and Case-based Reasoning

by

Colleen Michelle Ennett

M.A.Sc., Electrical Engineering, University of Ottawa, 1999

B.Sc. (Eng.), Biological Engineering, University of Guelph, 1997

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in Electrical Engineering

Department of Systems and Computer Engineering

Carleton University, Ottawa, Ontario, Canada

© Colleen Michelle Ennett, 2003

National Library
of Canada

Bibliothèque nationale
du Canada

Acquisitions and
Bibliographic Services

Acquisitions et
services bibliographiques

395 Wellington Street
Ottawa ON K1A 0N4
Canada

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

ISBN: 0-612-88717-0

Our file Notre référence

ISBN: 0-612-88717-0

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

The undersigned hereby recommend to
the Faculty of Graduate Studies and Research
acceptance of the thesis

**Imputation of Missing Values by Integrating
Artificial Neural Networks and Case-based
Reasoning**

Submitted by
Colleen Michelle Ennett

in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

Thesis Supervisor

External Examiner

Chair, Department of Systems and Computer Engineering

Carleton University

May 2003

May 2003

Carleton University

Abstract

IMPUTATION OF MISSING VALUES BY INTEGRATING ARTIFICIAL
NEURAL NETWORKS AND CASE-BASED REASONING

by Colleen Michelle Ennett

Thesis Supervisor: Dr. Monique Frize, P.Eng., O.C.

Department of Systems and Computer Engineering

The implementation of a neural network and case-based reasoning hybrid system to impute missing values was successful. The Canadian Neonatal Network database of 19427 patient cases was used to test this approach. The connection weights of a linear neural network were extracted and used as the match weights in the case-based reasoner to find the closest-matching cases. The missing values in the queries were replaced with the means of the matched cases. A neural network with the weight-elimination cost function and the log-sensitivity stopping criterion was used to develop a new neonatal mortality model that identified the most influential risk factors for predicting mortality in the neonatal intensive care unit. The resultant model classified the patients equally well or better than the statistically-based models in the literature.

To my mom, Gaye (Heartwell) Ennett, for giving me the
courage to challenge myself.

To my brother, Craig Ennett, for being my truest friend and
confidant.

In loving memory of my dad, Mike Ennett – not the kind of
doctor you were expecting, but a doctor nonetheless ☺

To my life partner and soulmate, Kevin Eade, for keeping me
focused on the important things in life.

Acknowledgments

My sincerest gratitude goes to my thesis supervisor, Dr. Monique Frize. Thank you for talking me through the roadblocks, and keeping me on track to finish my thesis. I have enjoyed the past six years as your graduate student and friend.

Many thanks to my thesis committee: Dr. Robin Walker, Dr. Rafik Goubran and Dr. David Gibbons. I appreciated your feedback and encouragement throughout the duration of my graduate studies.

I am indebted to Dawn Rybchynski for modifying the CBR to meet my needs and ensuring that the system worked properly before leaving for Inuvik. I appreciate the sweat and tears that you invested for me! I'd also like to acknowledge Neda Faregh for her assistance with the statistical aspects of my work. Thanks to Kirby Ward from the University of New Brunswick for returning my data files for the CBR in a timely fashion. Thank you to Dr. Shoo Lee from the Canadian Neonatal Network and the University of British Columbia for providing access to the Canadian Neonatal Network database for my thesis research.

I am grateful for the scholarships that I received through the Ontario Graduate Scholarship Program, Carleton University and the Canadian Engineering Memorial Foundation, as well as the additional funding from Dr. Frize.

And most importantly, I'd like to acknowledge the continuous support of my fiancé, Kevin Eade. Thanks for riding the roller coaster of thesis life with me and for your enthusiasm for my career aspirations. And a special thank you to our baby-on-the-way for providing the extra incentive to finish on time!

Contents

1	INTRODUCTION.....	1
1.1	Motivation.....	1
1.2	Problem Identification.....	2
1.3	Problem Statement.....	2
1.4	Hypotheses	3
1.5	Significance of the Research.....	3
1.6	Summary of Results	4
1.7	Thesis Outline.....	6
2	A BRIEF REVIEW OF ARTIFICIAL NEURAL NETWORKS AND CASE-BASED REASONERS.....	8
2.1	Artificial Neural Networks	8
2.1.1	Basic Principles of the Neural Network Process	12
2.1.2	Transfer Function	14
2.1.3	Weight-elimination Cost Function	14

2.1.3.1	Weight-decay Constant.....	16
2.1.3.2	Weight-decay Scale Factor	17
2.1.4	Backpropagation Algorithm.....	17
2.1.5	Adaptive Network Parameters	20
2.1.5.1	Error Ratio.....	21
2.1.5.2	Learning Rate.....	21
2.1.5.3	Momentum	22
2.1.5.4	Weight-decay Constant.....	23
2.1.6	Network Generalization	23
2.1.7	Stopping Criteria	24
2.1.7.1	Error Bounds.....	25
2.1.7.2	Early Stopping.....	25
2.1.7.3	Correct Classification Rate.....	26
2.1.7.4	Logarithmic-sensitivity Index	26
2.1.8	Estimating Rare Outcomes	27
2.2	Case-Based Reasoners	28
2.2.1	The CBR Cycle.....	29
2.2.2	Case Representation	31
2.2.3	Indexing.....	31
2.2.4	Storage.....	32
2.2.5	Case Retrieval.....	33
2.2.5.1	Inductive Clustering	33

2.2.5.2	Nearest Neighbours	33
2.2.6	Adaptation.....	34
2.2.7	The Easy Reasoner	34
2.2.7.1	The Easy Reasoner's Retrieval Mechanism	35
2.3	Hybrid Systems	36
2.3.1	Stand-alone Models	37
2.3.2	Transformational Models	38
2.3.3	Loosely-coupled Models.....	38
2.3.4	Tightly-coupled Models	39
2.3.5	Fully-integrated Models	40
2.3.6	Examples of Other ANN-CBR Hybrid Systems.....	41
2.3.7	Hybrid System Used in this Thesis.....	43
3	THE MISSING VALUE PROBLEM	44
3.1	The Reason for Missing Data	44
3.2	Outliers.....	46
3.3	Complete-case Analysis.....	46
3.4	Statistical Imputation of Missing Values.....	47
3.4.1	Mean Imputation	47
3.4.2	Multiple Draw Imputation	48
3.4.3	Random Number Generator	48
3.5	Committee of Classifiers	49

3.6	Nearest Neighbours Algorithm	49
3.7	Reduction of Viable Output Alternatives	50
3.8	Packet Loss Recovery Techniques	50
3.8.1	Insertion-based Repair	51
3.8.2	Interpolation-based Repair.....	52
3.8.3	Regeneration-based Repair.....	53
4	THE NEONATAL MEDICAL ENVIRONMENT	55
4.1	The Neonatal Intensive Care Unit	55
4.2	Neonatal Mortality Models in the Literature.....	56
4.2.1	Score for Neonatal Acute Physiology	57
4.2.2	SNAP-II.....	60
4.3	The Canadian Neonatal Network Database	61
4.3.1	Missing Values in Canadian Neonatal Network Database.....	63
5	PROBLEM FORMULATION	66
5.1	Problem Statement.....	66
5.2	Hypotheses Restated.....	67
5.3	Evidence that Problem Has Not Previously Been Solved	67
5.3.1	Limitations of Previous Imputation Approaches	67
5.3.1.1	Complete-case Analysis	68
5.3.1.2	Statistical Imputation of Missing Values	69

5.3.1.3	Committee of Classifiers	70
5.3.1.4	Nearest Neighbours Algorithm	71
5.3.1.5	Reduction of Viable Output Alternatives	71
5.3.1.6	Packet Loss Recovery Techniques	71
5.3.2	Limitations of SNAP and SNAP-II	72
5.3.3	Prior Neural Network Approaches to Predict Neonatal Mortality	74
5.4	Value of Developing a New Imputation Approach	76
5.4.1	Other Applications	77
6	METHODOLOGY.....	78
6.1	Step 1: Validation of Hybrid Imputation Approach for <i>Known</i> Missing Values	78
6.1.1	Step 1: Summary.....	79
6.1.2	Step 1: Detailed Discussion.....	80
6.1.3	Neural Network Architecture.....	84
6.1.4	Case-based Reasoning System.....	86
6.1.5	Rationale for Using a Linear Neural Network	88
6.1.6	Rationale for Using a Case-based Reasoner to Match Cases.....	88
6.1.7	Rationale for Equating Neural Network and CBR Weights.....	89
6.1.8	Measures of Performance.....	90
6.1.8.1	Contingency Table.....	91
6.1.8.2	Constant Predictor	92

6.1.8.3 Receiver Operating Characteristic Curve.....	92
6.2 Step 2: Validation of Hybrid Imputation Approach for <i>Unknown</i> Missing Values	93
6.2.1 Vertical Expansion.....	94
6.2.2 Horizontal Expansion.....	95
6.2.3 Step 2: Summary.....	96
6.2.4 Step 2: Detailed Description.....	97
6.3 Step 3: Development of New Neonatal Mortality Model	99
6.3.1 Step 3: Summary.....	100
6.3.2 Step 3: Detailed Description.....	101
6.3.2.1 Variable-wise Horizontal Expansion	103
6.3.2.2 Step-wise Horizontal Expansion	104
7 SIMULATION RESULTS AND EVALUATION	107
7.1 Step 1 Results: Validation of the Hybrid Imputation Approach.....	107
7.1.1 Weights with the Complete-case Database.....	107
7.1.2 Development of the Artificial Database with Missing Values	109
7.1.3 Calculation of Mean and Random Values.....	110
7.1.4 Imputation Error for Known Values.....	112
7.1.5 Neural Network Results with Imputed Values in Artificial Data Sets	119
7.2 Step 2 Results: Vertical Expansion of the Database.....	120

7.3	Step 3 Results: Horizontal Expansion of the Database and Development of New Neonatal Mortality Model.....	124
7.3.1	Creation of the Initial Expanded Data Sets	124
7.3.2	Weight Extraction for CBR Imputation of Missing Values	126
7.3.3	Development of New Mortality Model.....	129
7.3.4	Comparison with Other Models in the Literature.....	132
7.3.4.1	SNAP Models.....	132
7.3.4.2	Very Low Birth Weight Model.....	134
7.3.4.3	Extremely Low Birth Weight Model	135
7.3.4.4	MASc Thesis Work by Tong [2000]	135
7.4	General Discussion of Results	136
7.4.1	Prediction for Groups Versus Individuals	136
7.4.2	Neural Network Performance for SNAPPE-II Variables Versus Hybrid Model	137
8	CONCLUDING REMARKS	138
8.1	Conclusions	138
8.2	Academic Contributions to New Knowledge.....	142
8.3	Future Research.....	143

List of Tables

2-1. Comparison of KBS and CBR systems for problem solving.....	29
3-1. Types of insertion-based repair.....	52
3-2. Types of interpolation-based repair	53
3-3. Types of regeneration-based repair.....	54
4-1. Definitions of SNAP point assignment	58
4-2. SNAP variable list and scoring system	59
4-3. SNAP-II variable list and scoring system	60
4-4. Perinatal risks and patient profile	63
4-5. Missing values in the Canadian Neonatal Network database.....	65
6-1. Missing value statistics for SNAPPE-II variables	81
6-2. Truth table	91
6-3. Division of database for horizontal expansion.....	102
7-1. Recalibrated mean connection weights from Step 1 ANN experiments.....	108
7-2. Distribution of missing values in artificial Step 1 query data set, SNAPPE-II variables.....	110

7-3. Average values for Step 1 test sets with imputed values in artificial missing value database.....	112
7-4. Percent error for Step 1 imputed values in artificial missing values database.....	114
7-5. Range containing 95% of data for variables with missing values.....	115
7-6. Percent error for Step 1 hybrid-imputed values separated according to the number of missing values in each case.....	117
7-7. Step 1 test set results of linear ANN experiments with imputed data ($N = 5102$); training stopped at epoch 846.....	119
7-8. Step 1 test set results of linear ANN experiments with imputed data ($N = 5102$); best test set performance.....	119
7-9. Mean values of Step 2 databases.....	122
7-10. Step 2 test set results of linear ANN experiments with imputed data for SNAPPE-II variables.....	123
7-11. Missing values in the SNAPPE-II database.....	125
7-12. Normalized Step 3 ANN weights for CBR match weights.....	127
7-13. Progression of variables deleted for development of new neonatal mortality model.....	130
7-14. Step 3 test set results for 13 input variables – new Neonatal Mortality Model..	131
7-15. List of risk factors for predicting neonatal mortality.....	133
A-1. Apgar scoring system.....	161
A-2. CRIB scoring system.....	162
A-3. NTISS scoring system.....	164

D-1. SPSS variable name and transformation formulae for SNAP variables	186
D-2. Collinearity and cross-validation table for SNAPPE-II variables.....	187
E-1. Default network settings for Step 1 experiments to extract ANN weights for use as CBR match weights	189
E-2. Optimal network settings for Step 2 experiments for imputed data	189
E-3. Optimal network settings for Step 3 experiments for horizontal expansion	190
E-4. Optimal network settings for Step 3 experiments for variable reduction following horizontal expansion.....	190
E-5. Optimal network settings for Step 3 experiments for new Neonatal Mortality Model with 13 inputs	191
F-1. Statistics for <i>lbloodp</i> data	193
F-2. Paired samples correlations for <i>lbloodp</i> data.....	193
F-3. Paired samples <i>t</i> -test results for <i>lbloodp</i> data	193
F-4. Statistics for <i>lserum</i> data	194
F-5. Paired samples correlations for <i>lserum</i> data.....	194
F-6. Paired samples <i>t</i> -test results for <i>lserum</i> data	194
F-7. Statistics for <i>lurine</i> data	195
F-8. Paired samples correlations for <i>lurine</i> data.....	195
F-9. Paired samples <i>t</i> -test results for <i>lurine</i> data	195
F-10. Statistics for <i>po2fio2r</i> data	196
F-11. Paired samples correlations for <i>po2fio2r</i> data.....	196
F-12. Paired samples <i>t</i> -test results for <i>po2fio2r</i> data	196

List of Figures

2-1. Architecture of a three-layer feedforward neural network	9
2-2. Diagrammatic representation of an artificial neuron	12
2-3. Pseudocode for neural network parameters	20
2-4. The CBR cycle.....	30
6-1. Screen capture of the IDEAS for NICUs (Research Version) case-based reasoner	87
6-2. Expansion of imputation approach for the NICU database.....	94
6-3. Distribution of missing values in entire database for SNAPPE-II variables	98
6-4. Variable-wise division of database for horizontal expansion	103
6-5. Step-wise division of database for horizontal expansion	105
7-1. Syntax for random imputation of missing data	111

List of Appendices

A: Neonatal Scoring Systems	160
B: SNAP Guidelines	164
C: Technical Report for Neonatal Intensive Care Unit Database: Canadian Neonatal Network	177
D: Derivation and Statistics of SNAP Variables	185
E: Neural Network Parameter Settings	188
F: Step 1 Statistical Analysis.....	192
G: List of Publications.....	197

Glossary

Apgar score. An additive scoring system usually taken 1, 5 and 10 minutes after birth to assess risk of complication or bad outcome. APGAR = Appearance, Pulse, Grimace, Activity, Respiration. Score is out of 10 points.

Artificial neural network. An artificial intelligence tool that identifies patterns in databases to predict outcomes. Two-layer networks (i.e. input and output layers) are linear. Networks with three or more layers (i.e. one or more hidden layers) are nonlinear.

Case-based reasoner. An artificial intelligence tool that uses past case histories to predict future outcomes and match similar cases. Particularly useful when working with text fields in databases. Most commonly used for retrieval of similar cases to the query. Complicated and knowledge-intensive updating procedures make it difficult to completely automate the system.

MIRG. Medical IDEAS (Intelligent Decision Aid Systems) Research Group located in Ottawa, ON. Principle investigators: Dr. Monique Frize, P.Eng., O.C. (Carleton University and University of Ottawa) and Dr. C. Robin Walker, MBBS, FRCPC (Children's Hospital of Eastern Ontario and University of Ottawa).

Nearest neighbours algorithm. An algorithm that calculates the distance between two points in an n -dimensional space. The k -nearest neighbours algorithm finds the k closest-matching cases to the query. A weighted-nearest neighbours algorithm allows the system parameters to be weighted differently as opposed to uniform weighting. Generally used for pattern classification.

Neonatal death. Defined by the World Health Organization as death occurring before 28 days of age.

SNAP. Score for Acute Neonatal Physiology; contains 37 input variables and is validated for predicting mortality and is highly correlated with nursing workload, length of hospital stay and therapeutic intensity in the Neonatal Intensive Care Unit (NICU).

SNAPPE. SNAP with Perinatal Extension; uses the 37 SNAP variables plus birth weight, small for gestational age and Apgar score at 5 minutes.

SNAP-II. SNAP Version 2; a parsimonious version of SNAP with only six input variables (of the original 37 inputs) and is validated for predicting mortality, severe intraventricular hemorrhage and chronic lung disease in the NICU.

SNAPPE-II. SNAP-II with Perinatal Extension; uses the six SNAP-II variables plus birth weight, small for gestational age and Apgar score at 5 minutes.

Nomenclature

ANN. Artificial neural network

APACHE. Acute physiology and chronic health evaluation

CBR. Case-based reasoner/reasoning

CRIB. Clinical risk index for babies

GFMM. General fuzzy min-max neural network

GSLN. Generalized single-layer networks

ICU. Intensive care unit

IFOS-DMR. Iterative fast orthogonal search with dynamic model resizing

NICU. Neonatal intensive care unit

NTISS. Neonatal therapeutic intervention scoring system

ROC. Receiver operating characteristic curve

SGA. Small for gestational age ($< 5^{\text{th}}$ percentile)

SNAP. Score for neonatal acute physiology

SSE. Sum of squared errors

TISS. Therapeutic intervention scoring system

1 Introduction

1.1 Motivation

The human mind has an extraordinary ability to draw inferences and conclusions when presented with incomplete information about a particular problem. The ability to learn patterns, derive information from past experiences and draw conclusions based on minimal evidence is the “holy grail” of artificial intelligence. Artificial intelligence systems attempt to achieve these goals by focusing on specific aspects of the human mind. Some of the artificial neural network’s objectives are to learn by observation of past cases, discover patterns, extract features and predict events based on this previously acquired knowledge. A case-based reasoner looks at different cases to determine which are the most similar. Often these systems are implemented independently. There has been interest in combining these artificial intelligence approaches to profit from the strengths of some systems that may overcome the limitations of others. Artificial neural networks usually have difficulty dealing with missing values, however, case-based reasoners are able to match cases despite missing information. There exists a potential advantage of integrating these two systems to

impute missing values such that useful information can be extracted from incomplete databases.

1.2 Problem Identification

The issue of missing values in databases is common to all applications. In theory, the best way to deal with missing values is to avoid them – more easily said than done. Missing data may be beyond the control of those collecting the data or those using the data for analysis. Current methods of working with missing values are unsatisfactory. Either valuable information is discarded or values based on the entire data set are imputed which may be inappropriate. A new approach that attempts to incorporate specific information about the characteristics of the cases with missing values is essential to impute relevant values.

As an example, the neonatal intensive care unit (NICU) mortality scoring systems that are currently in clinical use do not attempt to estimate the missing values in the database. Either the cases with missing values are deleted or the missing data is simply coded as normal and otherwise ignored. This may not be the best approach for dealing with missing data, so a new approach may be beneficial.

1.3 Problem Statement

The objective of this thesis was to impute missing values by integrating artificial neural networks (ANNs) and case-based reasoning (CBR). Using the weights of a linear ANN

as the matching weights of the CBR to find the closest-matching cases, the missing values in the NICU database were replaced with the means of the matched cases. The database used to test this approach was the Canadian Neonatal Network of seventeen Canadian NICUs. This database with imputed values was used to develop a new neonatal mortality model. A neural network with weight-elimination identified the most important input variables to predict neonatal mortality in the NICU, and this subset of risk factors became a new neonatal mortality model.

1.4 Hypotheses

Part 1. The integration of the artificial neural network with the case-based reasoning system for imputing missing values will result in values that are as close as or closer to the actual values than those found by mean or random approaches.

Part 2. After confidence in the missing value imputation approach is established, the resultant new neonatal mortality model developed using an artificial neural network with weight-elimination will predict mortality in the neonatal intensive care unit equally well or better than other neonatal mortality prediction models in the literature.

1.5 Significance of the Research

This thesis marked the first attempt to create a neural network and CBR hybrid system to impute missing data. The hybrid approach was flexible in that no prior statistical

knowledge was required of the data with missing values. Combining two artificial intelligence systems in such a unique way was an important step in the advancement of the knowledge and applications of these systems.

With the imputed database of neonatal patients, identifying the parameters that the neural network deemed important provided an interesting comparison with the significant parameters found via statistical techniques. This information led to a new parsimonious mortality model for the NICU. It was anticipated that the hybrid model might select variables for its mortality prediction model that may be different than those chosen using statistics.

1.6 Summary of Results

The objective of this thesis was met through the implementation of a neural network and case-based reasoning hybrid system that replaced the missing values in the NICU database with the mean of the ten closest-matching cases in the matched set of the CBR. The connection weights of the linear ANN were successfully used as the match weights in the CBR to impute relevant values. A new neonatal mortality model was developed using the database with imputed values following network pruning using the neural network with weight-elimination to identify the most influential risk factors for predicting mortality. At the point where the ANN performance degraded, thirteen variables remained.

The hypotheses were also confirmed. Part 1 of the hypothesis stated that the hybrid system would impute values that were as close as or closer to the true values than those found by mean or random imputation. This goal was achieved because the *t*-test showed that the values imputed by the hybrid approaches or the mean of only the complete cases were not significantly different from the actual values, whereas the values imputed using the random approach or the mean of the entire database were most often significantly different from the true values. The hybrid system models with the imputed values had the greatest areas under the ROC (receiver operating characteristic) curves and the highest sensitivity and specificity rates compared to the models developed with mean and random imputation. In addition, the model developed using the ANN weights as the match weights for the CBR achieved the best area under the ROC curve and had good sensitivity and specificity rates compared to the hybrid model developed using the clinician-advised weights. Both of these models outperformed the hybrid model that used uniform weights as the match weights in the CBR.

Part 2 of the hypothesis stated that a new neonatal mortality model would be developed using the ANN-CBR hybrid system and that this model would predict mortality in the NICU equally well or better than other models in the literature. This thesis work resulted in the development of a new model using the hybrid system that included thirteen input variables for predicting mortality. The second hypothesis was confirmed because the mortality prediction ability of this hybrid model was better than

the SNAP (Score for Neonatal Acute Physiology) and SNAP-II (SNAP, version 2) models, but was similar to models that included the three perinatal extension variables (birth weight, small for gestational age status and the Apgar score at 5 minutes) such as SNAPPE (SNAP with the perinatal extension variables) and SNAPPE-II (SNAPPE, version 2), because the hybrid model also used these parameters. The hybrid system identified a new subset of known risk factors for mortality prediction in the NICU.

1.7 Thesis Outline

The format of the thesis provides a literature review of ANNs and CBRs in Chapter 2 to ensure a thorough understanding of the core artificial intelligence concepts used in this thesis. The issue of missing values in databases is addressed in Chapter 3 where current techniques for dealing with missing data are described and evaluated. Chapter 4 provides a detailed description of the neonatal medical environment and the NICU database. The current knowledge of medical informatics and the development of mortality risk scores for the NICU is introduced, in particular, focusing on the Score for Neonatal Acute Physiology (SNAP score). The problem under investigation is presented in Chapter 5 along with justification of the problem and a discussion of its validity to add to knowledge. The approach for replacing missing data by combining different strengths of ANNs and CBRs is outlined in Chapter 6, along with the steps that were taken to validate the missing value imputation approach and how a new neonatal mortality model was created. Chapter 7 summarizes the results and presents an analysis and discussion of the findings of this thesis. And finally, the concluding

remarks are made in Chapter 8 where the contributions to academic knowledge are outlined and ideas for future work are presented.

2 A Brief Review of Artificial Neural Networks and Case-based Reasoners

2.1 Artificial Neural Networks

An artificial neural network (ANN) is a parallel, distributed, adaptive system that processes information in order to learn patterns as a result of exposure to a set of representative training cases [Penny & Frost 1996]. The objective of a neural network is to mimic human brain function to solve problems. The imitation goes as far as simulating the basic structure of a human brain. The ANN has simple processing units called nodes that are comparable to the neural processing units in the brain. The artificial processing units accept inputs of information about the current problem (like the receiving ends of the synapse). These inputs and a bias term (which is equal to one) are multiplied by weights (similar to synaptic connections), summed, and then fed through a transfer function to scale the output within a limited range defined by the transfer function. The role of the bias input is analogous to a nonzero intercept in statistics [Kattan & Beck 1995]. The bias weight provides a threshold above which the

Figure 2-1. Architecture of a three-layer feedforward neural network

node is activated [Penny & Frost 1996]. The node's output becomes either the input of the next layer or the network output.

The nodes in a network are connected in parallel to create layers as shown in Fig. 2-1. A feed-forward network allows information to travel directly from the inputs to the outputs. If any outputs are fed back as inputs to previous layers, this is a feedback or recurrent network [Fausett 1994]. In a fully-connected network, each node is connected to every node in the layer below it. Hidden layers allow more complex problems to be solved using non-linear algorithms, however, not all networks require a hidden layer. Networks with no hidden layers are inherently linear and thus can only solve linear problems (two-layer networks). They are similar to a generalized linear model in statistics. Training time increases with more hidden layers because there are more parameters to optimize. No substantial evidence has proven that more than one hidden layer (i.e. a three-layer network) improves the predictive performance of a network [Penny & Frost 1996].

Several factors must be considered when choosing the number of nodes in the hidden layer. The more nodes a network has, the more closely it can model the dataset. However, a model that is too closely related to the training set is unable to generalize to accommodate new data [Masters 1993]. Too many nodes may be inefficient, since training time also increases proportionally to the number of nodes in the hidden layer. On the other hand, too few nodes will not provide sufficient resources to solve the problem, resulting in underfitting. Underfitting means that the network did not learn enough about the dataset to correctly classify cases. If a network has sufficient hidden nodes, one hidden layer is adequate for describing almost any function to any desired degree of accuracy, as long as the function and its derivative are continuous [Hornik et al. 1989, Fausett 1994, Penny & Frost 1996].

A pruning method can reduce the network size to prevent overfitting. Training time increases exponentially with the number of inputs and nodes, and polynomially with the number of training cases. Pruning the network to include only the most pertinent information is essential [Penny & Frost 1996].

Network training can be accomplished using either supervised or unsupervised learning. Supervised learning means that the true output is known and the network uses this information to correct its errors and learn by example. This approach is most useful for database classification, function approximation, pattern recognition and data compression [Fausett 1994]. The true output is unknown for unsupervised training. This self-organizing technique must independently select which features it will use to

roughly classify the inputs. At this point, unsupervised training is not well understood [Anderson & McNeill 1992]. Each cycle through the training patterns during network training is called an epoch or iteration.

The main advantage of neural networks over statistical techniques is that the model does not have to be explicitly defined before beginning the experiments to develop the model. ANNs are able to recognize the relevant data and patterns, whereas a statistical model requires prior knowledge of the relationships between the factors under investigation [Blum 1992, Livingstone et al. 1997]. Also, with statistics, it is difficult to integrate data of different formats (i.e. working simultaneously with continuous, binary, ordinal and nominal data). Research has found that ANNs are useful for modeling nonlinear data [Penny & Frost 1996]. Medical data is often believed to have nonlinear relationships between the input variables and the outcome, so neural networks are a logical choice for analysis in the medical environment [Baxt 1994].

The backpropagation neural network has been compared to various other modelling techniques in a variety of areas of research (medicine, telecommunications, etc.) including logistic regression [i.e. Ambalavanan & Carlo 2001], radial basis functions [i.e. Abeyratne et al. 2001], quasi-Newton neural networks [i.e. Lin et al. 1997], and generalized single-layer networks (GSLNs) trained using iterative fast orthogonal search with dynamic model resizing (IFOS-DMR) [Adeney et al. 1999]. In each case, the backpropagation ANN has been found to have similar or better classification performance compared to the other approaches.

2.1.1 Basic Principles of the Neural Network Process

The code for the ANN was implemented in MATLAB (version 6.5). Fig. 2-2 shows a diagrammatic representation of an artificial neuron to help explain how the neural network works. The first step is to randomly initialize the initial weights w_i for the inputs p_i and bias term b_j . Vector W_1 , representing the weights of the inputs of the first layer (the input-to-output connections for a two-layer network or the input-to-hidden layer connections for a three-layer network), was initialized between -1 and 1 . Vector W_2 is only present for a three-layer network. It represents the weights for the hidden-

Figure 2-2. Diagrammatic representation of an artificial neuron

to-output connections that were randomly assigned values between -0.1 and 0.1. The initial weights for the bias terms were assigned using the same approach such that vector B_1 had random values ranging from -1 to 1 (output layer for two-layer network, hidden layer for three-layer network) and vector B_2 ranged between -0.1 and 0.1 (output layer of three-layer network). The bias vectors are actually unit vectors. The transfer function of each node was the hyperbolic tangent. Given this information, the output was calculated according to (2-1), where n corresponds to the n -th layer, A_n is the output of the node of the n -th layer and p is the number of inputs to the current node [Demuth & Beale 1997, Scales 2001].

$$A_n = \tanh\left(\sum_{i=1}^p W_n[i]A_{n-1}[i] + b_n[i]\right), n \geq 1 \quad (2-1)$$

The patient data are represented in the system by A_0 , which is a matrix whose rows consist of patient cases and whose columns contain variables regarding the patient's medical information and the outcomes under investigation. The system output (vector A_1 for a two-layer network and vector A_2 for a three-layer network) represents the network's prediction of a given patient's outcome (i.e. survival) ranging from -1 to 1. The output is compared with the true output (i.e. -1 for a survivor, 1 for a nonsurvivor) and the error E_0 is the difference between these two values. The cut-off value for the network was typically 0, such that a value in the output vector that was greater than 0 classified the patient as a nonsurvivor and a patient with a value less than 0 was classified as a survivor [Demuth & Beale 1997, Scales 2001].

2.1.2 Transfer Function

The hyperbolic tangent transfer function was used in these experiments because of its sharp transition between its output values that permits faster learning. Like the logarithmic transfer function, the hyperbolic tangent (and its derivative) is continuous, differentiable and monotonically nondecreasing, which simplifies the training algorithm [Fausett 1994]. The nonlinearity of this function means that after a certain point of stimulation, the output becomes saturated and is no longer affected [Itchhaporia et al. 1996]. The hyperbolic tangent (ranging from -1 to 1) offers a sharp sigmoidal transition at zero which results in very few output values near zero, so a nonzero output is more likely [Fausett 1994]. Weights with a zero value do not impact the neural network during the learning process because the resultant weight change in the error term would also be zero. Using an output range of -1 to 1 with inputs scaled to have zero mean and unit variance allows the network to learn faster [Penny & Frost 1996].

2.1.3 Weight-elimination Cost Function

Weight-elimination and weight-decay add a complexity term to the cost function to prevent overfitting to the training set [Weigend et al. 1990, Krogh & Hertz 1992]. Weight-decay limits how large the connection weights are allowed to grow. The resulting output has less variance and is therefore a more stable network. On the other hand, weight-elimination focuses on the small weights by trying to reduce them to zero. The small weights only add unwanted “white noise” to the model. This network

pruning approach simplifies the network and can eliminate variables that offer little or no assistance in predicting the correct outcome [Weigend et al. 1990, 1991, Ennett & Frize 1998, Frize & Ennett 2000, Frize et al. 2001a, 2001b].

In a two-layer network, an input variable is eliminated when its connection weight is zero. The situation is more complex when dealing with a hidden layer. In order for an input variable to be deleted when the network has a hidden layer, all of the connection weights connected to that input must be small. Therefore, eliminating variables is easier to accomplish when there are fewer nodes in the hidden layer and thus fewer weights to deal with. The likelihood of all weights going to zero when there are numerous hidden nodes is low [Ennett 1999, Scales 2001]. These cost functions work best when the initial conditions involve a large network structure, relatively small weights and a relatively small learning rate [Weigend et al. 1990]. Here, the qualifier “relatively” is determined *ad hoc* through preliminary experiments to assess the influence of the initial random weights and initial learning rate on the network performance.

The weight-elimination cost function can be used in conjunction with the backpropagation algorithm. Eq. (2-2) shows that the combined error function $E(W)$ includes the initial cost function (in these experiments, this is the sum of squared errors cost function where $E_o(W)$ is the difference between the target and network output) and the weight-elimination term (the second term in (2-2)). W represents the weight vector, λ is the weight-decay constant, w_j indicates the individual weights of the

ANN and w_0 is the weight-decay scaling parameter. When added to the sum of squared errors cost function, the weight-elimination penalty term focuses on simultaneously minimizing the performance error while reducing the number of connection weights [Weigend et al. 1991].

$$E(W) = \sum_{i=1}^p (E_0(W))^2 + \lambda \sum_{ij} \frac{\frac{w_{ij}^2}{w_0^2}}{1 + \frac{w_{ij}^2}{w_0^2}} \quad (2-2)$$

A backpropagation network with the weight-elimination cost function offers several benefits over the standard backpropagation technique. First, it acts as a variable-reduction method by forcing already small weights to zero, hence eliminating the influence of those variables on the ANN model. This reduces the complexity of the model, which could improve the model's ability to generalize for new data. Secondly, the largest weights on the remaining variables have been shown to affect the outcome [Weigend et al. 1991, Trigg 1997, Frize et al. 1997, 1998]. Frize et al. [1998] found that reducing the input variable set to just the largest weights enhanced the performance of the ANN.

2.1.3.1 Weight-decay Constant

The weight-decay constant (λ , λ) determines the relative importance of the weight-elimination term in (2-2) which translates to how strongly weights are penalized. The larger the value of the weight-decay constant, the closer the weights

must be to zero to be considered among the “noise” distribution. Larger weight-decay constants also put more pressure on smaller weights to go to zero. If the value is too small, there will be no impact on the network; however, if it’s too large, all weights will be forced to zero. Weigend et al. [1990] recommend initially setting the weight-decay constant to zero to allow the network to gather information for all input variables, then gradually increase its value until the performance begins to degrade. At that point, the weight-decay constant should be fine tuned to optimize the performance.

2.1.3.2 Weight-decay Scale Factor

The weight-decay scale factor w_0 determines the size of the small weights that will be eliminated with the weight-elimination cost function. A small value for the scale factor forces small weights to zero resulting in fewer large weights (i.e. weight-elimination). A large scale factor allows the small weights to remain, but limits the size of the large weights (i.e. weight-decay) [Weigend et al. 1991].

2.1.4 Backpropagation Algorithm

To permit learning, the weights need to be individually adapted to reduce the network’s prediction error. Error backpropagation is among the most commonly used network training algorithm (also known as the generalized delta rule) [Werbos 1974, Rumelhart et al. 1986a, 1986b]. Similar to the steepest descent algorithm, backpropagation uses the error between the target (true output) and system output to modify the weights of the network. The error is propagated back through the layers,

adjusting the node weights to minimize the overall error or to reinforce the important weights [Livingstone et al. 1997].

The following description outlines how backpropagation with weight-elimination works for a three-layer neural network. The simplicity of a two-layer network means that backpropagation is not necessary since there is only one set of weights between the inputs and the output, but weight-elimination can still play a role in changing the node weights [Ennett 1999].

Define D_2 as proportional to the rate of change of the network output A_2 with respect to the error E and D_1 as proportional to the rate of change of the output of the first layer A_1 with respect to D_2 :

$$D_2 = \frac{dA_2}{dE} \bullet E = \left[\frac{d \tanh(E)}{dE} \right] \bullet E \quad (2-3)$$

$$D_1 = \frac{dA_1}{dD_2} \bullet W_2 \bullet D_2 = \left[\frac{d \tanh(D_2)}{dD_2} \right] \bullet W_2 \bullet D_2 \quad (2-4)$$

The change in weights can be calculated as follows:

$$W = W + dW \quad (2-5)$$

where dW can be found from:

$$dW_n = m * dW_n + (1 - m) * lr * D_n * A_{n-1}^T - 2\lambda \left[\frac{w_0^2 W_{n_y} \left(1 + \frac{W_{n_y}^2}{w_0^2} \right) - W_{n_y}^3}{w_0^4 \left(1 + \frac{W_{n_y}^2}{w_0^2} \right)^2} \right] \quad (2-6)$$

Eq. (2-6) represents the rate of change of the weights with respect to time. The first term in the equation includes the momentum term m that will help prevent the network from getting caught in local minima. The second term is proportional to the rate of change of error with respect to the output D_n and includes the learning rate term lr .

Similarly, the bias vectors can be updated by:

$$B = B + dB \quad (2-7)$$

where dB can be found from:

$$dB_n = m * dB_n + 1 - 2\lambda \left[\frac{w_0^2 B_{n_y} \left(1 + \frac{B_{n_y}^2}{w_0^2} \right) - B_{n_y}^3}{w_0^4 \left(1 + \frac{B_{n_y}^2}{w_0^2} \right)^2} \right] \quad (2-8)$$

After each training epoch, (2-6) and (2-8) update the weight and bias vectors. To monitor the classification performance of the test set, the measures of performance are calculated for the test set after each training epoch as well.

The main drawback of the backpropagation algorithm is the possibility of finding a local minimum rather than the global minimum [Rumelhart et al. 1986b]. Training the network several times with different initial weight settings may overcome this problem [Penny & Frost 1996].

2.1.5 Adaptive Network Parameters

Adjustable parameters attempt to fine tune network training to develop the best ANN model. The user defined error ratio *err_ratio* determines how the learning rate *lr*, momentum *m* and the weight decay constant λ (or *lambda*) are updated. These parameters may be incremented or decremented depending on the most recent change in error. The change in the learning rate is controlled by *lr_inc* and *lr_dec*. Similarly, *lambda_inc* and *lambda_dec* are responsible for the appropriate changes in the weight decay constant. The network's performance is very sensitive to these parameter settings. These parameters interact as shown in the pseudocode in Fig. 2-3. *TSSE* refers to the training set error from the current epoch; *SSE* is the error from the

```

if TSSE>SSE*err_ratio
    lr=lr*lr_dec;
    lambda=lambda*lambda_dec;
    m=0;
else if TSSE<SSE*err_ratio
    lr=lr*lr_inc;
    lambda=lambda*lambda_inc;
end

```

Figure 2-3. Pseudocode for neural network parameters

previous epoch. The following sections describe the interactions shown in Fig. 2-3.

2.1.5.1 Error Ratio

The error ratio *err_ratio* is a user-defined value that measures the change from the current to previous error to determine how the learning rate, weight decay constant and momentum will be adapted to improve the network performance. It acts like a cutpoint to decide what the next course of action is. The objective is to have the error ratio get smaller with each iteration to show that the performance is improving. Typical values range from 1.03 to 1.15 to allow for some minor lapses in performance.

2.1.5.2 Learning Rate

The learning rate determines the size of the steps across the decision surface. It requires adjustments during the training process to optimize the model as it develops. After computing the new weights and biases at each epoch using the current learning rate, new outputs and errors are evaluated.

When the current-to-previous iteration error is less than the error ratio, the new weights and biases are retained. The learning rate is multiplied by the learning rate increment factor to speed up learning because the algorithm is heading in the right direction to minimize the model's error. When the iteration error ratio is greater than the user-defined ratio, the new weights and biases are rejected and multiplying the learning rate by its decrement factor slows learning. In this case, learning proceeds more slowly to creep closer to the global minimum of the decision surface without

skipping over it. Larger steps translate to faster learning; smaller steps mean slower learning. Care must be taken to ensure that the learning rate is neither too large (causing instability and oscillations) nor too small (resulting in slow convergence) [Demuth & Beale 1997].

2.1.5.3 Momentum

Just like a low-pass filter disregards low-level noise, momentum follows the general flow of the error surface by ignoring any small features. This attribute prevents the algorithm from getting caught in local minima and allows it to continue its search for the global minimum.

Momentum is implemented by adding a fraction of the last weight change to the new weight change. Refer to (2-6) and (2-8) for the updated weights and bias of the weight-elimination network. The impact of the last weight change is controlled by the momentum term, which ranges from 0 to 1. If the current-to-previous iteration error is greater than the user-defined error ratio, then the momentum is set to zero and the new weights and biases are rejected as shown in Fig. 2-3. Zero means that the newest weight change is based entirely on the change in gradient. Otherwise the new weights and biases are retained and the weight change is calculated as described above. When momentum is one, the updated weight change is equal to the previous weight change and the gradient is ignored [Demuth & Beale 1997]. Eqs. (2-9) and (2-10) demonstrates the above concepts for the weight and bias changes of the simple neural network (without weight-elimination).

$$dW_n = m * dW_n + (1 - m) * lr * D_n * A_{n-1}^T \quad (2-9)$$

$$dB_n = m * dB_n + (1 - m) * lr * D_n * B_{n-1}^T \quad (2-10)$$

2.1.5.4 Weight-decay Constant

The weight-decay constant (λ , λ) is concerned with how strongly weights are penalized. Similar to the learning rate, when the iteration error ratio is larger than the predefined ratio set by the user, the weight-decay constant is multiplied by its decrement factor. The goal is to back off the pressure on the weights to be small in order to continue moving in the direction of the gradient. The “white noise” from the small weights may possibly be interfering with the training of the network, so by forcing more weights to go to zero, their impact is reduced. Conversely, when the error ratio is smaller, the weight-decay constant is multiplied by its increment factor to be more lenient on the small weights to ensure that they are adequately represented in the network. This also prevents weights from being deleted that may grow to be important as training progresses.

2.1.6 Network Generalization

The purpose of developing ANN models is to apply the model to an unseen dataset and achieve a similar level of performance as that of the training set. The ability to generalize to a test set is dependent on a good balance between the complexity of the model and the richness of the training set data. Adding more layers, input variables or

nodes increases the complexity of the system, but also increases the possibility of overfitting the model to the training set. At present, no well-defined guidelines for selecting the appropriate ANN architecture exist, so these factors are chosen *ad hoc* [Itchhaporia et al. 1996].

There are two types of generalization: interpolation and extrapolation. In the case of interpolation, the new case is similar to cases in the training set, so using information from this database to predict an outcome is acceptable. Extrapolation occurs when the test pattern is outside the range of the training data or within a large “hole” in the training set. In this situation, the extrapolated outputs are unreliable because the model uses “rules” that do not necessarily apply to the test pattern. To be a representative sample of the population of interest, the training set should be sufficiently large to prevent the need for extrapolation [CVOR 1997]. Due to the high cost of data collection and the health care system’s limited budget, medical databases rarely contain patient cases that cover all possible situations. To determine when to stop training to achieve the best generalization, we use a stopping criterion to measure the network’s performance.

2.1.7 Stopping Criteria

There are several criteria used to identify the optimal performance of a network. Two commonly used criteria are setting bounds on the change in error and early stopping based on the test set change in error. As well, the best correct classification rate of the test set can be used as a stopping criterion. The Medical IDEAS (Intelligent Decision

Aid Systems) Research Group (MIRG) in Ottawa, ON has developed another criterion to work with automated networks that limit the requirement for user supervision and works well when the outcome has a low rate of occurrence: the logarithmic-sensitivity index [Scales 2001, Ennett et al. 2002, 2003].

2.1.7.1 Error Bounds

When you set error bounds, the objective is to stop training after the change in error is less than a predetermined value. The advantage is that the stopping criterion does not depend on the test set, however, at this point the network may have overfitted to the training set. Overfitting (or memorization) reduces the model's ability to generalize to new data.

2.1.7.2 Early Stopping

The early stopping approach requires at least two subsets of a database: a training set and a test set. This technique monitors the test set error rate and stops training when the test set error rate begins to rise. Some researchers have found that early stopping prevents both overtraining and overfitting [Livingstone et al. 1997]. Stopping training based on the performance of a test set is sometimes considered "cheating." Having a third dataset from the same sample (a validation set) is generally recommended. After deriving the best model based on the performance of the test set, the validation set can verify the model's ability to generalize.

2.1.7.3 Correct Classification Rate

The correct classification rate refers to the number of cases correctly classified according to their actual outcome with respect to the total number of cases in the database. This can be a good measure of performance when the outcome is relatively common (representative of approximately 15 percent of the database or greater). The classification rate is sensitive to the frequency distribution of the output. When a dichotomous output is highly skewed toward one outcome, the neural network typically classifies all cases as belonging to the outcome with the greatest *a priori* probability and results in a useless model [Ennett & Frize 1998].

2.1.7.4 Logarithmic-sensitivity Index

When working with data where the outcome is a rare occurrence (represented by only a few percent of the cases in the database), a new stopping criterion was required to help the automated networks achieve a better balance between the sensitivity and specificity of the classification results. The logarithmic-sensitivity index (or log-sensitivity index) in (2-11) attempts to achieve optimal sensitivity and specificity classification while slightly favouring higher sensitivity [Scales 2001].

$$\text{log-sensitivity index} = -\text{sensitivity}^n * \log_{10}(1 - \text{sensitivity} * \text{specificity}) \quad (2-11)$$

This function expands towards infinity as sensitivity and specificity approach one, but is slightly weighted in favour of sensitivity. The degree of weighting can be controlled by the exponent *n*. Studies have shown that an automated neural network which

optimizes the log-sensitivity index can classify outcomes better than manually-optimized networks, and can classify rare outcomes equally well or better than networks that use the maximum correct classification rate or the minimum mean squared error as the stopping criteria [Ennett et al. 2002, 2003].

2.1.8 Estimating Rare Outcomes

As mentioned above, often the outcome of interest is represented by only a small percentage of the cases, say, less than five percent of the cases. Under such circumstances it is difficult, if not impossible, to develop a model that can identify these rare outcomes. Typically, the resultant model classifies all cases according to the outcome with the greatest *a priori* probability, and none of the rare outcomes are identified, so the model is useless.

One approach that has been shown to improve the classification performance of a neural network is to selectively sample the cases with the rare outcome and create training sets with an artificially increased representative rate to compensate for the rarity of the outcome [Ennett 1999, Ennett et al. 1999, Ennett & Frize 2000, 2003]. When the neural network is trained with the higher percentage of rare cases, the test set is better classified.

2.2 Case-Based Reasoners

Case-based reasoners (CBR) are an extension of the expert systems called knowledge-based systems (KBS). CBRs attempt to overcome some of the limitations of KBSs. In contrast to KBSs that try to assess a new problem based on answers to predetermined questions and rules, CBRs draw on past experience from the case base and can adapt a previously successful solution of a similar problem to the new problem [Watson & Marir 1994]. The reasoning approach used by CBRs is quite intuitive because it is similar to the way people often solve problems, by experience rather than based on a set of rules. The underlying assumption is that similar problems have similar solutions [Schmidt et al. 2001]. CBRs can work with text-based (nominal) information in addition to numerical (ordinal) data. Four aspects of these two systems are compared in Table 2-1 [Watson & Marir 1994].

Despite the KBS-solutions that attempt to address these problems, CBRs offer a level of flexibility and adaptability that is difficult to achieve with KBSs. One of the strengths of CBRs is their ability to work in domains that are not well understood, as opposed to areas where there exists a concrete and well-known set of rules or generalizations [Kolodner 1993]. A CBR is suitable for applications where a set of past cases exists, previous experience is useful for solving problems, domain specialists have difficulty articulating their problem solving thought processes and experience is at least as valuable as textbook knowledge. In such situations, knowledge acquisition for a classical KBS is difficult and might produce inaccurate results [Harrison 1997].

Table 2-1. Comparison of KBS and CBR systems for problem solving

Domain issue	KBS approach	CBR approach
Knowledge elicitation	Gather information to be used for problem solving – difficult and time consuming, possibly require domain experts	Collect a variety of case histories and solutions to adequately cover the spectrum of the problem – need not be extensive
System implementation	Organize information in the form of rules or object models and develop an explicit model to represent this information	Identify significant features to describe a case – no need for explicit model
Case base management	Access to large data repositories is difficult and the procedure is slow	Use database search techniques to easily handle large volumes of data
Case base maintenance	Update underlying decision rules – time-consuming and often requires domain experts again	Add new cases to the case base to expand the breadth of problems that can be addressed – possibly need domain experts to adapt cases

2.2.1 The CBR Cycle

Aamodt and Plaza [1994] describe the CBR approach to problem solving as the cyclic process represented in Fig. 2-4. The procedure involves four basic steps:

1. RETRIEVE the most similar case(s);
2. REUSE the case(s) to attempt to solve the problem;
3. REVISE the proposed solution if necessary; and
4. RETAIN the new solution as a part of a new case.

Figure 2-4. The CBR cycle (adapted from Aamodt & Plaza [1994])

In other words, the new problem is matched to one or more similar cases in the case base. The solution from these matched cases can be used directly as a solution for the new problem, and is tested for success. If the proposed solution is not satisfactory then the solution can be modified to fit the new problem and the resulting new solution can be added to the case base to broaden its scope [Kolodner 1993, Watson & Marir 1994, Harrison 1997].

Unfortunately, this process cannot always be entirely automated, so some human intervention is necessary. Since the automation of the case retrieval and reuse aspects is relatively straightforward, CBRs are often used solely for those purposes. Case revision and adaptation are more complex and may require an expert or manager of the case base to intervene. This is not necessarily a drawback because it supports human-computer interaction in decision support [Watson & Marir 1994].

2.2.2 Case Representation

A case contains contextualized information about a particular event or individual related to the domain under investigation [Kolodner 1993]. The recorded data describes the problem's attributes using a set of input variables, and also generally has information about the derived solution to the problem and/or the actual outcome of the problem. Problem and solution sets can be used to solve new problems, whereas cases with problems and outcomes can evaluate new situations. Cases containing problems, solutions and outcomes can be used to test and validate proposed solutions [Kolodner 1993, Watson & Marir 1994].

2.2.3 Indexing

Manual or automated indexing is necessary so that the cases can be recognized and retrieved. Indices should follow guidelines to facilitate case base management and maintenance [Birnbaum & Collins 1989, Hammond 1989]. The indices should be intuitive, clearly define the purpose of the case, general enough to permit future

expansion of the case base, and specific enough to be recognized in the future. Kolodner [1993], an expert in CBR, feels that people are more effective at choosing indices than algorithms, so it is preferable to manually index a case base [Watson & Marir 1994].

2.2.4 Storage

Effective and efficient case storage is essential for successful case matching and retrieval. The mode of storage must consider the indices that describe the cases, and strike a balance between the semantic richness of the cases and indices and the ease of access and retrieval of cases. Two of these case memory models are the dynamic memory model and the category-exemplar model [Aamodt & Plaza 1994]. The basic objective of the dynamic memory model is to classify cases with similar properties under a more general heading and then subclassify the cases according to their differences. To prevent explosive growth in the number of indices as the number of cases increases, most CBR systems restrict the number of permissible indices to a limited vocabulary. In the category-exemplar approach, features are assigned different weights to reflect its importance with respect to the membership in different categories. This approach is sometimes useful for explaining the CBR's proposed solution to a user [Watson & Marir 1994].

2.2.5 Case Retrieval

The query describes the case in question for which a proposed solution is needed from the case base. The CBR uses a retrieval algorithm that relies on the information stored in the case-memory indices to identify the most similar cases to the query. This algorithm must be flexible to find partial matches, since an exact match in the case base is unlikely available. There are several methods for case retrieval that can be used alone or combined into hybrid retrieval techniques. Two examples are inductive clustering and nearest neighbours [Watson & Marir 1994].

2.2.5.1 Inductive Clustering

Induction algorithms are used to select the features that best discriminate between the cases by creating a decision tree-type structure to organize the case base. Clustering partitions the case base better than a list, so that retrieval algorithms are more efficient than a serial search [Kolodner 1993]. This technique works well when matching on a single case feature such as mortality that is the outcome and is also dependent on other features [Watson & Marir 1994].

2.2.5.2 Nearest Neighbours

Matches using the nearest neighbour algorithm are based on the weighted distance between stored cases and the query. Since it is difficult to choose the appropriate feature weights, this algorithm may not find the best match to the query. Common methods for choosing the feature weights include assigning uniform weights (all

weights are equal), assigning expert-chosen weights (i.e. weights assigned by a physician [Wang 1997, Frize & Walker 2000] or discovering them through trial and error. As well, retrieval time is an obstacle as it increases linearly with the number of cases, so the nearest neighbour approach is most effective when using a relatively small case base [Watson & Marir 1994, Harrison 1997].

2.2.6 Adaptation

This step in the CBR process involves adapting a previous solution to meet the needs of the current situation. At this stage, the significant differences between the matched case and the query are identified, and then specific rules are applied to properly adapt the recommended solution. This can be achieved using structural adaptation where adaptation rules are directly applied, or derivational adaptation that reuses the algorithms, methods or rules that generated the solution to the original case. A comprehensive CBR system uses structural adaptation rules for adapting poorly understood solutions and derivational techniques for adapting well-understood solutions [Riesbeck & Shank 1989, Kolodner 1993, Watson & Marir 1994].

2.2.7 The Easy Reasoner

The CBR used in this thesis work was a software tool from Haley Enterprises that integrates *The Easy Reasoner* to provide the CBR functionality with a combination of the Eclipse inference engine, the ClassIE (the Class Induction and Indexing Engine) and TIM (The Intelligent Memory) [The Haley Enterprise 1993, 1994]. This system used

optimized forward chaining using the Rete match algorithm and was easily embedded into a C++ application. *The Easy Reasoner* employs the nearest neighbour algorithm with inductive case retrieval. It can match on text strings by ignoring noise words (i.e. *and, or, there*, etc.), using *N-M*-grams to deal with spelling mistakes and it uses stems to identify root words that are the same [Watson & Marir 1994]. *The Easy Reasoner* uses binary decision trees that permit soft thresholds and pruning.

The case adaptation function of *The Easy Reasoner* was not employed in this thesis work. *The Easy Reasoner* was basically used for case retrieval, as is common among CBR applications. As described later in the methodology section (Chapter 6), the queries were adapted by filling in the missing values with the means of the closest-matching cases. The match set was updated manually and this was not an original function of the CBR from Haley Enterprises.

2.2.7.1 The Easy Reasoner's Retrieval Mechanism

The retrieval process for *The Easy Reasoner* consists of two phases: a pre-query processing phase and a query processing phase. The statistics of the database are used to develop an index for all records in the pre-query stage. The query processing phase uses the information in this index to identify the closest-matching cases using a nearest neighbours algorithm. The information in the fields is normalized, so that each field contributes equally to the distance between two cases. The user may assign weights to each term to represent its relative importance to the problem. The weight of the k -th term in the record i is

$$W_{ik} = \frac{F_{ik} \log(n_k / N)}{\sqrt{\sum_j (F_{ij} \log(n_j / N))^2}} \quad (2-12)$$

where N is the number of records, n_k is the number of different records in which term k occurs, and F_{ik} is the number of occurrences of term k in record i divided by the total number of terms in record i . Using this weight, a normalized distance between two fields is calculated according to the similarity measure S_i :

$$S_i = \frac{\sum_k W_{ik} W_k}{\sqrt{\sum_k (W_{ik})^2 \sum_k (W_k)^2}} \quad (2-13)$$

where W_k is the weight of the k -th term in the query. The normalized distance between a query and its matched case ranges between 0 and 1. The smaller the similarity measure, the closer the case is matched to the query. Fields with missing values are not taken into consideration, therefore the matches depend on the completeness of the cases in the database and the query [Haque et al. 2000].

2.3 Hybrid Systems

Hybrid systems offer an opportunity to benefit from the strengths of different artificial intelligence techniques, like expert systems and neural networks. Traditionally, the approach researchers have taken is to use the nonlinear modelling ability of ANNs to strengthen the reasoning ability of expert systems. This way we get the expert

knowledge and explainability of CBRs while using the unique-case problem solving techniques of neural networks. Medsker & Bailey [1992] discuss five different models for hybrid system integration with an increasingly closer relationship between the two systems: stand-alone, transformation, loose coupling, tight coupling and full integration [Medsker 1994].

2.3.1 Stand-alone Models

As the name suggests, a stand-alone hybrid model is simply an expert system and a neural network that are attempting to solve the same problem, but do not interact. This combined system is typically the preliminary step to system integration. Stand-alone models are useful to compare problem-solving approaches between the two systems and to validate the outputs of each system similar to a committee of classifiers. An advantage of this system approach is its simplicity. There is no attempt to interface the two systems so commercially available software can easily be incorporated. The main drawback is that the two components do not benefit from each other's strengths because there is neither interface nor sharing of information. Also, both systems must be simultaneously updated to accommodate new features added to one of the components, which effectively doubles the maintenance requirements of the model [Medsker 1994].

2.3.2 Transformational Models

Similar to stand-alone models in that the systems remain independent, transformational models are converted from one type of system to another. This means that the system can either transform from an expert system to a neural network or vice versa. The system features that are best suited to each particular stage establish which component is used for development and delivery. Neural networks transformed into expert systems are useful for data analysis and preliminary knowledge engineering to provide reasoning and explanations, whereas the reverse system can profit from the knowledge of an expert system to determine the initial conditions and training set for the neural network. Transformational systems are easily developed for specific problem domains. Although adequate transformations are possible, there is no fully automated approach for completely transforming the systems from one to the other. Again, significant modifications to one component can have a dramatic effect on the other and require additional maintenance [Medsker 1994].

2.3.3 Loosely-coupled Models

As the first real step toward integration, a loosely-coupled model communicates between the two systems using data files. Examples include preprocessors, post-processors, co-processors and user interfaces. Preprocessing systems can benefit from the neural network's pattern recognition ability and use this information in the expert system for problem solving. Post-processing involves an expert system that prepares the data and makes preliminary decisions and then passes that information via a data

file to the neural network for data analysis or forecasting. A co-processor transfers data back and forth between the systems and has the potential to solve difficult problems like iterative problem solving. The development of loosely-coupled systems is more straightforward than more fully integrated applications because commercially available software can be adapted to meet the problem's needs and reduces maintenance time. Despite this, there is often excessive redundancy in loosely-coupled systems. This translates into increased operating times because each component has to individually process its own problem. Limited file sharing also means that the components cannot profit from each other's strengths [Medsker 1994].

2.3.4 Tightly-coupled Models

Although loose and tight coupling overlap somewhat, performance is significantly enhanced with tightly-coupled models because they share information via memory resident data structures rather than simple external data files. Tightly-coupled pre-, post- and coprocessors are typically faster than loosely-coupled systems. Within the tight-coupling category, there also exist variations such as blackboards, cooperative systems and embedded systems. Blackboard architectures provide shared data structures to facilitate interactive problem solving using independent agents, which are typically knowledge-based systems. Integrated blackboard systems are useful for complex pattern recognition and advanced decision support. In cooperating systems, the expert and neural components work together like in the loosely-coupled coprocessors but are more interactive because of their more advanced data-sharing

abilities. These systems can be used for monitoring, control and decision making. When one component is embedded into the other system, better control of the main component's functions can result. Robotics and classification are examples where these embedded systems are implemented. The closer relationship between the neural network and expert system means that there is less communications overhead and this permits a faster runtime. Commercial software may still be used at this stage which still allows for flexibility and robustness, but the verification and validation process are much more complex, particularly for embedded systems. Redundancy due to the independence of the components is still a limitation [Medsker 1994].

2.3.5 Fully-integrated Models

Full integration means that the hybrid system shares data structures and knowledge representations. The components communicate through the symbolic and neural modes of knowledge representation. Reasoning is either cooperative or managed by a controller. Information is exchanged between the two systems through the input and output nodes in the neural network and the symbols used by the expert system. Examples of fully-integrated models are connectionist systems, the utilization of input-output nodes, subsymbolic to symbolic connectivity and integrated control mechanisms. Full integration improves robustness, performance and problem solving capabilities. As well, redundancy in the development process is significantly reduced or eliminated because the two systems are so closely integrated. Fully-integrated models are much more flexible than non-integrated models, so they can handle a wider range

of situations. The drawback is the increased complexity when developing the model due to a lack of commercially available tools for full-integration. Verifying, validating and maintaining these systems is challenging as well [Medsker 1994].

2.3.6 Examples of Other ANN-CBR Hybrid Systems

Chen and Burrell [2001] reviewed several approaches for the integration of CBRs and neural networks. Their focus was mainly on *ANN-based* CBR systems (basically, a CBR with a neural network structure), rather than the true integration of a neural network with a CBR [Becker & Jazayeri 1989]. Chen and Burrell [2001] also discussed a non-deterministic system based on a probabilistic neural network and a CBR [Myllymaki & Tirri 1993] where they suggested that missing values in the input vector could be replaced by the values in the output vector (i.e. the closest-matching case). In essence, they are suggesting replacing the missing values with the *a priori* probabilities, which is not a new approach. None of the systems discussed was attempting to solve the problem of missing values.

Reategui et al. [1997] created a hybrid system that uses the neural network to make hypotheses that guide the CBR system to find a similar case in the case base. These hypotheses are stored as symbolic diagnosis descriptors that help to develop explanations for the proposed solution. This system was not working with replacing missing values, nor was the technique similar to the approach proposed in this thesis.

Shin et al. [2000] developed an approach where they used the values of the connection weights from a three-layer neural network as part of four different techniques to calculate associated values for use as weights in a memory-based reasoning system. Their feature weighting methods were based on:

- *Sensitivity of an input node* (determining the change in prediction value when this node is removed from the network);
- *Activity of a node* (determining how much the activation level of a node changes based on its input value – no change, activity is zero; significant change, activity is high);
- *Saliency of a weight* (using the second derivative of the error with respect to the weight to determine the importance of an input node – a low saliency weight would be deleted); and
- *Relevance of a node* (measuring the variance of the weights into a node where an increase in error would be expected when the node's largest weight was deleted meaning that the node was highly relevant).

To contrast their prior assumption that feature relevance was invariant over the domain that may not always be true, they later revised this technique to create a local feature weighting method where the neural network provided case-specific weights during the learning process [Park et al. 2001]. Although they showed that their weighting technique classified cases better than uniformly weighting the inputs, none of their techniques significantly outperformed the others. Shin et al. [2000]'s system is

similar to the one proposed in this thesis, although it involves extracting weights from a three-layer neural network rather than a two-layer (linear) network, but, once again, this technique was not used to replace missing values.

2.3.7 Hybrid System Used in this Thesis

The hybrid system used in this thesis had a “give-and-take” relationship between the two techniques similar to the loosely-coupled model described in Section 2.3.3. The systems do not physically interact, but the output of one system is used as the input to the other, and vice versa. In this thesis, the modelling strengths of the ANN are used to select the matching weights of the CBR and the CBR’s query-matching ability permits the use of cases with missing values in the ANN to model the system. Naturally, the final goal for a decision system, such as the one proposed in this thesis, is to have a fully-integrated model. Other members of the MIRG research team are focusing on steps to achieve this goal by standardizing the data format to increase opportunities for file sharing between different software components of the decision-support system [Catley 2002, Catley & Frize 2002, 2003, Catley et al. 2003].

3 The Missing Value Problem

The problem of missing data in research has been addressed in several ways, ranging from simple statistical approaches to more unique ways based on knowledge of the reason for the missing data. After addressing the most common reasons that data are missing, this chapter outlines current approaches for replacing missing values, and weighs the pros and cons of each procedure.

3.1 The Reason for Missing Data

Before taking any steps to estimate the missing values, the reason that the data are missing in the first place must be clearly identified. This information will impact which approaches are most appropriate for the data set. Many approaches for replacing missing values assume that the data are missing at random. This is the easiest situation to deal with, although it is commonly misused. In this case, the data are assumed to be adequately representative of the sample population, so the missing values can easily be replaced using straightforward techniques like mean imputation [Little 1999].

When the data are not missing at random, the cause must be established. When the data are simply not collected, these values are called “system missing.” These values can often be replaced with “normal” values assuming they were not collected because it was unnecessary for the problem diagnosis or the parameters were presumed normal. This approach for replacing missing values can result in under-measurement [Knaus et al. 1981]. In an urgent situation, perhaps there was no time to run the tests unless absolutely necessary. In a non-urgent situation, the tests may not be required to make the diagnostic decision. When the data are missing because the tests were only performed when absolutely necessary, the values recorded for that variable may be highly skewed toward the more challenging cases, making several commonly used imputation approaches inappropriate. In some cases not all the data are recorded (i.e. repeated measurements of a particular test), or the test results are missing because they were lost or contaminated. Lost data may be coded as “-9” or “88,” for example, to indicate errors. Such values can be replaced with the mean if deemed appropriate or simply delete the cases.

No method of missing value imputation has been developed that works well for all situations. Several methods have been commonly used, but are not suitable for every situation and are sometimes inappropriately used. In the medical environment, the problem of missing values is complicated by the fact that often there are few common parameters when diagnosing different problems. This leads to inconsistency in the data collection over the patient population and makes imputation of data difficult [Pesonen

et al. 1998]. Unfortunately, ANNs cannot run experiments with incomplete cases, so we require a technique for dealing with missing data for this thesis work.

3.2 Outliers

Outliers can pose challenges in databases. The outlier may be a data-entry error like entering “0.21” rather than “21” for the fraction of inspired oxygen; impossible values like a value less than 21% or greater than 100% for fraction of inspired oxygen (since the atmospheric oxygen concentration is 21% and it is not possible to inspire more than 100% oxygen); or even cases of extreme but plausible values for special cases such as a baby with a respiratory rate of 300 breaths per minute (rare but possible). The data must be thoroughly checked to ensure data integrity. Outliers are often simply replaced with the appropriate minimum or maximum value to reduce any biasing effects. It is best to test the model with and without them to ensure that there is no impact on the results.

3.3 Complete-case Analysis

With the complete-case approach for dealing with missing data, all incomplete cases are discarded. This approach is simple and easy to execute, however, information is lost and an important assumption is made about the missing values. The assumption is that the complete cases are a random subsample that adequately represents the original sample population with respect to the outcomes. This hypothesis is violated when

there is a specific reason for the missing values; for example, a test was not performed, so no value was recorded. This leads to biased estimates, which are amplified as the percentage of incomplete cases increases [Little 1999]. A similar idea is to remove the parameters (rather than cases) with no or little data. Fewer input parameters can weaken the decision power of the decision support system. Before eliminating parameters, it is important to show they are insignificant.

3.4 Statistical Imputation of Missing Values

Although imputation of values can allow the use of a broader range of cases, its major drawback is that, essentially, it is “made-up” data. Of course, the objective is to make sure that this artificial data represents as closely as possible what the observed data may have been. The assumptions made about the predictive distribution of the missing values must be clearly stated. Note also that when an analysis treats imputed values like observed values, the uncertainty of the model is frequently underestimated [Little 1999]. Such approaches to “repair” the data set may result in data that does not adequately represent what the actual values may have been and the solution may be invalid [Gabrys 2000]. The statistical methods described below assume that the data are missing at random.

3.4.1 Mean Imputation

In this case, the mean value of the entire data set is imputed for the missing value [Gabrys 2000]. Mean imputation attenuates the distribution of the imputed variables,

and biases estimates of association and nonlinear quantities such as variances or percentiles [Little 1999]. Also, this approach does not adequately differentiate the nuances of each individual case [Pesonen et al. 1998].

3.4.2 Multiple Draw Imputation

For draw imputation, values are randomly picked from the variable set to replace the missing values. To reflect the uncertainty of the imputed values, multiple imputation should be used. Create M datasets (say $M = 5$), each having a different set of draws for the variables with missing data. Then each dataset is analyzed and the results are pooled. More efficient point estimates are achieved through multiple imputation compared to single imputation of draws due to the averaging over the datasets [Little 1999].

3.4.3 Random Number Generator

Replacing missing values using a random number generator can be achieved by imputing a random value based on the known mean and standard deviation of the variable in the database. The advantage of this approach is that the distribution of the data set can be taken into consideration, whether the distribution is normal, skewed, uniform, etc.

3.5 Committee of Classifiers

The objective here is to develop a set of classifiers trained using different subsets of the input parameters. Although it introduces an interesting approach to the problem of missing values, it becomes complex when dealing with a large number of input parameters [Gabrys 2000]. The number of classifiers required increases exponentially with the number of input variables.

3.6 Nearest Neighbours Algorithm

Typically the nearest neighbours technique replaces the missing value of a query with the same value as the case that is closest to the query in an n -dimensional field, where n is the number of input variables. This approach requires an additional tool to reduce the search space when working with large volumes of data or many input variables, otherwise comparing the query against each of the data points in the database can become computationally-intensive [Skiena 1998]. As mentioned briefly in Section 2.2.5.2, the nearest neighbours algorithm also allows the input variables to be weighted according to their perceived importance to find closer matching cases, but the choice of appropriate weights is difficult. The match weights can be assigned uniform weights (all weights are equal), expert-chosen weights (i.e. weights assigned by a physician [Wang 1997, Frize & Walker 2000]) or discovered through trial and error.

3.7 Reduction of Viable Output Alternatives

Even after replacing the missing values using a specific technique, there still exists an undefined uncertainty about the output of a model. This means that after the missing data has been replaced, at the output of a system there is no way to identify that this case had missing values. For example, a case with truly complete data (no missing values) can be classified into the same output as a case that contains imputed values for all of its parameters (all missing values) and it is impossible to distinguish between the two cases. In response to this fundamental problem, Gabrys [2000] presented an approach using a general fuzzy min-max (GFMM) neural network using hyperbox fuzzy sets as a representation of data cluster prototypes. Rather than identify the most appropriate classification of a particular case, the objective of the GFMM neural network was to reduce the number of viable classification alternatives for a case with missing values. This eliminates the need for a substitution scheme for the missing data. This approach is useful when there are multiple output classes and does not preclude the possibility of there only being one suitable output class for a particular case given there is evidence within the parameters present in the case profile [Gabrys 2000].

3.8 Packet Loss Recovery Techniques

The following examples apply specifically to packet loss recovery techniques for streaming audio [Perkins et al. 1998]. Audio differs from medical patient profile data in that some error can occur in audio with a barely discernable difference between the

system output and the true output. In some cases medical data may be more sensitive to imputation errors, for example, medical imaging. This discussion focuses on receiver-based error concealment techniques because they are comparable to what might be used with medical data.

Naturally, the objective of error concealment is to replace the missing data with something that is similar to the original. Audio signals, especially speech, have a tendency to be similar over the short-term. Therefore, these error concealment techniques work well for relatively small loss rates (≤ 15 percent) and for small packets (4-40 ms). Longer loss rates mean that entire phonemes may be lost, which can be noticeable to a listener. Ideally, error concealment techniques will work in conjunction with sender-based repair schemes, but we will only discuss the former. There are three categories of receiver-based recovery techniques: insertion-based, interpolation-based and regeneration-based schemes.

3.8.1 Insertion-based Repair

These techniques are simple and inexpensive to implement, but generally have poor performance. Table 3-1 describes the different insertion-based repair approaches: splicing, silence substitution, noise substitution and repetition. Noise substitution and repetition provide the best performance of the insertion-based techniques, and can be satisfactory compromises between the other poorly-performing insertion-based

Table 3-1. Types of insertion-based repair

Type	Features
Splicing	Completely eliminates the missing data and disrupts the timing of the stream
Silence substitution	Fills the gap with silence to maintain the timing of the data stream
Noise substitution	Fills in the missing data with white noise and maintains proper timing
Repetition	Replaces loss packets with copies of previous packets

schemes and the more complex interpolation-based and regeneration-based techniques [Perkins et al. 1998].

3.8.2 Interpolation-based Repair

These interpolation-based repair schemes attempt to use information surrounding the missing data through pattern matching as a basis for the replacement data. The expected result is more closely related to the missing data than what one would expect from insertion-based repair techniques. The techniques that are included in this category (waveform substitution, pitch waveform replication and time scale modification) are described in Table 3-2.

The one-sided waveform substitution scheme repeats the pattern across the gap, whereas the two-sided scheme interpolates between the two sides. In general, the two-sided approach works better, but both are better than the insertion-based schemes. Pitch waveform substitution works marginally better than waveform substitution.

Table 3-2. Types of interpolation-based repair

Type	Features
Waveform substitution	A one- or two-sided approach that uses information from before, and optionally after, the loss occurs
Pitch waveform replication	Uses a pitch detection algorithm on either side of the loss to interpolate the missing information depending on whether the signal was voiced or unvoiced
Time scale modification	Stretches audio from either side of the loss across the loss

Time scale modification is more computationally demanding, but it works better than the other two interpolation approaches. Overall, interpolation-based repair is more complex than insertion-based techniques, but its performance is typically better [Perkins et al. 1998].

3.8.3 Regeneration-based Repair

These approaches synthesize the lost packet using information about the audio compression algorithm. Table 3-3 describes the two main types of regeneration-based repair. Interpolation of the transmitted state and model-based recovery are computationally-intensive and hence expensive, but generally perform well. The computational demand increases greatly for the more advanced repair techniques, but the quality improvement is incremental at best. Generally, one-sided repair is sufficient, since the trade-off in complexity of the two-sided approaches is greater than the performance improvement achieved [Perkins et al. 1998].

Table 3-3. Types of regeneration-based repair

Type	Features
Interpolation of transmitted state	Interpolates between states for codecs based on transform coding or linear prediction; longer losses gradually fade the reproduced signal
Model-based recovery	Uses speech on one, or both, sides of the loss to fit a model that will generate speech over the loss

4 The Neonatal Medical Environment

To better understand the neonatal medical environment where the ANN-CBR hybrid approach was tested, a brief description of the medical environment and several neonatal mortality models are discussed, in particular, the Score for Neonatal Acute Physiology (SNAP) score. Finally, the Canadian Neonatal Network database is described in detail.

4.1 The Neonatal Intensive Care Unit

In the NICU, the key predictors of outcome are birth weight, gestational age, gender, race, Apgar score and mode of delivery [Gray et al. 1992]. The Apgar score quickly assesses the risk of complication or bad outcome [Apgar 1953, Apgar et al. 1958]. Although these characteristics are strong predictors, they do not account for all differences in severity of illness adequately on their own. They are so effective in estimating mortality risk, however, that including them as risk factors in a model may cloud the impact of other potential predictors. A useful model would offer insight into neonatal severity of illness *in addition* to these key factors.

Stevens et al. [1994] report that clinicians in the NICU tend to overestimate mortality risk for their patients, which impacts patient triage, transfer, initiation and termination of life support and allocation of medical resources. In their study, the physicians predicted mortality with a 90% sensitivity and a 68% specificity [Stevens et al. 1994]. In other words, the physicians accurately predicted death 9 out of 10 times, but over-predicted death in patients who survived about one-third of the time. This study demonstrates the clinicians' tendency to overestimate mortality (i.e. they predict death with a high sensitivity). A model that correctly classifies more patients who will survive (i.e. predicts survival with a high specificity) would be beneficial and complementary to the way that these experts think [Walker 2003]. Ideally, to ensure confidence in the model, a specificity of close to 100% and a sensitivity of at least 50% would be preferred. These overestimates of risk lead to more aggressive management and increase admission rates and ICU costs. More accurate mortality estimates could minimize unnecessary and costly transfers from primary and secondary-level centres to tertiary units.

4.2 Neonatal Mortality Models in the Literature

Several mortality risk models have been developed for the NICU. NTISS (Neonatal Therapeutic Intervention Scoring System) [Gray et al. 1992], based on the adult version of the model TISS (Therapeutic Intervention Scoring System) [Cullen et al. 1974, Keene & Cullen 1983], measures the severity of illness and risk of mortality

based on the points scored for the use of specific medical therapies. CRIB (Clinical Risk Index for Babies) uses six input variables including birth weight and gestational age to adjust the mortality risk score for the severity of illness (51% sensitivity, 95% specificity) [INN 1993]. Appendix A has details about the variables used and points scored for the Apgar, CRIB and NTISS scores.

4.2.1 Score for Neonatal Acute Physiology

SNAP (Score for Neonatal Acute Physiology) [Richardson et al. 1993a] was developed as a mortality risk model for the NICU using the same modelling technique as the adult ICU mortality risk model APACHE (Acute Physiology And Chronic Health Evaluation) [Knaus et al. 1981] and the PSI (Physiologic Stability Index) [Yeh et al. 1984]. Both of these scores are based on the “worst” status of various physiological measures during a 24-hour scoring period. The original SNAP also adopted the protocol of recording the worst status during the 24-hour scoring period after admission to the NICU. SNAP’s objective was to meet the following specifications (adapted from Richardson et al. [1993a]):

1. Base the score on physiology rather than diagnosis or therapy to reduce impact of case mix, diagnostic labelling, changes in therapy over time, or differences in styles of practice between NICUs;
2. Present useful information beyond the traditional risk factors such as birth weight, gestational age, sex, race and Apgar scores;
3. Establish validity and reliability in different practice settings (primary, secondary and tertiary units);
4. Provide information that is clinically relevant to facilitate its adaptation into clinical practice by physicians as well as health care researchers;
5. Use readily available clinical information to avoid unnecessary tests and to permit retrospective reviews;

6. Be relevant for a broad range of cases common in NICUs;
7. Incorporate admission data to identify the obstetric and demographic risks apart from the effectiveness of the NICU treatment.

Points were assigned to each patient case according to the variable's deviation from normal physiological values (either above or below normal values) to achieve an arithmetic sum that estimates the infant's mortality risk. The scale ranges from 0 to 5 according to the definitions in Table 4-1 [Richardson et al. 1993a]. Missing values were assigned zero (0) points under the assumption that the results would have been measured and recorded if they were important for clinical management.

SNAP used 37 input variables to assess mortality risk listed in Table 4-2 with the points scored for each deviation. The more severe the condition, the higher the points scored. SNAP was also shown to be highly correlated with nursing workload, length of hospital stay and therapeutic intensity [Richardson et al. 1993a]. The 37 SNAP variables plus birth weight, small for gestational age (SGA < 5th percentile) and 5 minute Apgar score < 7 composed SNAP-Perinatal Extension (SNAPPE) [Richardson et al. 1993b]. Appendix B defines each of the SNAP variables according to the SNAP Guidelines, as distributed to each hospital that collects SNAP data or uses the score.

Table 4-1. Definitions of SNAP point assignment [Richardson et al. 1993a]

Points	Definition
0	Normal physiologic range/Missing value
1	Physiologic deviation was sufficiently abnormal to merit careful monitoring
3	Deviation was such that most clinicians would alter therapy to correct it
5	Acutely life-threatening

Table 4-2. SNAP variable list and scoring system [Richardson et al. 1993a]

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
pO ₂	Low	>65	50-65	30-50	<30
pO ₂ /FiO ₂ ratio	Low	>3.5	2.5-3.5	0.3-2.49	<0.3
pCO ₂	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 (<5 th percentile)	5 points

Note that SNAP and SNAP-PE were designed to analyze groups of patients, not for predicting individual survival or for decisions regarding discontinuation of life support [Richardson et al. 1993a].

4.2.2 SNAP-II

Richardson et al. [2001] modified the original SNAP score to make it simpler and more reliable. This new score called SNAP-II selected six of the original 37 variables from SNAP using logistic regression to predict in-hospital mortality, as listed in Table 4-3. A different number of points were assigned to a patient based on the value of the variable recorded, making the number of points scored per deviation for SNAP-II different from SNAP. Birth weight, small for gestational age and Apgar score at 5

Table 4-3. SNAP-II variable list and scoring system [Richardson et al. 2001]

Variable	Range	Points
Lowest blood pressure	MBP ^a 20-29 mmHg	9
	MBP ^a <20 mmHg	19
Lowest temperature	95-96 °F	8
	<95 °F	15
Lowest PO ₂ /FiO ₂ ratio	1.0-2.49	5
	0.3-0.99	16
	<0.3	28
Lowest serum pH	7.10-7.19	7
	< 7.10	16
Multiple seizures	>1	19
Lowest urine output	0.1-0.9 mL/kg/h	5
	<0.1 mL/kg/h	18
Birth weight	750-999 g	10
	<750 g	17
Small for gestational age	< 3 rd percentile	12
Apgar score at 5 minutes	< 7	18

^a MBP= mean arterial blood pressure

minutes were also included to take these important factors into consideration. This was called SNAPPE-II – SNAP-II with the Perinatal Extension. Excluded patients were moribund babies, those transferred within 24 hours to the regular nursery, those admitted after having been discharged or after more than 48 hours since birth. The time frame for data collection was reduced to the first 12 hours after admission to the NICU. SNAP-II has also proven effective in predicting morbidity outcomes such as severe intraventricular hemorrhage and chronic lung disease in the NICU [Chien et al. 2002]. The area under the receiver operating characteristic curves (ROC curves) for the SNAP-II score was 0.91 ± 0.01 (range 0.84 to 0.92) with the best classification performance for the group of infants of all birth weights and the worst performance for infants of birth weight less than 1500g.

4.3 The Canadian Neonatal Network Database

The patient data collected by the Canadian Neonatal Network from January 8, 1996 to October 31, 1997 contains all admissions to seventeen NICUs from across Canada during that period (all but one were tertiary-level perinatal centres). This data accounts for 75% of all tertiary-level NICU beds in Canada [Lee et al. 2000, Richardson et al. 2001].

The original database contained 20488 admissions during the 22-month collection period (19507 infants when readmissions and transfers are excluded). Data was

collected on Day 1 (admission to the NICU), Day 3, Day 14 and Day 28 (or discharge). The experiments carried out for this thesis focused on the admission data (Day 1). Day 1 data was collected within the first 12 hours in the NICU. Patients who were in the NICU for less than 24 hours were excluded to avoid the impact of any differences in admission protocols at the different hospitals.

Neonatal mortality is defined by the World Health Organization as death occurring before 28 days of age [Health Canada 2000]. In the Canadian Neonatal Network database, only deaths to discharge are recorded because there was no follow up after discharge from the hospital. Most neonatal deaths occur in premature infants and they usually stay more than 28 days in hospital. This means that the true mortality rate in the database may actually be slightly higher than reported. The Canadian database would also miss any deaths in infants that occurred before 28 days of age for those who had already left the hospital, however, this occurrence would be rare [Walker 2000]. Moribund babies were excluded from this study since they only receive comfort measures rather than aggressive therapy [Gray et al. 1992]. Initial studies with the SNAP scoring system showed that moribund infants were assigned a zero SNAP score because no tests were performed nor monitoring values recorded, which inaccurately estimates their risk of death [Richardson et al. 1993a]. After excluding moribund infants and those missing data about mortality, birth weight, small for gestational age status and Apgar score at five minutes, the number of cases was 19427. Table 4-4

identifies some general characteristics about the 19507 infants in the Canadian Neonatal Network database [Lee et al. 2000].

4.3.1 Missing Values in Canadian Neonatal Network Database

Since the neonatal scoring systems listed above (CRIB, SNAP, SNAP-II, NTISS) use an arithmetic sum based on specific ranges of physiologic parameters to determine the severity of illness or risk of mortality, missing values for these scoring systems are simply scored as zero, which assumes that the actual values were within the normal range. This may not always be the case, because SNAP, in particular, requires the collection of numerous parameters. It is possible that certain abnormalities are not recorded because there was insufficient time and/or resources to collect all of the clinical information for all new admissions during the 12 hour collection period. This means that in some cases the missing values may not be normal, thus making the

Table 4-4. Perinatal risks and patient profile
(N= 19507) [Lee et al. 2000]

	Gestational Age (Weeks)				Total
	<28	28 to 32	33 to 37	>37	
Number of infants	1601	3817	7174	6786	19507
Outborn status (%)	25	17	21	37	26
<i>Perinatal risks</i>					
No prenatal care (%)	2	3	2	2	2
Maternal hypertension (%)	11	16	17	6	13
Cesarian section (%)	40	50	40	31	39
Multiple births (%)	24	27	19	2	15
Apgar <7 at 5 min (%)	39	14	8	16	15
<i>Infant characteristics</i>					
Male (%)	54	57	56	59	58
SGA (%)	4	5	4	4	4
Congenital anomalies (%)	9	8	12	20	14

assumption of the other scoring systems incorrect. Table 4-5 identifies the percentage of missing values for each parameter in the SNAPPE scoring system for cases where there were no missing values for mortality, birth weight, small for gestational age and Apgar score at 5 minutes ($N = 19427$). This database was preprocessed to ensure data integrity and treated outliers as appropriate. Appendix C contains a technical report that describes in detail how the outliers were treated. Calculations required to transform some of the SNAP variables for use in the SNAP score are outlined in Appendix D along with a cross-validation table to determine collinearities among the inputs and output.

From Table 4-5, notice that vital signs that are regularly monitored by medical support systems such as heart rate, breathing rate and temperature have very few missing values. Also parameters that simply required a yes or no response have no missing values. The greatest number of missing values are for parameters such as serum bicarbonate, ionized calcium, total calcium, blood urea nitrogen, creatinine and bilirubin. In a previous study using this database, Ennett et al. [2001] showed that patients who have fewer missing values have a higher mortality rate than patients with more missing values.

Table 4-5. Missing values in the Canadian Neonatal Network database ($N = 19427$)

Parameter	Variable name	Percent missing
Birth weight	bthwt	0.0
Small for gestational age	sga	0.0
Apgar score at 5 minutes	apgar5	0.0
Presence of seizures	seizure	0.0
Presence of apnea	apnea	0.0
Stool guaiac	guaiac	0.0
Highest heart rate	hheartr	0.2
Lowest heart rate	lheartr	0.2
Lowest temperature	ltempf	0.3
Highest respiratory rate	hrespr	0.3
Highest glucose	hgluc	8.4
Highest mean blood pressure	hbloodp	12.7
Lowest glucose	lgluc	13.9
Lowest mean blood pressure	lbloodp	15.9
Lowest white blood count	lwbc	17.0
Highest hematocrit	hhema	18.1
Highest immature/total neutrophil ratio	hitnrat	28.0
Lowest absolute neutrophil count	lanc	28.1
Lowest platelet count	lplt	28.2
Lowest hematocrit	lhema	33.0
Lowest serum pH	lserum	39.9
Highest pCO ₂	hpcO2	40.1
Lowest urine output	lurine	52.7
Highest sodium	hsodium	59.8
Highest potassium	hpotass	62.9
Lowest pO ₂	lpo2po2	63.0
Lowest pO₂/FiO₂ ratio	po2fio2r	63.9
Lowest sodium	lsodium	66.0
Lowest potassium	lpotass	68.1
Highest oxygenation index	oi	72.6
Highest total calcium	hcalctot	75.0
Highest indirect bilirubin	hidbili	77.4
Lowest total calcium	lcalctot	80.0
Highest direct bilirubin	hdbili	82.6
Highest creatinine	hcreat	85.2
Highest serum bicarbonate	hhco3	85.5
Lowest serum bicarbonate	lhco3	85.8
Highest blood urea nitrogen	hbun	87.9
Highest ionized calcium	hcalcion	89.4
Lowest ionized calcium	lcalcion	90.7

* Variables in bold belong to SNAPPE-II.

5 Problem Formulation

This section focuses on the details of the missing value problem. After restating the problem addressed in this thesis, evidence that the problem has not already been solved and validation that this is a worthwhile problem to address are presented.

5.1 Problem Statement

This thesis tested the approach of replacing missing values by integrating artificial neural networks (ANNs) and case-based reasoning (CBR). The database used for this investigation was from the Canadian Neonatal Network of seventeen Canadian neonatal intensive care units (NICUs). The objective was to use the network weights of a linear neural network as the CBR matching weights to identify the closest-matching cases. The next step was to replace the missing values in the query by the mean value of the matched cases.

After validating this approach, the technique was expanded to estimate missing values for cases with more of the 37 input variables for the Score for Neonatal Acute Physiology (SNAP). The weight-elimination cost function aided pruning in the neural

network to identify the most important input variables to predict neonatal mortality in the NICU, and a new neonatal mortality risk model was developed.

5.2 Hypotheses Restated

Part 1. The integration of the artificial neural network with the case-based reasoning system for imputing missing values will result in values that are as close as or closer to the actual values than those found by mean or random approaches.

Part 2. After confidence in the missing value imputation approach is established, the resultant new neonatal mortality model developed using an artificial neural network with weight-elimination will predict mortality in the neonatal intensive care unit equally well or better than other neonatal mortality prediction models in the literature.

5.3 Evidence that Problem Has Not Previously Been Solved

5.3.1 Limitations of Previous Imputation Approaches

The following section outlines the main drawbacks of the imputation techniques discussed in Chapter 3.

5.3.1.1 Complete-case Analysis

Although this is a simple approach for dealing with missing values, the missing data in the Canadian Neonatal Network database is not missing at random as is assumed when applying this technique. As noted by Ennett et al. [2001], the fewer missing values a patient has, the sicker the infant is. Support for this statement is shown by the fact that babies with complete cases for the SNAP-II score have a mortality rate of 9.7%, whereas the overall mortality rate for the entire database was 3.9%. The mortality rate of those missing at least one of the six SNAP-II variables was 1.8% [Ennett et al. 2001]. These percentages show that clearly there is a difference in mortality risk between the population with fewer missing values and those with many missing values. Therefore, it is inappropriate to assume that any models developed using complete-case analysis will be valid for the entire neonatal population.

Given the SNAP Guideline definitions in Appendix B, we can see that for many of the SNAP variables, data were only entered if the particular test was performed or measurement taken. This is also evident from the way that SNAP is scored, since the guidelines state that missing values can be assumed normal [Richardson et al. 1993a]. Knowing the reason for missing data in the Canadian Neonatal Network's database, it would be inappropriate to apply a model derived from complete-case analysis directly to the entire database.

Since no tests or measurements were taken unless absolutely necessary in order to make a diagnosis or to appropriately monitor the infant, there are in fact no cases in

the Canadian Neonatal Network database with complete data for all of the 37 SNAP variables. It would be extremely expensive and inappropriate to increase the laboratory workload for the NICU, so it is reasonable to accept that not all tests were performed or measurements taken for each infant. Such a model requiring all tests and measurements for all 37 SNAP variables would not be clinically feasible.

5.3.1.2 Statistical Imputation of Missing Values

Using the continuous data for the SNAP-II variables in the Canadian Neonatal Network database, Tong [2000] attempted to develop a neural network model where the missing values were replaced with the mean and median values without success [Tong et al. 1999, 2002]. Regardless of the attempts to optimize the network parameters, the neural network classified every case as a survivor. The database was biased towards the survivors and as such the neural network did not have sufficient information about the non-survivors to classify them correctly. Knowing that values were only recorded when necessary for diagnosis or treatment, replacing the missing values with the mean or median of these sicker babies biases the database.

For neonates, it is difficult to determine “normal” values because there are a number of factors that affect different parameters such as birth weight and gestational age. What may be normal for one infant may be dangerously abnormal for another. SNAP Guidelines avoid this issue by defining a range of values as normal and then coding them as zero. Although this can account for some of the variability in the “normal” range for neonates, issues may arise when the missing values are blindly coded the

same as normal values. Tong [2000] also tried replacing missing values with the low end of the normal values according to the SNAP Guidelines, but still could not create a neural network that could distinguish between the survivors and non-survivors [Tong et al. 1999, 2002].

Neither multiple draw imputation nor random number generator approaches have been used to replace missing values in this neonatal database. Although both of these techniques have several advantages over mean imputation and single draw imputation, the random number generator approach was chosen as a benchmark for comparison with the ANN-CBR hybrid imputation approach due to its ease of implementation and its ability to incorporate the frequency distribution of the database into its random value selection. It was anticipated that this might result in the imputation of data that were more closely related to what the actual values would have been.

5.3.1.3 Committee of Classifiers

This approach requires that a new model be derived for each combination of input parameters. Since the objective was to develop a neural network model with more of the 37 SNAP variables so that the weight-elimination network would prune the input variable list to identify the most influential variables, the committee of classifiers approach was too complex. The number of classifiers increases exponentially with the number of input variables; therefore, we would need to develop up to 37^2 individual models (1369 different models!). Even if we only considered the six SNAP-II variables, 6^2 (=36) models were too computationally complex for the problem at hand.

5.3.1.4 Nearest Neighbours Algorithm

This algorithm is closely related to the case-matching approach used in *The Easy Reasoner*, although the CBR software from Haley Enterprises also used induction techniques to narrow the search space and make the system more efficient. The nearest neighbours algorithm, particularly with the ability to weight the input variables, may be useful as a comparison with the CBR used in this thesis. This step was left for future work following the development of a flexible nearest neighbours software program with all of the added features of the research version of the CBR by Cotea & Jiwani [2003].

5.3.1.5 Reduction of Viable Output Alternatives

The approach introduced by Gabrys [2000] and Gabrys & Bargiela [2000] of using a general fuzzy min-max (GFMM) neural network with hyperbox fuzzy sets as a representation of data cluster prototypes does not adequately address the issue of having a single dichotomous output. The GFMM neural network technique would be interesting for distinguishing which of a number of diagnoses is relevant to each case, but cannot work when the output is two mutually exclusive outcomes.

5.3.1.6 Packet Loss Recovery Techniques

These approaches proposed different methods for replacing missing data, however, they were typically used in the telecommunications domain. According to the literature review, the packet loss recovery techniques have not yet been directly applied to the

medical domain for replacing missing data used for pattern recognition and outcome prediction. The insertion-based repair approaches have similar statistical counterparts. For example, splicing is similar to deleting missing data, silence substitution is comparable to replacing the missing data with zeros, noise substitution is like using random values, and repetition is related to mean imputation. Several of the interpolation-based repair and the regeneration-based repair approaches are similar to statistical techniques, although they may approach the problem from a different point of view. It would be interesting to apply the model-based recovery technique to replace missing values in a medical database as a comparison to neural networks, however, this was left for future work.

5.3.2 Limitations of SNAP and SNAP-II

Based on the information presented in Chapter 4, we know that many of the limitations of the SNAP score were corrected with the development of SNAP-II. An expert panel of five neonatologists, a pediatric intensivist and a neonatal nurse selected the original SNAP variables. The development of SNAP-II identified that some of the inputs for SNAP were not statistically associated with mortality and as such were eliminated when SNAP-II was created [Richardson et al. 2001]. SNAP was also developed based on data collected during the first 24 hours after admission that can confound the effectiveness of therapy with the initial severity of illness. This issue was also dealt with using the SNAP-II score in that the scoring period was reduced from the initial 24 to 12 hours in the NICU, similar to CRIB.

Some variables included in SNAP-II are peculiar, particularly the presence of seizures [Walker 2002a]. The prevalence of seizures is extremely low [only 0.7% of infants (140 cases) suffer from a single seizure and 1.5% of infants (300 cases) have more than one seizure], so it was surprising that it was found to be among the most influential parameters. As well, premature infants (who generally make up the majority of cases in the NICU) are less likely to have seizures, so inclusion of seizures as an important input variable would be expected among an infant population with higher birth weights. This means that although SNAP-II has proven clinically useful for a variety of outcomes, the final list of variables that predict important outcomes such as mortality is somewhat in question. This fact is important for a model's integration into clinical practice and acceptance by clinicians and other health care professionals.

Another limitation of SNAP that has not been addressed by SNAP-II is that premature infants generally stay in the NICU for an extended period of time. The result is that the admission-day information collected may have poor predictive value for late events and other complications. To address this issue, the time-varying data in the Canadian Neonatal Network database could be modeled to observe the impact of using Admission data/Day 1, Day 3, Day 14 and Day 28/Discharge information. Due to the large number of missing values in these databases, this is a challenging problem until an improved imputation technique can be developed. Previous studies by MIRC on mortality and duration of ventilation using the complete cases only for Day 1 and Day 3 data from the Canadian Neonatal Network database has shown no

improvement in predictive accuracy using the time-varying data [Walker et al. 2001a].

It is possible that with the inclusion of cases with missing values that the time-varying data may be more useful.

5.3.3 Prior Neural Network Approaches to Predict Neonatal Mortality

The MIRG research group has completed a variety of studies of the Canadian Neonatal Network's database where neural networks have been used to predict outcomes such as duration of mechanical ventilation and mortality. The results showed that neural networks may be able to satisfactorily estimate the probable duration of ventilation for individual patients in the NICU to assist appropriate allocation of unit resources and also ANNs may have utility in estimating outcomes important to parental counselling in the NICU such as survival [Walker et al. 2001a, 2001b, 2001c, 2001d]. These experiments were performed on cases with complete data only, therefore, the issue of missing values was not addressed.

Although neural networks have been used many times to predict medical outcomes, until now only two neural network models in the literature (other than those published by MIRG) have been developed to specifically predict neonatal mortality. In both cases, the models were based on a specific subset of patients: very low birth weight ($< 1500\text{g}$) [Zernikow et al. 1998] and extremely low birth weight ($< 1000\text{g}$) [Ambalavanan & Carlo 2001], whereas the model developed in this thesis work was based on patients of all birth weights.

Zernikow et al. [1998] looked at the mortality risk assessment of preterm neonates (<32 weeks gestational age and/or < 1500g birth weight). The training set of 435 patients had a mortality rate of 8.7%, and the test set of 455 cases had a mortality rate of 7.9%. They used a feedforward backpropagation ANN with three layers and three hidden nodes and a genetic algorithm to prune input variable set. Training stopped when the test set error stopped declining. The resultant ANN model used thirteen inputs to predict neonatal mortality: gestational age, birth weight, condition on admission stable, condition on admission fair/unstable, condition on admission vitally endangered, inborn, capillary pH < 7.10, temperature on admission < 35.5°C, emergency delivery, 10 minute Apgar score < 5, 5 minute Apgar score < 3, 1 minute Apgar score, and congenital malformation. The area under the ROC curve for this ANN model was 0.954.

Ambalavanan & Carlo [2001] developed an ANN model to predict mortality for 810 extremely low birth weight infants (training=502, validation=59, test set=249 cases). Their neural network was backpropagation-trained with four layers (nine hidden nodes, sigmoid transfer function, momentum 0.8, learn rate maximum 0.30, learn rate minimum 0.001). They also stopped training when the test set error stopped declining. They identified seven risk factors: birth weight, 5 minute Apgar score, antenatal steroids, race, respiratory distress syndrome, gestational age and multiple gestation. Mortality was 34% in the training set and 33% in the test set. The area under the ROC

curve was 0.87 ± 0.03 . At 80% sensitivity, the specificity was 85%, positive predictive value was 72% and negative predictive value was 90%.

5.4 Value of Developing a New Imputation Approach

There is a need for a new technique for replacing missing data that incorporates information about the relationships between the input parameters and about the closely related cases. The neural network identified the relationship between the inputs without any restrictions on the data type or any requirements for prior knowledge of the relationships between the input parameters. The ANN filtered through the data to recognize relevant patterns and associations. Then using the weights of the linear neural network as the weights of the CBR, the closest-matching cases were selected. The CBR identified cases similar to those with missing data using a matching algorithm. By taking the mean of the variable that was missing in the query from the matched cases, the missing value was replaced. This technique may have allowed valuable information to be extracted from the cases with missing data and increased the number of cases used to create the neural network mortality prediction model for the NICU. This new neonatal mortality model that was pruned by the neural network with weight-elimination may be a valuable tool for neonatologists. The model identified a new set of input variables that predict mortality in the NICU and was compared to the reduced SNAPPE-II model created using statistics.

5.4.1 Other Applications

The objective was to develop the ANN-CBR imputation technique such that it was generalizable to any database with missing data for predicting an outcome or diagnosis. Certainly, the missing data problem is widespread in the medical domain, but it exists in other applications as well. For example, consider the situation where there is a system error in a manufacturing system. Given monitoring information recorded on a semi-regular basis, the ANN-CBR hybrid would diagnose the problem. This new imputation approach could also be useful in the telecommunications industry with respect to information lost during the transmission of information (i.e. packet loss).

6 Methodology

This section outlines the approach for imputing missing values using the ANN and the CBR. The methodology can be broken down into three steps:

STEP 1: Validation of hybrid imputation approach for *known* missing values

STEP 2: Validation of hybrid imputation approach for *unknown* missing values

STEP 3: Development of a new neonatal mortality model

Each step is summarized and then described in more detail to justify the approach used. Testing methods and appropriate performance measures are defined to determine the degree of successful implementation for each of the above concepts.

6.1 Step 1: Validation of Hybrid Imputation Approach for *Known* Missing Values

This first step involved evaluating the ANN-CBR hybrid system's ability to impute missing values in an artificial database. This was necessary to validate the use of the linear ANN weights as the matching weights of the CBR to find the closest-matching cases. Removing known values to create an artificial database provided an opportunity

to see how closely the hybrid system was able to estimate the missing values compared to what the actual values were. The hybrid system outputs were compared to two benchmark approaches: mean and random imputation.

6.1.1 Step 1: Summary

Evaluate the hybrid system's ability to impute missing values in an artificial database.

- a. Use only complete cases from the SNAPPE-II variable set ($N = 5102$).
- b. Divide database into training/match set and test/query set and train ANN to find weights for use in CBR.
- c. Extract the ANN weights and recalibrate such that the largest weight=100.
- d. Create artificial missing value test set by deleting *known* values.
- e. Use different weight combinations in CBR to impute missing values.
- f. Compare error between true data and imputed values for the 3 hybrid approaches, and mean and random imputation.
- g. Use imputed test sets in ANN trained with complete set (same training set as before).
- h. Evaluate ANN classification performance on the test sets.

6.1.2 Step 1: Detailed Discussion

These preliminary experiments only used input variables from the SNAPPE-II scoring system. This data set of nine inputs including *bthwt*, *sga* and *apgar5* provided an adequate starting point to test the ANN-CBR hybrid system's ability to impute missing values. The SNAPPE-II score contains commonly collected data so a sufficient number of complete cases were available ($N = 5102$). As well, these variables were found through statistical analysis to be relevant for predicting neonatal mortality in the Canadian Neonatal Network's database [Richardson et al. 2001].

The database was divided such that the training set consisted of two-thirds of the database and the remaining one-third became the test set. All networks for this series of experiments were trained and tested with complete cases only (i.e. no missing data in the SNAPPE-II variables) since the neural network cannot deal with missing values. As well, it was necessary for the preliminary experiments to use only complete cases to test how close the imputed values were to the known values. Following these preliminary tests in Step 1, incomplete cases with imputed missing values were gradually included to broaden the scope of the model in Steps 2 and 3.

To determine the appropriate weights for the CBR to identify matched cases, ANN experiments using only complete cases and the log-sensitivity index to optimize the performance on the SNAPPE-II database were performed. From these preliminary experiments, the weights of the linear network were extracted and used as the weights for the CBR to determine the importance of each input variable for matching the cases

to the query. The absolute values of these nine ANN weights were recalibrated to a range of 1 to 100 for use in the CBR. One is the minimum weight permitted in the CBR system.

To evaluate how accurately the ANN-CBR hybrid technique imputed missing values compared to what the actual values were, it was necessary to create an artificial test set with missing data by deleting *known* values. To create cases with missing values, the percentage of missing values (and which variables would have missing data) was based on the statistics of the entire database, shown in Table 6-1, to determine a realistic percentage of missing data for each of the input variables. In addition, due to the importance of *bthwt*, *sga* and *apgar5* (and mortality as the outcome), cases missing these variables were excluded from consideration. This reduced the dataset from 20488 (entire set) to 20045 (no missing values for caselink) to 19427 cases (no missing values for mortality, *bthwt*, *sga* and *apgar5*). This means that each patient case could be missing values for up to four of the nine input variables (namely, *lbloodp*, *po2fio2r*, *lurine* and

Table 6-1. Missing value statistics for
SNAPPE-II variables ($N = 20045$)

Variable	Valid data	Cases with missing data	Missing (%)	Missing in artificial data set (%)
lbloodp	16813	3232	16.1	16
ltempf	19961	84	0.4	0
po2fio2r	7168	12877	64.2	64
lurine	9428	10617	53.0	53
lserum	12032	8013	40.0	40
seizure	20041	4	0.0	0
bthwt	19884	161	0.8	0
sga	19938	107	0.5	0
apgar5	19565	480	2.4	0
mort	20001	44	0.2	0

lserum), since there were no or few missing values for *seizure* and *ltempf*.

After extracting the weights from the linear ANN, the CBR was used to impute the missing values. For each case with missing data, the CBR identified the ten closest-matching cases from the remaining complete cases in its particular match set. Then the missing value in the query was replaced with the calculated mean from these closest-matching cases for the variable that was originally missing. To illustrate, imagine that a patient record is missing a value for *lurine*. The mean value of *lurine* for the ten closest-matching cases would then be substituted into the query to make it complete. Cases were matched using only input information to simulate a clinical setting where the patient's outcome would not be known.

It was important to attain a satisfactory compromise between finding enough cases for an appropriate range of values for the missing variable and ensuring that the matched cases were still sufficiently similar to the query. The number of closest-matching cases was chosen *ad hoc* as ten for these preliminary experiments. Matching on ten cases generally meets the criteria discussed above where the matched cases are relatively close to the query without drawing in unrelated cases. Also, with a matching database of more than 3000 cases, it is reasonable to accept that the ten matched cases were in some way related to the query.

The hybrid-imputed values were the mean of the ten closest-matching cases. Three test sets were used with the CBR weights set as follows to impute the missing values:

1. The nine SNAPPE-II variable weights set to the weights of the linear neural network;
2. The six SNAP-II variable weights set to the weights of the linear neural network and the perinatal extension variables (*bthwt*, *sga* and *apgar5*) set to 100; and
3. The nine SNAPPE-II variable weights set to 100.

To establish a benchmark for how accurately the ANN-CBR hybrid system imputed missing values, it was necessary to replace the missing values using mean and random statistics for comparison with the hybrid system. When the imputed values were compared to the known values in the complete data set, the percent error was calculated according to (6-1). Eq. (6-2) was used to calculate the average error. These errors were then compared between the true and imputed data sets in the original test set database for the hybrid, mean and random approaches. The objective was to minimize the difference between the imputed and actual values.

$$\text{percent error} = \frac{(\text{imputed} - \text{known})}{\text{known}} * 100 \quad (6-1)$$

$$\text{average error} = \frac{\sum \text{percent error}}{\# \text{ cases with missing values}} \quad (6-2)$$

After comparing the hybrid-imputed values in the artificial dataset with the known values to determine the percent error, the next step was to test their impact on the

neural network model's performance. Each "complete" set was divided into the same two-thirds training and one-third test sets, so their classification performance could be adequately observed. Each set had the same cases with missing values that were imputed according to the specific approach. The neural network classification performance of the initial experiments with the known/true values was then compared to the hybrid-imputed data sets, and the mean and randomly imputed data sets.

6.1.3 Neural Network Architecture

Based on past experience of MIRG, the neural network architecture chosen for this thesis work was the backpropagation feed-forward neural network with the weight-elimination cost function and the hyperbolic tangent transfer function [Trigg 1997, Ennett 1999]. The stopping criterion was the log-sensitivity index [Scales 2001, Ennett et al. 2002, 2003]. The ANN employed in this work has been successfully validated. Trigg [1997] compared the performance of this ANN using standard backpropagation and backpropagation with weight-elimination cost function developed by Weigend et al. [1990, 1991] using Tong's [1983] sunspot data. The results confirmed that both algorithms are working properly [Trigg 1997, Frize *et al.* 1997]. Therefore, it is with confidence that these experiments were carried out using the weight-elimination cost function. For the standard backpropagation-trained ANN, the cost function minimized was the sum of squared errors (SSE) between the target values and the network's outputs.

The backpropagation feed-forward ANN was originally programmed by Buskard [1994] and modified by Trigg [1997] for use in the matrix program, MATLAB. Charette automated the program [Frize et al. 2000, 2001] and added several useful functions that allow users to modify the experiments, with the following choices:

- Standard backpropagation algorithm or backpropagation with the weight-elimination cost function;
- Two- or three-layer architectures (i.e. no hidden layer versus one hidden layer);
- Variable number of nodes in the hidden layer of the three-layer neural network;
- Stopping criteria: highest test set correct classification rate, lowest test set average squared error, exact epoch stop, highest test set log-sensitivity index;
- Automated or manual program to operate the neural network;
- Selection of the initial values and ranges for learning rate (and its adaptive parameters), momentum, error ratio, weight-decay constant (and its adaptive parameters), weight-elimination scale factor.

The output modules provide training and test set information on the *a priori* statistics, ROC (receiver operating characteristic curve) values, correct classification rate, sensitivity, specificity, log-sensitivity index, average squared error and the epoch when these best results occurred. Graphs are produced that show the progress of the training and testing with respect to the correct classification rate, average squared error,

sensitivity, specificity and log-sensitivity index. Each network parameter's sensitivity to change in values for the automated network is also displayed graphically.

6.1.4 Case-based Reasoning System

The CBR system used in this thesis was *The Easy Reasoner* from Haley Corporation [The Haley Enterprise 1993]. A license for this proprietary software was purchased by MIRG in 1996 as part of their on-going research into the development of a clinical decision-support system. Originally it was used for an adult intensive care unit, but was adapted to work with the NICU and the Canadian Neonatal Network database. *The Easy Reasoner* calculates the similarity between the fields of two cases pairwise. Initially, the input fields are normalized so that each field contributes equally to the global distance between two cases. However, a weight is assigned to each input so that variables of greater importance for matching (as determined by the user) can have larger weights than the less important variables.

Rybachynski [2002] created a research-version of the CBR to specifically meet the needs of this thesis work. She made it flexible such that the user could change the number of matches, and the weights of each input can be user-specified as well. Refer to Fig. 6-1 for a screen capture of the CBR. To facilitate the imputation of the missing values, Rybachynski also added an automated matching feature ("Run Auto Matching" button on the user interface) that automatically replaces the missing values in the queries with the means of the ten closest-matching cases in the match set for the entire query data set. If the user wishes to impute missing values for specific cases only or would like to

6.1.5 Rationale for Using a Linear Neural Network

The use of a linear (two-layer) network to determine the weights of the CBR was necessary, despite the fact that a nonlinear (three-layer) network may have achieved better classification performance. A nonlinear model may result in overmodelling which would restrict the model's ability to generalize, in which case a linear network is better. As mentioned in Section 2.1.3, it is difficult to interpret the weights of a nonlinear neural network since there are many weights connecting each input to the output. Shin et al. [2000] developed four feature weighting functions that encompass information from the three-layer neural network weights for use in a memory-based reasoner using k -nearest neighbours. Although their results showed improvement over uniformly weighting the variables in the CBR, the four weighting functions appeared to perform equally well without clearly identifying which function was best. After establishing the ANN-CBR hybrid approach presented in this thesis, a comparison with Shin et al.'s work would be valuable for future work.

6.1.6 Rationale for Using a Case-based Reasoner to Match Cases

The CBR identifies similar cases using a matching engine. This system determines how closely related cases are by calculating the absolute "distance" between the cases in a multi-dimensional space (usually using a k -nearest neighbours algorithm); the smaller the distance is, the more similar the cases are. The CBR can match cases with missing data, however, it is only considering the variables that are present in each of the

particular cases [Haque et al. 2000]. Therefore, by exploiting the CBR's matching ability and weighting only the variables present for the query, it was possible to determine which cases in the database most closely resembled the query, despite missing values. The index that *The Easy Reasoner* creates during the pre-query processing phase facilitates faster retrieval of matched cases than is achievable by the brute force nearest neighbours algorithm [Kolodner 1993]. The CBR was modified to identify a specific number of complete cases from the database that were most similar to the case with missing values. From this set of matched cases, the statistical mean of the variable that was missing in the query was calculated and replaced the query's missing value. As well, a license for the CBR software was purchased by MIRG and the objective of this thesis provided an opportunity to assess the usefulness of the CBR as part of a hybrid system with the ANN. As mentioned earlier in Section 5.3.1.4, a comparison of the CBR's matching ability with the weighted-nearest neighbours algorithm would be interesting for future work.

6.1.7 Rationale for Equating Neural Network and CBR Weights

The weights in a neural network are presumed to reflect the importance that the network places on each input variable with respect to its impact on the outcome under investigation. This assumption can be drawn from the fact that the weight-elimination cost function causes small weights to shrink even closer to zero, eventually eliminating their effect on the network [Weigend et al. 1991]. These small weights have been

compared to “white noise” in a signal [Le Cun et al. 1990]. Given this reasoning, we would expect that the larger the network weight, the greater its influence on the decision boundary and the more important its associated input variable is [Masters 1995].

If we accept that the ANN weights of the inputs are related to their influence on the outcome, then we can reasonably expect that the matching weights of the CBR would be similar to (or the same as) the neural network weights. Naturally, the CBR matching weights are related to the significance of each input with respect to the output. If we consider that a certain input variable has a greater influence on the outcome than other inputs (in this case, which cases are matched to the query), then we would expect that this variable would have a larger weight than others when matching cases. If a variable impacts the process of matching to the query, it is likely closely related to the output under investigation as well. Given this information, we can accept that the neural network and CBR weights are potentially closely associated and would then have similar values.

6.1.8 Measures of Performance

To evaluate the performance of the ANN, the sensitivity, specificity, correct classification rate, constant predictor and the area under the ROC curves were used, as described in more detail in the following sections.

Table 6-2. Truth table

		Correct output		
		Not present	Present	
Model output	Not present	True negative (TN)	False negative (FN)	TN+FN
	Present	False positive (FP)	True positive (TP)	FP+TP
		TN+FP	FN+TP	Total cases

6.1.8.1 Contingency Table

A contingency or truth table, shown in Table 6-2, is helpful for understanding the concepts of the correct classification rate, sensitivity and specificity. The correct classification rate, or accuracy, identifies the percentage of correctly classified cases. It is the sum of the number of cases that were correctly classified into their respective classes out of all cases (i.e. true positives plus true negatives divided by the total number of cases in the dataset). The model's sensitivity identifies the rate at which the rare outcome cases are correctly classified (i.e. true positives divided by the total number of rare outcome cases in the dataset). Specificity is the rate of correct classification of the common outcome out of all common outcome cases (i.e. true negatives divided by the total number of common outcome cases in the dataset). These parameters are sensitive to the prevalence of the situation under investigation [Penny & Frost 1996].

It is important to consider the cost of misclassification, since there may be different costs associated with the misclassification of a patient. For example, predicting that a patient will die (but the patient actually lives) has a different associated cost than

foretelling survival, where in actual fact the patient dies. Although medical decision aids are not yet sufficiently accurate to be used as definitive decision-makers, misclassification of a patient could result in reduction of therapy or inappropriate treatment. In such a situation of different misclassification costs, the classification rate is not necessarily the best measure of performance [Kattan & Beck 1995]. Ideally, a satisfactory balance between the sensitivity and specificity can be achieved, but slightly favouring sensitivity (since sensitivity reflects the classification of the rare outcome).

6.1.8.2 Constant Predictor

This basic statistical benchmark classifies all cases in the output category with the highest *a priori* probability. The constant predictor is not always an ideal measure of performance when working with skewed databases because it falsely gives the impression that it is a better model based on the number of correctly classified cases. The constant predictor has 100% specificity and 0% sensitivity meaning that none of the rare outcome cases are correctly classified. It is usually only useful as a point of reference.

6.1.8.3 Receiver Operating Characteristic Curve

The area under the receiver operating characteristic curve (ROC curve) assesses the ability of the model to discriminate between two outcomes. The curve is a visual description of the operating points and potential trade-off between the true and false positive rates by varying the threshold value of the output node across its range of

values [Woods & Bowyer 1997]. It may be used to discover the optimal cutpoint for the test [Forsström & Dalton 1995]. A perfect model would have an ROC curve that is a step function that means that all cases were correctly classified. A useless classifier is a 45-degree positive diagonal line indicating that the sensitivity and specificity are equal. This model offers no improvement over random guessing. The closer the ROC curve resembles a step function, the better the classification ability of the model. Despite being a well-accepted measure of performance, the area under the ROC curve can still be influenced by the prevalence of the outcomes [Buchman et al. 1994].

6.2 Step 2: Validation of Hybrid Imputation Approach for *Unknown* Missing Values

After validating the ANN-CBR hybrid imputation approach for missing data with the data set of known values, the technique was expanded to fill in the missing values in cases where the true values were unknown. The objective was to expand the data set of SNAPPE-II variables with complete cases to include more cases and more of the SNAP variables, thus broadening the usefulness of the final neural network model that would be developed to predict neonatal mortality in the NICU. Consider Fig. 6-2 that illustrates the research using the SNAPPE-II database thus far and the two options for expanding this work. Each approach had its own advantages and limitations:

1. Expand the database vertically to include more cases with missing values for the SNAP-II variables (with no missing values for *bthwt*, *sga* and *apgar5*) or

Figure 6-2. Expansion of imputation approach for the NICU database

2. Expand horizontally to include more variables from the SNAP database that have missing values.

6.2.1 Vertical Expansion

The vertical expansion method increased the number of cases available for the model and broadened the spectrum of infants by including the babies in the database who were not as severely ill. The issue with this approach was that the imputed values were drawn from the sickest population and may not reflect what the actual value would have been for a baby that was less ill. To overcome this, the first cases to be imputed were missing only one of the six SNAP-II variables (any one of the six) and the mean

of the ten closest-matching cases allowed some of the less sick babies to be included among the matched cases. We showed in previous work that the fewer missing values that an infant has, the sicker the infant is [Ennett et al. 2001]. Knowing this, we may assume that an infant who is only missing one of the input variables may be similarly ill to a baby who is not missing any SNAP-II values. Then groups of these infants were added to the match database so that the mean of the matched cases would be gradually less influenced by the data from the sickest babies. Similarly, the next set of cases for imputation was the infants missing two of the six values and so on.

6.2.2 Horizontal Expansion

The second approach of expanding the database horizontally was useful because it worked with infants with similar severity of illness, so it was more likely that their actual values for the unrecorded data may have been similar to each other. The drawback was that there were fewer cases with which to work and results here may not be valid for the entire range of severity of illness, since this was the sickest population. The result of the horizontal expansion approach was a model for the babies with the greatest severity of illness. From this smaller database, the neural network with weight-elimination identified the most important variables for predicting mortality. This subset of input variables allowed more than just the SNAPPE-II variables to be taken into consideration, but also reduced the number of parameters from the full SNAPPE database (40 inputs) that needed to be imputed when the database was subsequently expanded vertically.

The vertical and horizontal expansion approaches were implemented in Steps 2 and 3. Vertical expansion was completed first because there were fewer variables to impute and it showed the impact of an increasing number of imputed values on the neural network's classification performance. Regardless of the initial approach taken to impute the missing data, the final goal was to have all of the missing values in the NICU database replaced with appropriate values from their closest-matching cases.

6.2.3 Step 2: Summary

VERTICAL EXPANSION: Expand hybrid imputation approach to *unknown* missing data.

- a. Using ANN weights from preliminary complete case experiments (Step 1, a-c), impute unknown missing values using the CBR and 5102 complete cases in match set. Start with cases missing only one SNAPPE-II value, then two, three and four (no missing values for mortality, *bthwt*, *sga*, *apgar5*, *seizure* or *ltempf*).
- b. Check statistics of imputed database to show that means are less influenced by the sick babies.
- c. Randomly separate complete and imputed databases ($N = 19403$) into training and test sets for use in ANN – use same random separation key for each approach (hybrid, mean, random) for better comparability.
- d. Using 2-layer ANNs, compare classification performance among the different approaches.

6.2.4 Step 2: Detailed Description

The objective of Step 2 was to expand the database of complete cases vertically to include more cases with the SNAPPE-II variables by imputing unknown missing values. Successful imputation of these missing values would validate the hybrid imputation approach for unknown values so that missing data could be imputed for more variables when the database was expanded horizontally.

The weights from the preliminary neural network experiments were used for case-matching with the CBR because they were developed from the same data set. With vertical expansion, it was possible to use the entire set of complete cases from the SNAPPE-II database ($N = 5102$) as the match set to provide more cases to match against for the patient profiles with unknown missing values. As in Step 1, cases with missing values were imputed consecutively and then added to the match database to enrich the match set. Figure 6-3 illustrates the distribution of missing values in the SNAPPE-II database. Note that cases missing five values were not included in this aspect of the experiments because there was too little data in addition to *bthwt*, *sga* and *apgar5* to adequately estimate the missing data (leaving $N = 19403$ cases for vertical expansion). Once again, three sets of CBR data sets were created: one with all CBR weights equal to the ANN-derived weights, one with all CBR weights equal to 100, and one set with the SNAP-II variables equal to the ANN-derived weights and the perinatal variable weights equal to 100.

Figure 6-3. Distribution of missing values in entire database for SNAPPE-II variables

To verify the assumption that adding cases with missing values would cause the mean of the database to gravitate toward the mean of the less sick babies, the statistics of the database were checked as cases with one missing value were added, then two missing values, etc. These statistics were also compared to the statistics of the mean and randomly imputed databases.

After imputing the missing values for the SNAPPE-II variables in the entire data set using the hybrid system, the imputed and complete cases were combined and then randomly separated into 2/3 training and 1/3 test sets for use in the neural network. The random separation key was saved so that it could be used to separate the mean and randomly imputed databases for better comparability. Finally, two-layer neural network experiments were performed to observe the classification ability of the differently imputed databases.

6.3 Step 3: Development of New Neonatal Mortality Model

Horizontal expansion of the database facilitated the development of an ANN model that included more than just the nine SNAPPE-II variables. When more variables were included in the database, their importance was evaluated using a linear neural network with the log-sensitivity stopping criterion. The variables with the smallest weights were removed from the model, and the new model was tested to see if the classification performance had improved. This process continued until the best parsimonious model was developed. This reduced set of input variables became a new neonatal mortality model. The missing values from the remainder of the database were imputed and the resulting neural network performance was compared with the SNAP scores.

6.3.1 Step 3: Summary

HORIZONTAL EXPANSION: Expand CBR imputation approach to include more variables with unknown missing values.

- a. Separate SNAPPE-II database into data sets with complete cases for an increasing number of variables.
- b. Divide database into training and test sets, and train ANN to find weights for CBR – extract the weights.
- c. Use all complete cases in CBR match set (query set is remaining cases from 5102 cases) and impute missing values starting with cases missing one value, then two, three, etc..
- d. Randomly separate complete and imputed databases into training and test sets for ANN – use same random separation key for each approach.
- e. Run 2-layer ANN to eliminate variables, then re-run 2-layer ANN to observe impact on classification performance.
- f. When classification performance begins to degrade, extract final set of ANN weights for use as CBR weights.
- g. Impute missing values in remainder of database using only the refined variable set starting with cases missing one value, then two, three, etc..
- h. Separate complete/imputed database into training and test sets using same random separation key as in Step 2 for the entire database ($N = 19427$).
- i. Run 3-layer ANNs and observe classification performance.
- j. Compare remaining variables and performance to the SNAP score models.

6.3.2 Step 3: Detailed Description

Since the ANN requires complete cases in order to develop a model, for the horizontal expansion it was necessary to separate the cases according to the number of missing values in each case. By choosing variables with the fewest missing values first, Table 6-3 shows the number of cases in each set as the number of variables in the data set increased. The horizontal expansion started with the complete set of SNAPPE-II variables as a basis and expanded from there. Note that no infant has data collected for all 37 SNAP variables (40 inputs including *bthwt*, *sga* and *apgar5*).

As the number of cases diminished and the number of input variables increased, we risked undertraining the neural network because there may not be enough information to sufficiently learn the trends in the database. This means that although there were 27 cases with 38 of the 40 SNAPPE variables, the ANN would be undertrained because there are more input variables than cases in the data set. To ensure that the neural network was adequately trained, three data sets from Table 6-3 were used:

1. Set A with 19 inputs, 4153 cases and a mortality rate of 9.1%;
2. Set B with 24 inputs, 2908 cases and a mortality rate of 9.3%; and
3. Set C with 30 inputs, 1237 cases and a mortality rate of 11.9%.

As in Steps 1 and 2, each of these data sets was separated into training and test sets for the preliminary neural network experiments to extract the weights of a linear network

Table 6-3. Division of database for horizontal expansion ($N = 19427$)

Parameter	Variable name	Number of inputs	Percent missing	Number of cases	Mortality rate
Lowest pO ₂ /FiO ₂ ratio	po2fio2r	1	63.9	5102	
Lowest urine output	lurine	2	52.7	5102	
Lowest serum pH	lserum	3	39.9	5102	
Lowest mean blood pressure	lbloodp	4	15.9	5102	
Lowest temperature	ltempf	5	0.3	5102	
Presence of seizures	seizure	6	0.0	5102	
Birth weight	bthwt	7	0.0	5102	
Small for gestational age	sga	8	0.0	5102	
Apgar score at 5 minutes	apgar5	9	0.0	5102	
Presence of apnea	apnea	10	0.0	5102	
Stool guaiac	guaiaac	11	0.0	5102	
Highest heart rate	hheartr	12	0.2	5102	
Lowest heart rate	lheartr	13	0.2	5102	
Highest respiratory rate	hrespr	14	0.3	5092	
Highest glucose	hgluc	15	8.4	4939	
Highest mean blood pressure	hbloodp	16	12.7	4934	
Lowest glucose	lgluc	17	13.9	4633	
Lowest white blood count	lwbc	18	17.0	4221	9.1%
Highest hematocrit	hhema	19	18.1	4153	N = 379
Highest immature/total neutrophil ratio	hitnrat	20	28.0	3771	
Lowest absolute neutrophil count	lanc	21	28.1	3769	
Lowest platelet count	lplt	22	28.2	3539	
Lowest hematocrit	lhema	23	33.0	2915	9.3%
Highest pCO ₂	hpcO2	24	40.1	2908	N = 271
Highest sodium	hsodium	25	59.8	1912	
Highest potassium	hpotass	26	62.9	1808	
Lowest pO ₂	lpo2po2	27	63.0	1807	
Lowest sodium	lsodium	28	66.0	1770	
Lowest potassium	lpotass	29	68.1	1766	11.9%
Highest oxygenation index	oi	30	72.6	1237	N = 147
Highest total calcium	hcalctot	31	75.0	553	
Highest indirect bilirubin	hidbili	32	77.4	253	13.3%
Lowest total calcium	lcalctot	33	80.0	248	N = 33
Highest direct bilirubin	hdbili	34	82.6	86	
Highest creatinine	hcreat	35	85.2	66	
Highest serum bicarbonate	hhco3	36	85.5	27	
Lowest serum bicarbonate	lhco3	37	85.8	27	18.5%
Highest blood urea nitrogen	hbun	38	87.9	27	N = 5
Highest ionized calcium	hcalcion	39	89.4	0	
Lowest ionized calcium	lcalcion	40	90.7	0	

for use in the CBR for case-matching. These ANN weights were recalibrated to a maximum of 100 for the CBR weights.

The order of imputing the missing values in the horizontal database required some consideration. There were two possible approaches for expanding the database: variable-wise or step-wise.

6.3.2.1 Variable-wise Horizontal Expansion

As illustrated in Fig. 6-4, variable-wise horizontal expansion involved first developing the model for the data set with the largest number of variables (Set C), and then imputing the missing values for the 6 inputs that were missing in Set B. This would result in the cases in Set B having complete data for all 30 variables in Set C, thus “increasing” the size of the Set C database from 1237 to 2908 cases. The advantage of this approach was that the cases in Sets B and C would be similar because they were missing data for very few variables, so they had the most in common being among the

Figure 6-4. Variable-wise division of database for horizontal expansion

very sickest babies in the database. A drawback was that the ANN model would be built on fewer cases so the model may be less accurate. As well, after Set B was expanded to 30 variables, when working with Set A, which model would be used for imputing the 5 variables that differ between Sets A and B? The question arose of whether a new model would be developed for each set, or whether the Set C model would be used to impute the missing values in the entire database. These sickest babies were a different population, so it would not likely be appropriate to use the model developed solely on their circumstances and then apply it to the entire database. It may also be questionable to replace some values with one set of CBR weights and the other values with another set of weights.

6.3.2.2 Step-wise Horizontal Expansion

The step-wise horizontal expansion approach is illustrated in Fig. 6-5 and involved starting with Set A and imputing the missing values in the remaining 949 of the 5102 cases. Then the five new variables in Set B were imputed for the remaining 2194 cases that were missing these values. And finally, the 6 new variables in Set C were imputed for the remaining 3865 cases, so that the data set of 5102 cases had a complete set of values for the 30 input variables in Set C. In addition, each time the cases with missing values were consecutively replaced according to the number of missing values and then added to the match set to enrich it and gradually change the mean values. The advantage of this approach was that the means for the CBR-imputation were chosen from a larger data set providing a better opportunity to find the closest-matching cases.

Figure 6-5. Step-wise division of database for horizontal expansion

The main drawback was that as more variables were added to the model, the order of importance of each variable could change because there were more variables to take into consideration. Therefore, the step-wise horizontal expansion allowed a gradual change in the patient-base so that it was easier to find closer-matching cases. Given these reasons, the step-wise horizontal expansion approach was used.

The imputed databases were randomly separated into training and test sets for use in the neural network. At this point, linear ANN experiments were performed to attempt to eliminate variables using the weight-elimination cost function, and these 2-layer neural networks were re-run to see if there was any change in the classification performance. When the classification performance began to degrade after the removal of variables with the smallest weights, the weights of the remaining variables were

extracted for use in the CBR. The missing values in the remainder of cases in the entire database were imputed using only the ANN weight set as the CBR match weights, such that a complete/imputed database of 19427 cases with the refined number of input variables was created. Using the same random separation key as in Step 2 (and adding the remaining 24 cases to the training set), a final set of experiments using three-layer networks was performed. The classification performance of this new neonatal mortality model was compared with the SNAP score models. The refined list of variables in the ANN mortality model was compared with the variables in the SNAP scores to see if there was any agreement with respect to the most influential variables for predicting neonatal mortality.

7 Simulation Results and Evaluation

This chapter outlines the experimental results of the ANN-CBR hybrid system's attempt to replace missing values in the NICU database.

7.1 Step 1 Results: Validation of the Hybrid Imputation Approach

7.1.1 Weights with the Complete-case Database

Initially, the neural network was trained and tested with only complete cases (i.e. no missing values) to identify trends in the database. The importance of the input variables was reflected in the size of the neural network weights. The 5102 cases from the SNAPPE-II database were separated with 3358 cases (66%) in the training set and 1744 cases (34%) in the test set. A two-layer neural network was trained and tested on these data sets so the weights could be extracted for use in the CBR. In an attempt to see if the order that the variables are introduced to the neural network affected their final weights, the order of the inputs was changed (hence, “9 inputs (modified order)”). The recalibrated weights (adjusted such that the maximum weight is 100 for the CBR)

Table 7-1. Recalibrated mean connection weights from Step 1 ANN experiments

Input variable	9 inputs			9 inputs (modified order)		
	Average ANN weight	Recalibrated CBR weight	Variable order ^a	Average ANN weight	Recalibrated CBR weight	Variable order ^a
po2fio2r	-1.6081	100	3	-1.5964	100	1
lurine	-1.0508	65	4	-1.0449	65	2
apgar5	-0.6959	43	9	-0.6828	43	3
ltempf	-0.6793	42	2	-0.6682	42	4
sga	0.5176	32	8	0.5136	32	5
lserum	-0.4459	28	5	-0.4363	28	6
bthwt	-0.3126	19	7	-0.3062	19	7
lbloodp	-0.3016	19	1	-0.2935	18	8
seizure	0.3092	19	6	0.3041	19	9

^aOrder the variables were presented to the neural network.

for each input variable is shown in Table 7-1 (default neural network parameter settings are recorded in Table E-1 of Appendix E). As shown in Table 7-1, there appears to be virtually no effect on the final weights found by the neural network for each input variable. The magnitude of the weights for both sets of experiments is similar for each variable.

The perinatal extension variables *bthwt*, *sga* and *apgar5* did not have the largest weights compared to the other six input variables. These three variables were not among the original SNAP variables that were used to develop the SNAP-II model. Neonatologists use birth weight, small for gestational age status and Apgar score at 5 minutes like demographic variables, analogous to one's address and phone number. Using this information, neonatologists can quickly assess to which risk category the infant may belong. According to Dr. C.R. Walker, it may be difficult for a model to be accepted

by neonatologists in a clinical setting unless these three variables had the largest weightings in the CBR model for matching cases because of the importance clinicians place on these variables [Walker 2002b]. To accommodate the clinical expertise of neonatologists and to compare with uniformly weighting the input variables, three sets of CBR weight combinations were tested:

1. ANN-extracted weights for all nine inputs (hereafter referred to as “ANN weights”);
2. ANN-extracted weights for the six SNAP-II variables and weights = 100 for birth weight, small for gestational age status and Apgar score at 5 minutes (“clinician weights”); and
3. Uniformly weighting all nine input variables: all weights = 100 (“uniform weights”).

7.1.2 Development of the Artificial Database with Missing Values

An artificial database with *known* missing values for the test/query set was required to validate this hybrid imputation approach. Using the frequency statistics in Table 6-1, the artificial data set was created by randomly selecting 16% of the database and deleting their values for *lbloodp*. Then the database was reshuffled, and 64% of the database was selected so that the *po2fio2r* values for those cases could be deleted, and so on. This approach lead to the creation of an artificial missing values database that approximated the true distribution of missing values in the entire NICU database.

Eventually, these known values were compared with the values imputed by the three hybrid methods, mean and random approaches. Table 7-2 compares the resultant distribution of the number of missing values in the artificial query set with the entire database.

7.1.3 Calculation of Mean and Random Values

As benchmarks to evaluate the ANN-CBR hybrid system's imputation approach, a set with mean values and a set using random values for replacing missing data were created. The mean imputation was performed using the "series mean" function (SMEAN) in SPSS, which replaces missing values with the mean of the entire series.

Random imputation was achieved using the SPSS functions for randomly selecting a value based on a normal distribution with a specified mean and standard deviation ('RV.NORMAL(mean,stddev)') and for returning a random value from a uniform distribution with a specified minimum and maximum ('RV.UNIFORM(min,max)'). The normal random function was used for inputs whose distribution was approximately normal and the mean and standard deviation were specified. The

Table 7-2. Distribution of missing values in artificial Step 1 query data set, SNAPPE-II variables

Number of missing values	Query set, $N = 1744$		Entire database, $N = 19403$	
	Frequency	Percent	Frequency	Percent
No missing values	156	8.9%	5102	26.3%
One missing value	604	34.6%	3850	19.8%
Two missing values	632	36.2%	4279	22.1%
Three missing values	316	18.1%	3703	19.1%
Four missing values	36	2.1%	2469	12.7%

uniform random function was used for variables that had an approximately uniform distribution or for variables with a highly skewed distribution, but where the majority of values fell within a specified region. The minimum and maximum values chosen for *lurine* and *po2fio2r* specified the range within which 95% of the data fell. Fig. 7-1 shows the syntax for the random imputation of missing values, including whether normal or uniform random values were imputed for the specific variables with missing data.

There were also two possibilities for imputing the mean and random values: using the distribution of only the complete cases (the subset of the database, $N = 5102$) or the distribution of the entire database ($N = 19427$). Both were calculated to show the influence of using the entire distribution versus the subset. The entire distribution is a more likely scenario in practice because typically the frequency analysis would be of the entire database rather than just the complete cases. The average values and standard

Random imputation, $N = 5102$, complete cases only

```
IF sysmis(lbloodp) lbloodp=rv.normal(33.11,8.303).
IF sysmis(lserum) lserum=rv.normal(7.3025,0.11430).
IF sysmis(lurine) lurine=rv.uniform(0,2.30).
IF sysmis(po2fio2r) po2fio2r=rv.uniform(0,4.20).
```

Random imputation, $N = 19427$, entire database

```
IF sysmis(lbloodp) lbloodp=rv.normal(37.88,10.003).
IF sysmis(lserum) lserum=rv.normal(7.3162,.10967).
IF sysmis(lurine) lurine=rv.uniform(0,2.45).
IF sysmis(po2fio2r) po2fio2r=rv.uniform(0,4.30).
```

Figure 7-1. Syntax for random imputation of missing data

deviations for the variables with imputed data are shown in Table 7-3. Table 7-3 shows how the frequency distribution of the entire database skews the mean and random values away from the true values of the artificial database, because the database is a distinct population of the sickest infants in the NICU. However, to provide a better comparison for the hybrid-imputed values, the mean and random imputed values from only the complete cases were used for the subsequent re-training of the neural network to observe differences in the classification performance caused by the imputed values.

7.1.4 Imputation Error for Known Values

Pairwise comparisons between the true values and the imputed values using the hybrid, mean and random approaches using a *t*-test were performed (tables shown in Appendix F). The *t*-test showed no significant difference between the true and

Table 7-3. Average values and standard deviations for Step 1 test sets with imputed values in artificial missing value database

Imputation approach	<i>lbloodp</i>	<i>lserum</i>	<i>lurine</i>	<i>po2fio2r</i>
True values	32.41±8.08	7.30±0.12	0.85±0.75	1.83±1.30
Hybrid: ANN weights	32.55±5.40	7.30±0.05	0.85±0.29	1.88±0.53
Hybrid: Clinician weights	32.71±5.57	7.30±0.05	0.84±0.29	1.88±0.50
Hybrid: Uniform weights	32.69±5.50	7.30±0.05	0.84±0.30	1.89±0.55
Mean (<i>N</i> = 5102) ^a	33.11±0.00	7.30±0.00	0.84±0.00	1.85±0.00
Mean (<i>N</i> = 19427) ^b	37.88±0.00*	7.32±0.00*	0.94±0.00*	1.90±0.00
Random (<i>N</i> = 5102) ^a	33.77±7.97*	7.30±0.11	1.14±0.67*	2.11±1.22*
Random (<i>N</i> = 19427) ^b	38.68±9.61*	7.32±0.11*	1.22±0.71*	2.16±1.25*
No. of cases	286	689	919	1066

^aComplete cases only

^bEntire database

*Significantly different from true values

imputed values for the three hybrid approaches and mean ($N = 5102$) imputation for all four variables with missing values in the artificial database ($p > 0.05$). These results suggest that the imputed values of the hybrid methods and mean imputation based on only complete cases are similar to the true values in the original database. There was a significant difference between the true and imputed values for random ($N = 19427$) imputation for all four variables with missing values ($p < 0.05$). The results also showed that the imputed values for mean ($N = 19427$) imputation were significantly different than the true values for *lbloodp*, *lserum* and *lurine* ($p < 0.05$), but not for *po2fio2r* ($p > 0.05$). The imputed values for random ($N = 5102$) imputation were significantly different than the true values for *lbloodp*, *lurine* and *po2fio2r* ($p < 0.05$), but not for *lserum* ($p > 0.05$).

Confirmation of Hypothesis #1

These results confirm Hypothesis #1 because the ANN-CBR hybrid approaches for imputing missing values resulted in values that were not significantly different from the true values. The hybrid-imputed values were as close to the actual values as those found by complete case mean imputation and closer to the actual values than those found by random imputations.

Table 7-4. Percent error of Step 1 imputed values in artificial missing values database (SD= standard deviation)

Imputation approach		<i>lbloodp</i>	<i>lserum</i>	<i>lurine</i>	<i>po2fio2r</i>
Hybrid: ANN weights	Mean (%)	15.73	1.14	174.23	94.62
	SD (%)	16.13	1.18	488.09	146.80
	Max (%)	113.33	13.26	10300.00	1683.33
Hybrid: Clinician weights	Mean (%)	15.97	1.16	172.21	97.66
	SD (%)	15.09	1.17	449.45	155.11
	Max (%)	106.67	13.10	7600.00	1762.50
Hybrid: Uniform weights	Mean (%)	15.78	1.12	176.62	93.40
	SD (%)	15.21	1.17	485.67	142.98
	Max (%)	100.00	13.57	9600.00	1690.00
Mean (<i>N</i> = 5102) ^a	Mean (%)	22.60	1.16	191.86	107.07
	SD (%)	21.97	1.22	533.35	181.61
	Max (%)	154.71	13.92	8319.56	2217.75
Mean (<i>N</i> = 19427) ^b	Mean (%)	30.95	1.15	216.32	110.24
	SD (%)	29.30	1.28	598.24	186.90
	Max (%)	191.62	14.20	9300.00	2275.00
Random (<i>N</i> = 5102) ^a	Mean (%)	30.45	1.72	298.56	146.02
	SD (%)	28.06	1.49	948.56	267.40
	Max (%)	215.61	13.09	18300.00	4487.50
Random (<i>N</i> = 19427) ^b	Mean (%)	39.40	1.68	319.67	149.94
	SD (%)	37.04	1.50	1012.07	274.55
	Max (%)	266.84	13.34	19500.00	4600.00

^aComplete cases only

^bEntire database

Table 7-4 shows the percent error associated with the imputed values for each variable with missing data. The percent error and average error were calculated according to (6-1) and (6-2) as defined in Section 6.1.2, respectively. The range of possible values for each variable may have played a role in the percent error for the imputed values. The maximum error was presented as a point of reference. Table 7-5 shows the range that contains approximately 95% of the data for each of these variables.

Table 7-5. Range containing 95% of data
for variables with missing values

Variable	Percent missing	Range
lbloodp	16	20-56
lserum	40	7.09-7.52
lurine	53	0-2.24
po2fio2r	64	0-4.20

The mean errors for the hybrid-imputed values were similar, whereas there was often greater error with the mean and randomly imputed values. The maximum error showed that if the range is broad, there was a greater possibility that the imputed value could deviate from the true value. Still, most often the hybrid-imputed values had a smaller maximum error than the mean and random approaches. Note also that the lowest error occurred for imputed values of *lbloodp* and *lserum*. These two variables also had the smallest number of missing values, although *lserum* was missing 40% of its data. Conversely, a narrower range meant that there was less variance and therefore less possibility for error in the imputed value. Note the restricted range of *lserum* that may explain its low average percent error.

Both *po2fio2r* and *lurine* are derived variables rather than raw values and this may explain the larger errors found for these imputed values. *Po2fio2r* is actually the lowest value of three ratios: $lp02po2/lp02fio2$, $fio2po2/fio2fio2$ and $mauppo2/maupfio2$, and *lurine* is calculated by dividing the lowest urine output by the infant's birth weight. Initially, it was anticipated that using the raw values for the *po2* and *fio2* variables would reduce the error for the imputed values. However, upon closer examination of the NICU

database, it was discovered that it was rare for one of the values to be recorded and not the other. Therefore, it was not reasonable to attempt to separate these derived values since in practice, when both values were missing, they would result in the same CBR matches as if the query case was searching for the derived value. Also, the *po2/fio2* ratio has more clinical value than the individual raw values that are meaningless by themselves, because it gives a good snapshot of the patient's overall lung health [Walker 2002]. Similarly, *lurine* could not be broken down into its component values.

Another factor affecting the model's ability to accurately impute a value may be the number of missing values the case had. Since the CBR was only matching on complete data, fewer values offer less information for the CBR to assess how closely matched a case is to the query. To determine if the number of missing values in each case affected the model's accuracy, Table 7-6 separates the patient cases according to the number of missing values each case had. The specific combination of variables missing is not noted here, only the number of missing values in a particular case.

There are two factors that influence the percent error: (1) the number of missing values in the case; and (2) the number of cases missing values for that particular variable. In general, the more missing values a case has, the less information the CBR has to find the close-matching cases, so there is a greater possibility for imputation error. As well, the more cases missing values for a particular variable, the greater the room for error.

Table 7-6. Percent error for Step 1 hybrid-imputed values separated according to the number of missing values in each case

Variable	No. cases	ANN weights		Clinician weights		Uniform weights	
		Avg error	Max error	Avg error	Max error	Avg error	Max error
		(%)	(%)	(%)	(%)	(%)	(%)
<i>One missing value</i>							
lbloodp	48	14.87	61.11	13.44	50.00	13.44	50.00
lserum	109	1.06	5.61	1.11	5.61	1.02	5.04
lurine	184	160.88	2625.00	154.32	2375.00	165.21	2350.00
po2fio2r	263	99.98	778.57	102.29	814.29	97.90	792.86
<i>Two missing values</i>							
lbloodp	97	16.33	71.43	16.39	69.23	15.99	71.43
lserum	280	1.11	6.45	1.08	6.08	1.06	6.37
lurine	413	207.08	10300.00	202.73	7600.00	208.39	9600.00
po2fio2r	474	88.28	1683.33	90.82	1750.00	86.65	1690.00
<i>Three missing values</i>							
lbloodp	105	14.31	68.75	15.32	68.75	15.33	75.00
lserum	264	1.19	13.26	1.23	13.10	1.21	13.57
lurine	286	142.11	2700.00	142.82	2475.00	143.47	2450.00
po2fio2r	293	98.89	1462.50	103.30	1762.50	98.76	1440.00
<i>Four missing values</i>							
lbloodp	36	19.36	113.33	20.07	106.67	19.66	100.00
lserum	36	1.35	9.92	1.38	10.38	1.32	10.23
lurine	36	120.65	1700.00	147.14	2400.00	133.76	1866.67
po2fio2r	36	104.09	446.67	107.82	446.67	105.72	484.44

From Table 7-6, we see that in general as the number of missing values in a case increased, the average error also increased. The exception occurred amongst individual variables where there were a similar number of cases missing, particularly between two and three missing values. In these situations, the error either remained the same or was slightly lower. For example, the average error for *lbloodp* when all weights were set to the ANN weights was 16.33% when it was among two missing values and 14.31% when it was among three missing values in a case. However, if we compare the

number of cases missing values for *lbloodp* when it was among two or three missing values, we see that there was a similar number of cases missing: 97 versus 105 cases. This may account for the similar error even though the number of missing values increased. Overall, the percent errors were similar amongst the three hybrid weight combinations and it was difficult to identify one approach as being significantly better than the others, at least using this particular test.

This artificial database represented a “worst-case scenario” with respect to the errors that occurred when the missing values were imputed, because the values were randomly deleted. According to the SNAP Guidelines, we know that only relevant data in the NICU database were collected and data not recorded were likely “normal.” Since the cases in the artificial data set all belonged to the same patient population, the missing data in this artificial setting did not follow the assumption of the SNAP Guidelines because in this case the missing data were relevant to the patient’s profile. These factors may have affected which cases the CBR identified as the closest-matching cases because the CBR was not basing the matches on all of the relevant data that were actually collected. This may account for some of the error that occurred when the missing values were imputed in the artificial data sets for the ANN-CBR hybrid approaches.

7.1.5 Neural Network Results with Imputed Values in Artificial Data Sets

The ANN experiments were executed to observe any changes in classification performance with the imputed data. Two-layer (linear) networks were used for a direct comparison with the optimized networks that were used to extract the weights for CBR matching (same network parameter settings as preliminary experiments). The results of those experiments are displayed in Table 7-7.

When the neural network was stopped at the same point that the best test set

Table 7-7. Step 1 test set results of linear ANN experiments with imputed data ($N = 5102$); training stopped at epoch 846

Performance measure	True values	ANN weights	Clinician weights	Uniform weights	Mean	Random
Sensitivity (%)	25.7	15.1	14.5	14.5	15.6	16.2
Specificity (%)	97.6	98.2	98.2	98.3	98.1	98.1
Classification rate (%)	90.2	89.7	89.6	89.7	89.7	89.7
Area under ROC curve	0.8290	0.7372	0.7353	0.7404	0.7280	0.7292

Table 7-8. Step 1 test set results of linear ANN experiments with imputed data ($N = 5102$); best test set performance

Performance measure	True values	ANN weights	Clinician weights	Uniform weights	Mean	Random
Sensitivity (%)	25.7	24.0	24.0	23.5	22.3	22.9
Specificity (%)	97.6	96.8	96.8	97.0	97.4	96.9
Classification rate (%)	90.2	89.3	89.3	89.0	89.7	89.3
Area under ROC curve	0.8290	0.7786	0.7697	0.7814	0.7815	0.7701

in Table 7-8. The sensitivity for the hybrid approaches was slightly higher than that of the mean and random approaches. Again, none clearly outperformed the rest at this point, although it is important to note that the hybrid system's imputation techniques performed equally well or slightly better than the mean and random approaches.

The experiments performed in Step 1 involved a relatively homogenous population of patients. Patients with complete data for the SNAPPE-II variables ($N = 5102$) have a mortality rate of 9.5% ($N = 484$ deaths) whereas the remaining 14325 cases (which have at least one missing value among the SNAPPE-II variables) have a mortality rate of 1.7% ($N = 243$ deaths). Clearly, these are two very different patient populations. Although the mean and random imputation approaches appeared to work quite well with the SNAPPE-II complete case patients, it was suspected that when the less ill patients were included in the database, their classification performance would degrade. This degradation was not anticipated for the hybrid imputation approach because the missing values were gradually imputed and were based on the mean of the ten closest-matching cases which was expected to be better than imputing the mean of the entire database or simply a random value.

7.2 Step 2 Results: Vertical Expansion of the Database

This step involved the imputation of unknown missing values in the SNAPPE-II data set using the weights extracted from the ANN in Step 1 as the match weights for the

CBR. Only the complete cases for the SNAPPE-II variables were used in the match set ($N = 5102$), and cases with missing values were consecutively imputed starting with those missing one value, then two, three and four missing values. Cases missing five values (i.e. those with data for only *seizure*, *bthwt*, *sga* and *apgar5*, $N = 24$) were excluded because there was insufficient information about the cases. This resulted in an imputed/complete case database of 19403 cases for these experiments.

To verify the assumption that consecutively imputing the cases with an increasing number of missing values would gradually add cases with lower severity of illness, the statistics of the match sets as they changed when new cases were added are shown in Table 7-9. Notice that as more cases with missing values are added to the database, the means of nearly all of the variables tend towards values of less severity of illness, according to the SNAP guidelines.

Once again, these imputed databases were separated into training ($N = 12784$ cases) and test ($N = 6619$ cases) sets for the subsequent ANN experiments. The same random separation key was used to allow for better comparability of the results. The results of these linear ANN experiments are presented in Table 7-10 (optimal network parameter settings are recorded in Table E-2 of Appendix E).

Comparing the sensitivity of the five models, we see that the models that used mean and random values to replace missing values have the lowest sensitivity of 35.2%, meaning that they correctly classified fewer patients who died than the hybrid

Table 7-9. Mean values and standard deviations of Step 2 databases

Data set	No. of cases	<i>po2fio2r</i>	<i>lurine</i>	<i>lserum</i>	<i>lbloodp</i>
No MVs	5102	1.85±1.25	0.84±0.81	7.30±0.12	33.07±8.29
ANN weights					
+ 1 MV	8952	1.93±1.14	0.88±0.79	7.31±0.11	34.65±8.83
+ 2 MVs	13231	1.98±0.99	0.91±0.77	7.32±0.10	36.43±9.40
+ 3 MVs	16934	2.00±0.90	0.92±0.72	7.32±0.09	37.90±9.88
+ 4 MVs	19403	2.01±0.85	0.92±0.67	7.32±0.09	38.24±9.42
Clinician weights					
+ 1 MV	8952	1.92±1.13	0.88±0.79	7.31±0.11	34.65±8.83
+ 2 MVs	13231	1.96±0.98	0.90±0.77	7.32±0.10	36.43±9.40
+ 3 MVs	16934	1.98±0.89	0.91±0.71	7.32±0.09	37.90±9.88
+ 4 MVs	19403	1.99±0.84	0.90±0.67	7.32±0.09	38.22±9.42
Uniform weights					
+ 1 MV	8952	1.93±1.14	0.88±0.79	7.31±0.11	34.65±8.83
+ 2 MVs	13231	1.99±0.99	0.91±0.77	7.32±0.10	36.43±9.40
+ 3 MVs	16934	2.02±0.90	0.92±0.72	7.32±0.09	37.90±9.88
+ 4 MVs	19403	2.02±0.85	0.92±0.68	7.32±0.09	38.22±9.42
Mean	19403	1.90±0.76	0.94±0.63	7.32±0.09	37.91±9.17
imputation					
Random	19403	2.06±1.25	1.09±0.82	7.32±0.11	37.91±9.99
imputation					

imputation models. The model imputed using uniform weights had a sensitivity of 40.1% which was better than the benchmark approaches, but not as good as the model using the ANN weights (44.5%) or the clinician weights (46.3%). All five models had approximately the same specificity ranging from 97.3% to 98.7%. These specificity values indicate that less than 3% of patients who survived were incorrectly classified as having died, which is important in a clinical setting. The constant predictor for this database, based on the test set mortality rate of 3.4%, was 96.6%, whereas the correct classification rates for the imputation models ranged from 95.7% to 96.5%. The areas

Table 7-10. Step 2 test set results of linear ANN experiments with imputed data for SNAPPE-II variables (test set mortality rate = 3.4%, $N = 19427$)

Performance measure	ANN weights	Clinician weights	Uniform weights	Mean	Random
Sensitivity (%)	44.5	46.3	40.1	35.2	35.2
Specificity (%)	97.7	97.4	98.3	98.6	98.7
Classification rate (%)	95.9	95.6	96.3	96.5	96.5
Area under ROC curve	0.8659	0.8561	0.8627	0.8296	0.8587

under the ROC curve ranged from 0.8296 for the mean imputation to 0.8659 for the ANN-weights model.

These experiments show that the ANN-CBR hybrid imputation system achieved slightly better overall classification performance than the benchmark approaches of mean and random imputation. Among the hybrid models, the model using the ANN weights for matching cases with the CBR had the best classification performance based on the area under the ROC curve. Also, compared to the classification performance of the Step 1 models (Table 7-8) that used only complete cases with the SNAPPE-II variables, there is an improvement in the classification performance with the sensitivity nearly doubling from approximately 24% to approximately 44% while maintaining the same specificity of approximately 97%. There was also an improvement in the area under the ROC curve from near 0.77 (0.8290 for the true values) to approximately 0.87. These results suggest that overall these approaches for replacing missing values make it easier for the neural network to identify patients who die despite the fact that the mortality rate is lower for this larger data set (9.5% for the

complete cases versus 3.4% for the entire database), which usually makes it more difficult for the ANN to identify rare cases.

7.3 Step 3 Results: Horizontal Expansion of the Database and Development of New Neonatal Mortality Model

The objective of Step 3 was to expand the hybrid system's imputation approach for replacing missing values so that more variables were included in addition to the SNAPPE-II variables. When these larger data sets were created and the missing values were imputed into the 5102 cases, two-layer neural network experiments with weight-elimination eliminated unimportant variables, thus refining the list of risk factors that predict neonatal mortality. This refined list of variables was used to impute the missing values into the remainder of the database ($N = 19427$), and the neural network's classification performance on the entire database was compared to that of the SNAP score models.

7.3.1 Creation of the Initial Expanded Data Sets

Using the base set of 5102 cases with complete data for the SNAPPE-II variables, variables with the fewest missing values were added to the database and cases missing those new variables were removed. This process was repeated one-by-one to gradually increase the number of variables that resulted in a complete set of cases that gradually got smaller, as presented in Table 7-11. Note that the mortality rate increases as the

Table 7-11. Missing values in the
SNAPPE-II database (N = 5102)

Parameter	Variable name	Percent missing	No. of cases	No. of variables	Data set
Lowest temperature	ltempf	0.0	5102	1	
Lowest mean blood pressure	lbloodp	0.0	5102	2	
Lowest serum pH	lserum	0.0	5102	3	
Lowest urine output	lurine	0.0	5102	4	
Lowest pO₂/FiO₂ ratio	po2fio2r	0.0	5102	5	
Presence of seizures	seizure	0.0	5102	6	
Birth weight	bthwt	0.0	5102	7	
Small for gestational age	sga	0.0	5102	8	
Apgar score at 5 minutes	apgar5	0.0	5102	9	
Presence of apnea	apnea	0.0	5102	10	
Stool guaiac	guaiac	0.0	5102	11	
Highest heart rate	hheartr	0.2	5102	12	
Lowest heart rate	lheartr	0.2	5102	13	
Highest respiratory rate	hrespr	0.3	5092	14	
Highest glucose	hgluc	8.4	4939	15	
Highest mean blood pressure	hbloodp	12.7	4934	16	
Lowest glucose	lgluc	13.9	4633	17	
Lowest white blood count	lwbc	17.0	4221	18	
Highest hematocrit	hhema	18.1	4153	19	Set A
Highest immature/total neutrophil ratio	hitnrat	28.0	3771	20	
Lowest absolute neutrophil count	lanc	28.1	3769	21	
Lowest platelet count	lplt	28.2	3539	22	
Lowest hematocrit	lhema	33.0	2915	23	
Highest pCO ₂	hpcO2	40.1	2908	24	Set B
Highest sodium	hsodium	59.8	1912	25	
Highest potassium	hpotass	62.9	1808	26	
Lowest pO ₂	lpo2po2	63.0	1807	27	
Lowest sodium	lsodium	66.0	1770	28	
Lowest potassium	lpotass	68.1	1766	29	
Highest oxygenation index	oi	72.6	1237	30	Set C

* Variables in bold belong to SNAPPE-II.

number of complete cases for the increasing number of variables increases. This agrees with the observations made by Ennett et al. [2001] that patients who have more values recorded are sicker than those who have fewer values recorded.

Three extended databases were chosen from this list of possible data sets:

- Set A (19 input variables, 4153 cases, mortality rate = 9.1%);
- Set B (24 input variables, 2908 cases, mortality rate = 9.3%); and
- Set C (30 input variables, 1237 cases, mortality rate = 11.9%).

These data sets were selected to adequately balance the number of cases with the number of variables in the data set. As mentioned in Section 2.1, a valid model must have sufficient information about the cases and variables. Having more variables than cases results in an under-specified model. In general, there should be at least ten times the number of cases as there are variables in the data set [Masters 1993]. Sets A, B and C meet this criterion.

7.3.2 Weight Extraction for CBR Imputation of Missing Values

Two-layer ANN experiments were conducted on Sets A, B and C to identify the network weights for extraction and use as the CBR match weights. These data sets were divided into training and test sets for the ANN experiments:

- Set A: 2724 training, 1429 test
- Set B: 1904 training, 1004 test
- Set C: 830 training, 407 test

Table 7-12 shows the normalized weights for the CBR for Sets A, B and C (optimal network parameter settings are recorded in Table E-3 of Appendix E). Recall that the

Table 7-12. Normalized Step 3 ANN weights for CBR match weights

Set A variables	Set A weights	Set B variables	Set B weights	Set C variables	Set C weights
lurine	100	po2fio2r	100	lurine	100
po2fio2r	99	lurine	75	guaiac	92
guaiac	90	hrespr	74	hrespr	78
hhema	85	guaiac	63	lpotass	53
hrespr	69	apgar5	60	lplt	47
sga	62	lplt	48	sga	47
lserum	44	sga	34	hgluc	42
ltempf	37	hpco2	33	lhema	41
hheartr	33	lserum	32	hpotass	30
apgar5	33	bthwt	31	apgar5	21
bthwt	25	hgluc	29	lbloodp	19
lbloodp	24	ltempf	29	hhema	17
apnea	16	hitnrat	29	lgluc	16
hbloodp	11	lgluc	25	bthwt	16
lheartr	11	lhema	23	hsodium	15
lgluc	10	hheartr	20	hitnrat	14
hgluc	6	lbloodp	18	lwbc	14
seizure	2	lheartr	16	lanc	13
lwbc	1	lanc	13	hbloodp	11
		lwbc	12	lheartr	11
		hbloodp	10	hheartr	10
		apnea	8	oi	9
		seizure	4	apnea	8
		hhema	4	ltempf	6
				seizure	6
				lsodium	3
				lpo2po2	3
				lserum	2
				po2fio2r	1
				hpco2	1

* Variables in bold belong to SNAPPE-II.

SNAPPE-II scoring system was developed on the entire Canadian Neonatal Network database where the majority of patients were significantly less sick than the populations in Sets A, B and C. This may account for the dilution of the importance of the

SNAPPE-II variables within these data sets as more input variables are included in the model.

In particular, look at the relative position of *po2fio2r* in each of the data sets in Table 7-12. In Sets A and B, the relative importance of this variable is 99 or 100% (as it has been in all of the other ANN models), whereas in Set C it is nearly the least important variable in the data set. A possible explanation for this dramatic drop in relevance is that each of these models is developed on a different population of patients. As more variables were added to the database and cases with missing values were excluded, the patient population became gradually sicker and a different set of variables played an important role in predicting mortality. Based on previous experiments, we know that *po2fio2r* is very important for predicting mortality for most patients, however, for the sickest patients (those used to develop the model for Set C) other factors may be more significant. This finding provides evidence that a model developed using only the sickest patients (i.e. those with no missing data) may not provide the best model for predicting the same outcome on a population with a lower severity of illness.

As in Steps 1 and 2, the missing values were imputed by consecutively adding cases missing one value, then two, three, etc. so that the database contained complete cases for all 30 variables in the 5102 cases, according to the variable-wise horizontal expansion approach. These sets were then separated again into training and test sets using the same random separation key as in Step 1.

7.3.3 Development of New Mortality Model

Two-layer ANNs were executed for the hybrid system data set with 30 input variables ($N = 5102$ cases) that were imputed using the ANN weights and the weights were normalized to identify the influence of each of the variables on mortality prediction. Variables with the smallest weights were eliminated from the database and the two-layer networks were re-run to see the impact of removing these variables. When the classification performance began to degrade, the weights of this final set of input variables were extracted for use in the CBR. Table 7-13 shows the progression of the variable elimination as the neonatal mortality model became more refined (optimal network parameter settings are recorded in Table E-4 of Appendix E). Notice that *po2fio2r* resumed its position as one of the most influential variables for predicting mortality. In general, the variables maintained their importance as other variables were eliminated. The new neonatal mortality model created using the ANN-CBR hybrid system contained thirteen input variables: *po2fio2r*, *lurine*, *lserum*, *apgar5*, *lplt*, *sga*, *hsodium*, *brespr*, *hpcO2*, *bthwt*, *lgluc*, *ltempf*, and *hbloodp*.

The missing values in the remainder of the entire database ($N = 19427 - 5102 = 14325$ cases) were imputed using the ANN weights as the CBR match weights for the remaining thirteen variables listed in Table 7-13. Once again, the cases with the fewest missing values were imputed first and then added to the match set before imputing the missing values in cases with more missing data. The resultant database contained 19427 cases with complete/imputed data for all thirteen of the ANN-CBR hybrid

Table 7-13. Progression of variables deleted for development of new neonatal mortality model ($N = 5102$)

30 inputs		22 inputs		16 inputs		14 inputs		13 inputs	
lurine	100	po2fio2r	100	po2fio2r	100	po2fio2r	100	po2fio2r	100
po2fio2r	89	lurine	81	lurine	57	lurine	84	lurine	72
lplt	59	lplt	60	apgar5	38	apgar5	55	lserum	36
bthwt	56	bthwt	50	sga	28	lserum	47	apgar5	33
hpcO2	54	lserum	41	hsodium	27	hrespr	46	lplt	29
lserum	52	hpcO2	39	lserum	26	sga	41	sga	28
hrespr	49	apgar5	38	bthwt	23	hsodium	37	hsodium	28
lwbc	44	hrespr	36	lgluc	21	lplt	34	hrespr	24
apgar5	40	hbloodp	35	hrespr	20	hpcO2	29	hpcO2	22
lanc	39	guaiaic	33	ltempf	17	bthwt	28	bthwt	21
guaiaic	37	sga	29	lplt	16	ltempf	28	lgluc	20
hheartr	34	hheartr	27	hpcO2	15	lgluc	27	ltempf	18
sga	34	lheartr	25	hbloodp	15	hbloodp	24	hbloodp	18
hbloodp	30	lgluc	21	hheartr	11	<i>hheartr</i>	16		
lheartr	28	hsodium	21	<i>guaiaic</i>	9				
lpo2po2	26	ltempf	21	<i>lheartr</i>	4				
lgluc	25	<i>lanc</i>	16						
hitnrat	23	<i>oi</i>	15						
ltempf	23	<i>hitnrat</i>	14						
oi	22	<i>lpo2po2</i>	11						
hsodium	20	<i>hpotass</i>	6						
hpotass	17	<i>lwbc</i>	3						
<i>lsodium</i>	10								
<i>lbloodp</i>	8								
<i>lpotass</i>	7								
<i>lhema</i>	6								
<i>apnea</i>	5								
<i>seizure</i>	3								
<i>hbema</i>	3								
<i>hgluc</i>	2								

* Variables in italics were eliminated from data set before next model iteration.

system's neonatal mortality model. This database was separated into training and test sets using the same random separation key as in Step 2 (with the remaining 24 cases added to the training set), and three-layer neural network experiments were performed.

Table 7-14. List of risk factors for predicting neonatal mortality

Hybrid model	SNAPPE-II
Po2/FiO2 ratio	Po2/FiO2 ratio
Lowest urine output	Lowest urine output
Lowest serum pH	Lowest serum pH
Apgar score at 5 minutes	Apgar score at 5 minutes
Lowest platelet count	Small for gestational age status
Small for gestational age status	Birth weight
Highest sodium concentration	Lowest temperature
Highest respiratory rate	Lowest blood pressure
Highest pCO2 reading	Multiple seizures
Birth weight	
Lowest glucose concentration	
Lowest temperature	
Highest blood pressure	

When we examine the list of risk factors selected by the hybrid model as the variables that have the greatest influence on the prediction of mortality in the NICU, shown in Table 7-14, we notice that the majority of the variables included in SNAPPE-II are also included in the hybrid system's model. The only exceptions are *seizure* and *lbloodp*. As mentioned in Section 5.3.2, *seizure* was a peculiar variable for inclusion as a risk factor for mortality due to its rare occurrence in the database (only 1.5 % of patients had multiple seizures). In nearly all of the neural network experiments, *seizure* was among the least important variables, and eventually it was excluded from the model using the neural network with weight-elimination.

Although *lbloodp* was excluded from the hybrid system's mortality model, *hbloodp* was included in its place. *Lbloodp* and *hbloodp* actually refer to the *lowest* blood pressure recorded during the first 12 hours in the NICU and the *highest* blood pressure recorded

during the first 12 hours in the NICU (as defined by the SNAP data collection guidelines), as opposed to patients who would have low blood pressure or high blood pressure. When viewed from this perspective, it is possible to see that *bbloodp* may reflect abnormally low values that were recorded as the highest blood pressure in the 12-hour period, and thus have a similar impact as *lbloodp*.

The hybrid model included four variables that were not among the SNAPPE-II variables: lowest platelet count, highest sodium concentration, highest pCO₂ reading and lowest glucose concentration.

7.3.4 Comparison with Other Models in the Literature

7.3.4.1 SNAP Models

Table 7-15 shows the classification performance of this new mortality model compared to the SNAP score models (optimal network parameter settings shown in Table E-5 of Appendix E). To compare the SNAP models to the hybrid model, two situations were considered: (1) when the specificity of the models was approximately 98.9% (chosen because the ANN found this to be the point of optimal classification performance) and (2) when the sensitivity of the models was approximately 80.0% (chosen because other authors commonly report their model's performance at 80% sensitivity). These performance measures were found by varying the cutoff value used to classify the patients after the cases were categorized according to each specific SNAP score.

When the hybrid system's mortality model is compared to the SNAP scores based on a specificity of approximately 98.9%, we see that the hybrid model has a higher sensitivity rate (38.3%) than any of the SNAP models (i.e. SNAP 23.0%, SNAPPE 23.4%, SNAP-II 30.4% and SNAPPE-II 31.6%). As well, when we consider a sensitivity of approximately 80.0%, the specificity of the hybrid model (84.4%) is higher than those of the SNAP and SNAP-II scores (70.4% and 77.8%, respectively), but similar to SNAPPE and SNAPPE-II (84.1% and 85.4%). The inclusion of *bthwt*, *sga* and *apgar5* in all three of these models may account for their consistently better classification performance than the basic SNAP scores that exclude these variables.

Table 7-15. Step 3 test set results for 13 input variables – new Neonatal Mortality Model

Performance measure	Hybrid model	SNAP cutoff =30	SNAPPE cutoff =56	SNAP-II cutoff =42	SNAPPE-II cutoff =65
SPECIFICITY=98.9%					
Sensitivity (%)	38.3	23.0	23.4	30.4	31.6
Specificity (%)	98.9	98.7	98.8	98.9	98.9
Classification rate (%)	96.8	95.9	96.0	96.3	96.4
SENSITIVITY=80.0%					
		Cutoff =15	Cutoff =24	Cutoff =13	Cutoff =28
Sensitivity (%)	80.1	80.6	79.1	79.9	79.5
Specificity (%)	84.4	70.4	84.1	77.8	85.4
Classification rate (%)	84.2	70.8	84.0	77.9	85.2
Area under ROC curve	0.8699				0.89±0.01 ^a 0.91±0.01 ^b

^aFrom Richardson et al. [2001], test set performance, $N = 5588$ cases.

^bFrom Richardson et al. [2001], training set performance, $N = 10819$ cases.

The SNAPPE-II model had a higher area under the ROC curve compared to the hybrid model (0.89 ± 0.01 versus 0.8699, respectively).

Confirmation of Hypothesis #2

These results confirm Hypothesis #2 because the ANN-CBR hybrid model predicted mortality in the NICU better than the SNAP and SNAP-II models, and equally as well as the SNAPPE and SNAPPE-II models.

7.3.4.2 Very Low Birth Weight Model

Zernikow et al. [1998] developed a model specific to preterm neonates (<32 weeks gestational age and/or <1500 g birth weight). Their patient population of very low birth weight infants had a mortality rate of 8.7% in the training set and 7.9% in the test set. The area under the ROC curve for Zernikow et al.'s model was 0.954 ± 0.011 and the specificity at 80% sensitivity was 97.5% (89.8% to 99.6%, 95% confidence interval). These performance measures are higher than those achieved with the hybrid system's model, however, Zernikow et al. used a more homogenous database (and a much smaller database with only 890 infants) than was used to develop the hybrid model. A possible reason that Zernikow et al. chose this patient population was because there were no missing values in the database. This obstacle was challenged with the development of the ANN-CBR hybrid system's model by imputing missing values using the mean of the closest-matching cases. A direct comparison of the hybrid

model's classification performance on only the patients that match Zernikow et al.'s population would be interesting and has been left as future work.

7.3.4.3 Extremely Low Birth Weight Model

The neural network model developed by Ambalavanan & Carlo [2001] for extremely low birth weight babies (<1000 g) had a specificity of 85% when the sensitivity was 80%. The area under the ROC curve for their model was 0.87 ± 0.03 . The hybrid system's model has similar performance measure values of 84.4% specificity at 80% sensitivity and an area under the ROC curve of 0.8699. The main difference between the two models is that Ambalavanan & Carlo's model is only valid for infants who birth weight was less than 1000g, whereas the hybrid system's model is valid for infants of all birth weights. Ambalavanan & Carlo's model was developed on a patient population of much sicker infants whose mortality rate was 34% in the training set and 33% in the test set, compared to the overall mortality rate of the Canadian Neonatal Network's database of 3.4%. Therefore, the hybrid system's model can be used on a broader range of patients.

7.3.4.4 MASc Thesis Work by Tong [2000]

Previous attempts to predict mortality using an ANN with this NICU database completed by Tong [2000] were unsuccessful. Despite attempts to replace the missing values with the mean, median or normal values, Tong was unable to develop a neural network model that could identify patients who died. This current thesis work showed

that an ANN built using mean and random imputation of missing data in the SNAPPE-II model was possible, although its classification performance was not as good as the models developed using the hybrid system. The main difference between Tong's attempts and the models presented in this thesis was the use of the log-sensitivity stopping criterion (developed by Scales [2001]). Tong used the maximum correct classification rate, which naturally steered the learning of the ANN toward classifying all cases as survivors. This led to the highest correct classification rate for that ANN due to the skewness of the database with less than 4% of patients dying, despite the fact that none of the nonsurvivors were correctly classified.

7.4 General Discussion of Results

7.4.1 Prediction for Groups Versus Individuals

The SNAP scores result in a "value" (say, 14) indicating a patient's severity of illness. Certainly, severity of illness has been shown to be linked to neonatal mortality risk [Richardson et al. 1993a, 1993b, 2001], but the SNAP scores don't actually predict mortality for individual patients. In practice, these scoring systems are used to compare the mortality rates of "Hospital A" and "Hospital B" for a group of patients, rather than for individuals.

On the other hand, the output of a neural network can be binary (for example, either survival or death). The ANN predicts outcomes for individual cases, therefore, mortality can be predicted for one infant at a time. This aspect of outcome prediction

using neural networks is advantageous for implementation as part of a decision support system.

7.4.2 Neural Network Performance for SNAPPE-II Variables Versus Hybrid Model

If we examine Tables 7-10 and 7-15, we notice that the neural network classification performance on the nine SNAPPE-II variables and the thirteen variables from the hybrid model was similar. This leads to the question of why add four more variables if they have no impact on the classification performance?

The good performance with the SNAPPE-II variables may have been related to the method of missing value imputation rather than the nine input variables on their own. The hybrid imputation approach may have imputed values that made it easier to the ANN to distinguish survivors from nonsurvivors. The experiments from Table 7-10 used the continuous variables, whereas the SNAPPE-II score categorizes each variable to achieve an additive score for each patient identifying mortality risk. When the ANN was trained and tested with the categorized variables for SNAPPE-II (with missing values coded as zero), it was not possible to get the neural network to identify any nonsurvivors (i.e. all cases were classified as survivors). In future work when outcomes other than mortality are investigated, we may find that the hybrid model is better for predicting other rare outcomes or that it works on a broader population of patients.

8 Concluding Remarks

To wind up the successful completion of this thesis work, this chapter summarizes the conclusions drawn from the results presented in Chapter 7, specifies the contributions to knowledge based on the thesis results and outlines ideas that developed during the process of completing this thesis that should be continued by future researchers.

8.1 Conclusions

This thesis work tested two hypotheses and both were confirmed. The following points highlight the major findings of this thesis work.

1. **A new approach for replacing missing values using an ANN-CBR hybrid system was successfully implemented.**
 - This system used the weights of a linear neural network as the match weights of the case-based reasoner to impute missing data using the mean of the ten closest-matching cases.

2. The hybrid system's imputation approach was validated on a data set of known missing values using the SNAPPE-II variable set ($N = 5102$ cases) to confirm that the imputed values were not significantly different from the true values.

- The average values of the data imputed using the hybrid system were not significantly different from the true values. The hybrid-imputed values were as close to the actual values as those found by mean imputation and were closer to the actual values than those found by random imputation.

CONFIRMATION OF HYPOTHESIS #1.

- Overall, the average error of the imputed data for the hybrid system models was less than the average error that occurred when mean or random values were used to replace the missing data.

3. The imputation approach using the hybrid system was validated on a data set of unknown missing values as the SNAPPE-II database was expanded vertically to include patients missing one to four of the SNAPPE-II variables, thereby increasing the database from 5102 cases to 19403 cases.

- By gradually imputing the missing values in cases missing one value, then two, three and four values, the match set was slowly enriched with cases

with lower severity of illness to provide better matches for patients of similar illness severity.

- When this imputed database was used to develop a linear ANN, the hybrid model that used the weights of the original linear neural network (“ANN weights”) achieved the highest area under the ROC curve (0.8659) and high sensitivity, specificity and classification rates (44.5%, 97.7% and 95.9%, respectively).
 - All three of the hybrid system models achieved better classification performance than the mean and randomly imputed models.
 - Compared to the classification performance of the preliminary linear ANN developed on only the cases with complete data for the SNAPPE-II variables ($N = 5102$), the ANN model developed using the hybrid-imputed data ($N = 19403$) provided an improvement in the classification performance with the sensitivity nearly doubling from approximately 24% to approximately 44% while maintaining the same specificity of approximately 96%. The area under the ROC curve was also better using the imputed database at 0.86 compared to 0.77 for the preliminary model.
4. **The hybrid imputation approach successfully increased the number of cases with complete data for more input variables in addition to the SNAPPE-II variables by expanding the database “horizontally” to replace missing values.**

- Using the three new data sets with 19, 24 and 30 input variables, respectively, to gradually impute missing values into the SNAPPE-II database of 5102 cases, a data set of 5102 cases with no missing values for 30 of the SNAPPE variables was created.
- 5. A new neonatal mortality model was developed that identified 13 input variables as important predictors of neonatal mortality.**
- The number of input variables in the data set with 5102 cases with no missing values for 30 input variables was successively refined using a neural network with weight-elimination to identify variables that contributed less to the prediction of mortality. This data set of 30 variables was reduced to 13 input variables that were found to be important for classifying patients with respect to neonatal mortality for babies of all degrees of illness severity.
 - The refined list of 13 risk factors identified using the hybrid system included the SNAPPE-II variables with the exception of *seizure* and *lbloodp*.
- 6. The hybrid system's neonatal mortality model achieved equal or better classification performance than the SNAP score models for all types of patients, and was comparable to other models developed on more homogenous patient populations.**

- The missing values in the remainder of the Canadian Neonatal Network database were imputed for these 13 input variables and a neural network was developed that achieved an area under the ROC curve of 0.8699, sensitivity of 38.3%, specificity of 98.9% and a correct classification rate of 96.8%. **CONFIRMATION OF HYPOTHESIS #2.**

8.2 Academic Contributions to New Knowledge

1. Development of a new method of imputing missing values by integrating a neural network and a case-based reasoner that benefits from the neural network's strengths in pattern recognition and feature extraction and from the case-based reasoner's ability to identify similar cases despite missing data.
2. Application of this new ANN-CBR hybrid system that facilitates a flexible approach to impute missing values and offers an improvement over commonly-used data imputation techniques like mean and random imputation.
3. Development of a neonatal mortality model that is valid for infants of all birth weights rather than just a narrow range of patients using the ANN-CBR hybrid system with a new subset of 13 input variables from the SNAP score: *po2fio2r*, *lurine*, *lserum*, *apgar5*, *lplt*, *sga*, *hsodium*, *hrespr*, *hpcO2*, *bthwt*, *lgluc*, *ltempf*, *hbloodp*.

4. Further evidence of the modelling advantages of the log-sensitivity index as a stopping criterion in the neural network with weight-elimination that makes it possible to develop a neural network model for the Canadian Neonatal Network's database which had previously been unsuccessful because the mortality output was highly skewed in favour of survivors.
5. Discovery that the order of input variables for the neural network has virtually no effect on the final values of the network connection weights.

8.3 Future Research

1. Use the hybrid mortality model from this thesis work to predict other NICU outcomes such as length of stay, duration of artificial ventilation, intraventricular hemorrhage, chronic lung disease and other morbidity outcomes.
2. Impute missing values in Day 3, Day 14 and Day 28 (discharge) data using the approach outlined in this thesis work, and develop ANN models to attempt to improve classification performance with this data. Observe the impact of this time-varying data on the prediction of neonatal mortality.
3. Test the classification performance of the hybrid model on specific patient populations of infants <1000g birth weight, <1500g birth weight and <32 weeks gestational age. Compare the results directly with the results of SNAP,

SNAPPE, SNAP-II, SNAPPE-II, Zernikow et al. [1998] (very low birth weight infants) and Ambalavanan & Carlo [2001] (extremely low birth weight infants).

4. Using the ANN-CBR hybrid system's methodology outlined in this thesis, develop one model for very low birth weight babies and another model for extremely low birth weight infants.
5. Based on the hybrid neonatal mortality model with 13 input variables developed in this thesis work, create another ANN model that excludes *bthwt*, *sga* and *apgar5* (i.e. 10 input variables remaining). Observe the impact of these variables on the model's classification performance. Exclude each of the 13 input variables one at a time to observe the each variable's influence on the classification performance.
6. Consult a panel of experts to determine what weights they would assign to the SNAPPE-II variables for CBR matching. Compare matching with these expert-assigned weights to the matching approaches used in this thesis work.
7. For the CBR matching, experiment with different approaches for finding the closest-matching cases such as using the median value or some other approach for imputing the missing data.

8. Compare the matches made using *The Easy Reasoner* CBR system from Haley Enterprises with the k -nearest neighbours algorithm employed in the flexible CBR system developed by Cotea & Jiwani [2003], as mentioned in Section 5.3.1.4.
9. Create a fully-integrated version of the ANN-CBR hybrid system as described by Medsker [1994] in Section 2.3.5 to provide better compatibility between the two systems and reduce the need for manual database manipulation.
10. Extract the weights from a nonlinear neural network using the four techniques presented by Shin et al. [2000] for use as the match weights in the CBR to replace the missing values and compare the imputation results with those found in this thesis.
11. Develop a model using the packet loss recovery techniques described in Section 3.8 for comparison with the ANN-CBR hybrid system's model for replacing missing values in the NICU database.
12. Apply this ANN-CBR hybrid imputation approach for replacing missing values to new databases: medical and non-medical.

References

- Aamodt A, Plaza E. "Case-based reasoning: Foundations and issues, methodological variations, and system approaches." *AI Communications* 1994;7(i);39-59.
- Abeyratne UR, Zhang G, Saratchandran P. "EEG source localization: comparative study of classical and neural network methods." *Int J Neural Syst* 2001 Aug;11(4):349-59.
- Adeney KM, Ennett CM, Frize M, Korenberg MJ. "Prediction of patient outcomes in an intensive care unit." *Proc IJCNN Symp* 1999:7-12.
- Ambalavanan N, Carlo WA. "Comparison of the prediction of extremely low birth weight neonatal mortality by regression analysis and by neural networks." *Early Hum Dev* 2001;65:123-37.
- American Academy of Pediatrics. "Policy statement: Use and abuse of the Apgar score." *Pediatrics* 1996;98(1):141-142.
- Anderson D, McNeill G. "Artificial neural networks technology." Data & Analysis Center for Software, Rome Laboratory, Griffiss AFB, NY. 1992, pp. 83.
- Apgar V. "A proposal for a new method of evaluation of the newborn infant." *Curr Res Anesth Analg* 1953;32:260-267.

- Apgar V, Holaday DA, James LS, Weisbrot IM, Berrien C. "Evaluation of the newborn infant: second report." *JAMA* 1958;168:1985-1988.
- BabyCenter. "Labor and delivery: The Apgar score."
<http://www.babycenter.com/refcap/3074.html> (Accessed March 27, 2003).
- Baxt WG. "Complexity, chaos and human physiology: the justification for non-linear neural computational analysis." *Cancer Lett* 1994;77:85-93.
- Becker L, Jazayeri K. "A connectionist approach to case-based reasoning." *Proc Case-Based Reasoning Workshop*, Morgan Kaufmann, 1989;:213-217.
- Birnbaum L, Collins G. "Reminders and engineering design themes: A case study in indexing vocabulary." *Proc 2nd Workshop on Case-Based Reasoning*, Pensacola Beach, FL. 1989.
- Blum A. "Neural Networks in C++." John Wiley & Sons, Inc.: New York, 1992.
- Buchman TG, Kubos KL, Seidler AJ, Siegforth MJ. "A comparison of statistical and connectionist models for the prediction of chronicity in a surgical intensive care unit." *Crit Care Med* 1994;22(5):750-762.
- Buskard TT. "Estimating duration of ventilation, length of stay and mortality in the intensive care unit using artificial neural networks." Master's thesis, Department of Electrical Engineering, University of New Brunswick, Fredericton, NB, 1994.
- Catley C. "Medical database management and interoperability for neural network applications: An XML-based solution." Master's thesis, Department of Systems and Computer Engineering, Carleton University, Ottawa, ON, 2002.

- Catley C, Frize M. "Design of a health care architecture for medical data interoperability and application integration." *Proc IEEE EMBS- BMES* 2002;1952-1953.
- Catley C, Frize M. "A prototype XML-based implementation of an integrated 'intelligent' neonatal intensive care unit." *Proc IEEE-ITAB* 2003.
- Catley C, Frize M, Walker CR, St. Germain L. "Integrating customizable clinical alerts into an XML-based health care framework for the NICU." *Proc IEEE-EMBS* 2003.
- Chen D, Burrell P. "Case-based reasoning system and artificial neural networks: A review." *Neural Comput & Applic* 2001;10:264-276.
- Chien L-Y, Whyte R, Thiessen P, Walker R, Brabyn D, Lee SK, The Canadian Neonatal Network. "SNAP-II predicts severe intraventricular hemorrhage and chronic lung disease in the neonatal intensive care unit." *J Perinatol* 2002; 22:26-30.
- Consortium for Virtual Operations Research (CVOR). Boutell T, maintainer. West Virginia University, Penn State University, 1997.
<http://cvor.pe.wvu.edu/faq/nnfaq.htm>
- Cotea C, Jiwani S. "A case-based reasoning system for the neonatal intensive care unit." Term project, School of Information Technology and Engineering (Computer Science), University of Ottawa 2003.

- Cullen DJ, Civetta JM, Briggs BA, Ferrara LC. "Therapeutic intervention scoring system: a method for quantitative comparison of patient care." *Crit Care Med* 1974 Mar-Apr;2(2):57-60.
- Demuth H, Beale M. Neural Network Toolbox User's Guide, Version 3.0. The MathWorks Inc.: Natick, MA, 1997.
- Ennett CM. "Coronary artery surgery mortality prediction using artificial neural networks." Master's thesis, School of Information Technology and Engineering, Electrical Engineering, University of Ottawa, Ottawa, ON, Dec 1999.
- Ennett CM, Frize M. "An investigation into the strengths and limitations of artificial neural networks: an application to an adult ICU patient database." *Proc AMLA Symp* 1998:998.
- Ennett CM, Frize M. "Selective sampling to overcome skewed a priori probabilities with neural networks." *Proc AMLA Symp* 2000;(20 Suppl):225-9.
- Ennett CM, Frize M. "Weight-elimination neural networks for mortality prediction in coronary artery surgery." *IEEE Trans Info Technol Biomedicine* 2003 Jun;7(2):86-92.
- Ennett CM, Frize M, Shaw RE. "Methodologies for predicting coronary surgery outcomes." *Proc IEEE EMBS-BMES* 1999.
- Ennett CM, Frize M, Walker CR. "Influence of missing values on artificial neural network performance." *Medinfo* 2001;10(Pt 1):449-53.

- Ennett CM, Frize M, Scales N. "Logarithmic-sensitivity index as a stopping criterion for neural networks." *Proc IEEE EMBS-BMES* 2002.
- Ennett CM, Frize M, Scales N. "Evaluation of the logarithmic-sensitivity index as a neural network stopping criterion for rare outcomes." *Proc IEEE ITAB* 2003.
- Fausett L. *Fundamentals of Neural Networks*. Prentice-Hall: Englewood Cliffs, NJ, 1994.
- Forsström JJ, Dalton KJ. "Artificial neural networks for decision support in clinical medicine." *Ann Medicine* 1995;27:509-517.
- Frize M, Ennett CM. "Improving the potential clinical significance of decision-support systems using artificial neural networks." *Proc AMLA Symp* 2000:1011.
- Frize M, Walker R. "Clinical decision-support systems for intensive care units using case-based reasoning." *Med Eng Phys* 2000 Nov;22(9):671-677.
- Frize M, Trigg HCE, Stevenson M, Solven FG. "Decision-support systems for critical care." *Proc AMLA Symp* 1997:CD-ROM
- Frize M, Wang L, Ennett CM, Nickerson B, Solven FG, Stevenson MH. "New advances and validation of knowledge management tools for critical care using classifier techniques." *Proc AMLA Symp* 1998:553-558.
- Frize M, Ennett CM, Charette E. "Automated optimization of the performance of artificial neural networks to estimate medical outcomes." *Proc IEEE ITAB-ITIS* 2000:168-73.

- Frize M, Ennett CM, Stevenson M, Trigg HCE. "Clinical decision-support systems for intensive care units using artificial neural networks." *Med Eng Phys* 2001 Apr; 23(3):217-25. [2001a]
- Frize M, Ennett CM, Hebert P. "Improving the efficiency and effectiveness of clinical decision-support systems using artificial neural networks." *Medinfo* 2001. [2001b]
- Gabrys B. "Pattern classification for incomplete data." *IEEE International Conference on Knowledge-based Intell Engineering Syst & Allied Technologies* 2000 Sep;1:454-457.
- Gabrys B, Bargiela A. "General fuzzy min-max neural network for clustering and classification." *IEEE Trans Neural Networks* 2000 May;11(3):769-783.
- Gray JE, Richardson DK, McCormick MC, Workmann-Daniels K, Goldman DA. "Neonatal therapeutic intervention scoring system: a therapy-based severity-of-illness index." *Pediatrics* 1992 Oct;90(4):561-7.
- Hammond KJ. "On functionally motivated vocabularies: An apologia." *Proc 2nd Workshop on Case-Based Reasoning*, Pensacola Beach, FL. 1989.
- Haque BU, Belecheanu RA, Barson RJ, Pawar KS. "Toward the application of case based reasoning to decision-making in concurrent product development (concurrent engineering)." *Knowledge-Based Systems* 2000; 13(2-3):101-112.
- Harrison I. "Case-based reasoning." Artificial Intelligence Applications Institute 1997. <http://www.aiai.ed.ac.uk/links/cbr.html>

Health Canada. "Perinatal health indicators for Canada: A resource manual."

Minister of Public Works and Government Services Canada, 2000.

<http://www.hc-sc.gc.ca/hpb/lcdc/brch/reprod/phic-isp/pdf/indperie.pdf>

Hornik K, Stinchcombe M, White H. "Multilayer feedforward networks are universal approximators." *Neural Networks* 1989;2:359-366.

Itchhaporia D, Snow PB, Almassy RJ, Oetgen WJ. "Artificial neural networks: Current status in cardiovascular medicine." *J Am Coll Cardiol* 1996;28(2):515-521.

International Neonatal Network. "The CRIB (clinical risk index for babies) score: a tool for assessing initial neonatal risk and comparing performance of neonatal intensive care units." *Lancet* 1993;342:193-8.

Kattan MW, Beck JR. "Artificial neural networks for medical classification decisions." *Arch Pathol Lab Med* 1995;119:672-677.

Keene AR, Cullen DJ. "Therapeutic intervention scoring system: update 1983." *Crit Care Med* 1983;11:1-3.

Knaus WA, Zimmerman JE, Wagner DP, Draper EA, Lawrence DE. "APACHE – acute physiology and chronic health evaluation: a physiologically based classification system." *Crit Care Med* 1981;9(8):591-7.

Kolodner JL. Case-Based Reasoning. Morgan Kaufmann: New York, 1993.

Krogh A, Hertz JA. "A simple weight decay can improve generalization." In: Lippmann RP, Moody J, Touretzky, eds. *Advances in Neural Information Processing Systems 4 (NIPS'91)*. Morgan Kaufmann: San Matteo, CA, 1992:950-957.

- Le Cun Y, Denker JS, Solla SA. "Optimal brain damage." In: Touretzky D, ed. *Advances in Neural Information Processing Systems 2 (NIPS'89)*. Morgan Kaufmann: Denver, CO, 1990;598-605.
- Lee SK, McMillan DD, Ohlsson A, Pendray M, Synnes A, Whyte R, Chien L-Y, Sale J. The Canadian NICU Network. "Variations in practice and outcomes of the Canadian NICU Network: 1996-1997." *Pediatrics* 2000;106:1070-9.
- Lin Z, Maris J, Hermans L, Vandewalle J, Chen JD. "Classification of normal and abnormal electrograms using multilayer feedforward neural networks." *Med Biol Eng Comput* 1997 May;35(3):199-206.
- Little RJA. "Methods for handling missing values in clinical trials." *J Rheumatology* 1999;26(8):1654-1656.
- Livingstone DJ, Manallack DT, Tetko IV. "Data modelling with neural networks: Advantages and limitations." *J Computer-Aided Molecular Design* 1997;11:135-142.
- Masters T. *Practical Neural Network Recipes in C++*. Academic Press: San Diego, CA, 1993.
- Masters T. *Advanced Algorithms for Neural Networks: A C++ Sourcebook*. John Wiley and Sons, Inc.: New York, NY, 1995.
- McMillan D, Chan HWP, Lee SK. "Canadian NICU Network 1996-1997, Volume 1." Canadian NICU Network, 1999.
- Medsker LR. "Hybrid neural network and expert systems." Kluwer Academic Publishers: Boston, MA, 1994.

- Myllymaki P, Tirri H. "Massively parallel case-based reasoning with probabilistic similarity metrics." *Proc 1st European workshop on Case-Based Reasoning*, Springer-Verlag, 1993;:144-154.
- Park JH, Shin C-K, Im KH, Park SC. "A local weighting method to the integration of neural network and case based reasoning." *Proc 2001 IEEE Signal Processing Society Workshop* 2001:33-42.
- Pesonen E, Eskelinen M, Juhola M. "Treatment of missing data values in a neural network based decision support system for acute abdominal pain." *Art Intell Med* 1998;13:139-46.
- Penny W, Frost D. "Neural networks in clinical medicine." *Med Decis Making* 1996;16:386-398.
- Perkins C, Hodson O, Hardman V. "A survey of packet loss recovery techniques for streaming audio." *IEEE Network* 1998;12(5):40-48.
- Reategui EB, Campbell JA, Leao BF. "Combining a neural network with case-based reasoning in a diagnostic system." *Artif Intell Med* 1997;9:5-27.
- Richardson DK, Gray JE, McCormick MC, Workmann K, Goldmann DA. "Score for neonatal acute physiology: a physiologic severity index for neonatal intensive care." *Pediatrics* 1993 Mar;91(3):617-23. [1993a]
- Richardson DK, Phibbs CS, Gray JE, McCormick MC, Workmann-Daniels K, Goldmann DA. "Birth weight and illness severity: independent predictors of neonatal mortality." *Pediatrics* 1993;91(5):969-75. [1993b]

- Richardson DK, Corcoran JD, Escobar GJ, Lee SK. "SNAP-II and SNAPPE-II: Simplified newborn illness severity and mortality risk scores." *J Pediatr* 2001;138:92-100.
- Riesbeck CK, Shank RC. "Inside Case-Based Reasoning." Lawrence Erlbaum Associates, Inc., New Jersey, 1989.
- Rumelhart DE, Hinton GE, Williams RJ. "Learning internal representations by error propagation." In DE Rumelhart, JL McClelland, eds, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol 1:318-362*. MIT Press: Cambridge, MA, 1986. [1986a]
- Rumelhart DE, Hinton GE, Williams RJ. "Learning representations by back-propagating errors." *Nature* 1986;323:533-536. [1986b]
- Rybchynski D. "Intelligent Decision Aid System (IDEAS) for NICU 4.0 Research Version; User's Manual (Draft)," University of Ottawa, March 2002.
- Scales N. "Mortality prediction in an ICU and coronary surgery database using neural networks." Term project, Department of Systems and Computer Engineering, Carleton University, Dec 2001.
- Schmidt R, Montani S, Bellazzi R, Portinale L, Gierl L. "Case-based reasoning for medical knowledge-based systems." *Int J Med Inf* 2001;64:355-367.
- Shin CK, Yun UT, Kim HK, Park SC. "A hybrid approach of neural network and memory-based learning to data mining." *IEEE Trans on Neural Networks* 2000 May;11(3):637-646.
- Skiena SS. "The Algorithm Design Manual," New York:Springer-Verlag, 1998.

Stevens SM, Richardson DK, Gray JE, Goldmann DA, McCormick MC.

“Estimating neonatal mortality risk: an analysis of clinicians’ judgements.”

Pediatrics 1994;93(6):945-950.

The Haley Enterprise. “The Easy Reasoner, Eclipse 3.2g Reference Manual.” The

Haley Enterprise Inc., Sewickley, PA, 1993.

The Haley Enterprise, Inc. “Reasoning with case bases.” [Online document] 1994.

www.haley.com

Tong H. Threshold models in non-linear time series analysis. New York:Springer-

Verlag, 1983.

Tong Y. “Developing ANN approaches to estimate neonatal ICU outcomes.” Master’s

thesis, School of Information Technology and Engineering, Electrical

Engineering, University of Ottawa, Ottawa, ON, May 2000.

Tong Y, Frize M, Walker R. “Estimating ventilation using artificial neural networks in

intensive care units.” *Proc IEEE EMBS-BMES* 1999.

Tong Y, Frize M, Walker R. “Extending successful predictions of ventilation needs

using artificial neural networks from adult to neonatal ICU patients.” *IEEE*

Trans Inf Technol Biomed 2002 Jun;6(2):188-191.

Trigg HCE. “An investigation of methods to enhance the performance of artificial

neural networks used to estimate ICU outcomes.” Master’s thesis, Department

of Electrical Engineering, University of New Brunswick, Fredericton, NB, Jan

1997.

- Walker CR [MBBS, FRCPC]. Chief Neonatologist, Children's Hospital of Eastern Ontario (CHEO), Ottawa Hospital – General Campus, Ottawa, ON. Personal communication, May 20, 2003.
- Walker CR [MBBS, FRCPC]. Chief Neonatologist, Children's Hospital of Eastern Ontario (CHEO), Ottawa Hospital – General Campus, Ottawa, ON. Personal communication, June 3, 2002.
- Walker CR [MBBS, FRCPC]. Chief Neonatologist, Children's Hospital of Eastern Ontario (CHEO), Ottawa Hospital – General Campus, Ottawa, ON. Personal communication, April 8, 2002.
- Walker CR [MBBS, FRCPC]. Chief Neonatologist, Children's Hospital of Eastern Ontario (CHEO), Ottawa Hospital – General Campus, Ottawa, ON. Personal communication, June 20, 2000.
- Walker CR, Ennett CM, Frize M. "Use of an artificial neural network to estimate probability of mortality and duration of ventilation in neonatal intensive care unit patients." *Medinfo* 2001.
- Walker CR, Frize M, Ennett CM. "Will artificial intelligence systems assist planning of resource allocation in the neonatal intensive care unit (NICU)?" *Paediatr Child Health* 2001;6:28A.
- Walker CR, Ennett CM, Frize M. "Will artificial intelligence systems help counselling of parents in the neonatal intensive care unit (NICU)?" *Paediatr Child Health* 2001;6:28A.

- Walker RC, Ennett C, Frize M. "Use of an artificial neural network to estimate probability of mortality in neonatal intensive care unit patients." *Pediatr Res* 2001;49:310A.
- Wang L. "Constructing case bases from medical databases." Master's thesis, Computer Science, University of New Brunswick, NB. 1997.
- Watson I, Marir F. "Case-based reasoning: A review." *The Knowledge Engineering Review* 1994;9(4) [Online document]. <http://www.ai-cbr.org/classroom/cbr-review.htm>.
- Weigend AS, Rumelhart DE, Huberman BA. "Back-propagation, weight-elimination and time series prediction." In DS Tourestzky, JL Elman, TJ Sejnowski, GE Hinton, eds. *Proc 1990 Connectionist Models Summer School*. Morgan Kaufmann: San Mateo. 1990:105-116.
- Weigend AS, Rumelhart DE, Huberman BA. "Generalization by weight-elimination with application to forecasting." In RP Lippmann, J Moody, DS Touretzky, eds. *Advances in Neural Information Processing Systems (NIPS'90), Volume 3*. Morgan Kaufmann: San Mateo. 1991;3:875-882.
- Werbos P. "Beyond regression: new tools for prediction and analysis in the behavioral sciences." PhD Thesis, Harvard University, Cambridge, MA. August 1974.
- Woods K, Bowyer KW. "Generating ROC curves for artificial neural networks." *IEEE Trans Medical Imaging* 1997;16(3):329-337.

- Yeh TS, Pollack MM, Ruttimann UE, Holdbrook PR, Fields AI. "Validation of a Physiologic Stability Index for use in critically ill infants and children." *Pediatr Res* 1984;18:445-451.
- Zernikow B, Holtmannspoetter K, Michel E, Pielemeier W, Hornschuh F, Westermann A, Hennecke KH. "Artificial neural network for risk assessment in preterm neonates." *Arch Dis Child Fetal Neonatal Ed* 1998;79:F129-F134.

Appendix A

Neonatal Scoring Systems

Apgar Score

Developed in 1952 by neonatologist Dr. Virginia Apgar, the Apgar score [Apgar 1953, Apgar et al. 1958] is measured at one, five and ten minutes after birth to quickly and easily assess a newborn's initial condition and response to any resuscitative efforts [AAP 1996]. Table A-1 shows the parameters that are assessed and scored a value of 0 to 2, summing to a maximum score of 10. A score of 7 to 10 is normal (it is rare to receive a one-minute Apgar score of 10 because newborns are usually somewhat pale from the delivery process). Infants scoring between 4 and 6 may require some minor interventions, but will likely recover quickly. A score under 4 indicates that the baby may require immediate resuscitative measures, however, there is still a possibility of good outcome. Prematurity and medication given to the mother may account for a lower Apgar score [BabyCenter 2003].

Table A-1. Apgar scoring system

	0	1	2
Appearance	Pale	Blue	Pink
Pulse	Absent	<100	>100
Grimace	Absent	Grimace	Cry active
Activity	Limp	Some tone	Active
Respiration	Absent	Irregular	Regular and cry

CRIB Score

The Clinical Risk Index for Babies (CRIB) was developed retrospectively on a population of 735 infants whose birth weight was 1500g or less, or whose gestational age was less than 31 weeks [INN 1993]. The data was collected from four UK tertiary-level hospitals between 1988-1990. The CRIB score uses data collected in the first 12 hours of birth to predict in-hospital death, and excluded infants with inevitably lethal congenital malformations. Table A-2 defines the parameter ranges and their associated scores for this additive model. CRIB predicted death in the validation set with 51% sensitivity at 95% specificity. This score was developed before surfactant therapy became widely used.

Table A-2. CRIB scoring system

Risk factor		Score
Birth weight (g)	> 1350	0
	851-1350	1
	701-850	4
	≤ 700	7
Gestation (week)	> 24	0
	≤ 24	1
Congenital malformations (excluding inevitably lethal malformations)	None	0
	Not acute life-threatening	1
	Acutely life-threatening	3
Maximum base excess in first 12 h (mmol/L)	> -7.0	0
	-7.0 to -9.9	1
	-10 to -14.9	2
	≥ -15.0	3
Minimum appropriate FiO ₂ in first 12 h	≤ 0.40	0
	0.41-0.60	2
	0.61-0.90	3
	0.91-1.00	4
Maximum appropriate FiO ₂ in first 12 h	≤ 0.40	0
	0.41-0.60	1
	0.61-0.90	3
	0.91-1.00	5

NTISS Score

The Neonatal Therapeutic Intervention Scoring System (NTISS) [Gray et al. 1992] was based on the adult-ICU score called the Therapeutic Intervention Scoring System (TISS) [Cullen et al. 1974] to predict in-hospital mortality. The score was developed using data from 1643 newborns admitted to 3 NICUs between November 1989 and September 1990, and excluded moribund patients. A detailed description of the variables measured using NTISS is shown in Table A-3.

Table A-3. NTISS scoring system

Item	Subscore	Item	Subscore
RESPIRATORY		MONITORING (con't)	
Supplemental oxygen	1 ^a	Thermoregulated environment	1
Surfactant administration	1	Noninvasive oxygen	1
Tracheostomy care	1 ^b	Arterial pressure	1
Tracheostomy placement	1 ^b	Central venous pressure	1
Continuous positive airway pressure administration	2 ^a	Urinary catheter	1
Endotracheal intubation	2	Quantitative intake and output	1
Mechanical ventilation	3 ^a	Extensive phlebotomy (>10 blood draws)	2 ^h
Mechanical ventilation with muscle relaxation	4 ^a	METABOLIC/NUTRITION	
High-frequency ventilation	4 ^a	Gavage feeding	1
Extracorporeal membrane oxygenation	4	Intravenous fat emulsion	1
CARDIOVASCULAR		Intravenous amino acid solution	1
Indomethacin administration	1	Phototherapy	1
Volume expansion (≤ 15 mL/kg)	1 ^c	Insulin administration	2
Vasopressor administration (1 agent)	2 ^d	Potassium infusion	3
Volume expansion (> 15 mL/kg)	3 ^c	TRANSFUSION	
Vasopressor administration (>1 agent)	3 ^d	Intravenous gamma globulin	1
Pacemaker on standby	3 ²	Red blood cell (≤ 15 mL/kg)	2 ⁱ
Pacemaker used	4 ^c	Partial volume exchange	2
Cardiopulmonary resuscitation	4	Red blood cell (>15 mL/kg)	3 ⁱ
DRUG THERAPY		Platelet	3
Antibiotic (≤ 2 agents)	1 ^f	White blood cell	3
Diuretic (enteral)	1 ^g	Double volume exchange	3
Steroid (postnatal)	1	PROCEDURAL	
Anticonvulsant	1	Transport of patient	2
Aminophylline	1	Single chest tube in place	2 ^j
Other unscheduled medication	1	Minor operation	2 ^k
Antibiotic (> 2 agents)	2 ^f	Multiple chest tubes in place	3 ^j
Diuretic (parenteral)	2 ^g	Thoracentesis	3
Treatment of metabolic acidosis	3	Major operation	4 ^k
Potassium binding resin	3	Pericardiocentesis	4 ^j
MONITORING		Pericardial tube in place	4 ^j
Frequent vital signs	1	Dialysis	4
Cardiorespiratory	1	VASCULAR ACCESS	
Phlebotomy (5-10 blood draws)	1 ^h	Peripheral intravenous line	1
		Arterial line	2
		Central venous line	2

* Superscript letters represent mutually exclusive variables.

Appendix B

SNAP Guidelines

**Taken from: McMillan D, Chan HWP, Lee SK. "Canadian NICU Network
1996-1997, Volume 1." Canadian NICU Network, 1999.**

General:

SNAP on day one should be scored from the time of admission (defined as the time of first vitals in the NICU) for twelve (12) hours. Day three is scored from admission date plus three days and goes from 6:00AM to 5:59AM on the following day (24 hours). Day fourteen is scored from admission date plus fourteen days and goes from 6:00AM to 5:59AM on the following day (24 hours). Values occurring during an operation SHOULD be included.

Vital Signs:

Vital signs recorded while a baby is agitated or crying should generally be recorded. However, if the value seems very abnormal, call the SCC for advice. Score any non-zero values that are recorded in the chart for babies who are dying.

Lab Values:

Lab values should be included in the scoring period if they are DRAWN during the scoring period. Time of draw should be taken from the flow sheet when this is explicitly recorded. If the time of the draw is not explicitly recorded on the flow sheet, assume the time the lab received the samples is within 15 minutes of the draw (in other words, include values listed as occurring within 15 minutes of the end of the scoring period).

Computer values should generally be considered more accurate than flow sheet/progress note values unless they are clearly being discounted by the clinicians. Lab values discounted by clinicians should not be recorded on SNAP. Hemolyzed values are acceptable EXCEPT for hemolyzed potassium which should be discounted. Pathology blood draws, cord specimens and other non-blood draws (CSF, urine) should NOT be scored on SNAP. Do not score "diluted" lab values.

Values Listed as a Range:

Vital signs and d-stix listed as a range should be scored as the midpoint. Vital signs and d-stix listed as "< a certain value" should be scored as one less than the value listed (e.g. A low temp of <94 should be scored as 93). Vital signs and d-stix listed as "> a certain value" should be scored as one more than the value listed (e.g. A d-stix high of >700 should be scored as 701).

Data Item	Definition
Mean blood pressure: high	Highest mean arterial pressure (MAP), also called mean blood pressure (MBP), during the TIME PERIOD, as recorded in

	the nursing flow sheet. <i>Arterial line pressures and cuff pressures should be weighted equally in choosing high/low values. If these values are very different though, ask in the NICU or call the SCC for advice.</i> If only systolic and diastolic are recorded, assume mean blood pressure = diastolic + 1/3 (systolic-diastolic). E.g. 55/43: $MBP=43+(1/3)(55-43)=47$.
Mean blood pressure: low	Lowest mean arterial pressure (MAP), also called mean blood pressure (MBP), during the TIME PERIOD, as recorded in the nursing flow sheet. <i>Arterial line pressures and cuff pressures should be weighted equally in choosing high/low values. If these values are very different though, ask in the NICU or call the SCC for advice.</i> If only systolic and diastolic are recorded, assume mean blood pressure = diastolic + 1/3 (systolic-diastolic).
Heart rate: high	Highest heart rate during the TIME PERIOD sustained for more than one minute continuously. Do not include transient heart rate values that reflect bradycardia associated with apnea/dsats.
Heart rate: low	Lowest heart rate during the TIME PERIOD sustained for more than one minute continuously. Do not include transient heart rate values that reflect bradycardia associated with apnea/dsats. <i>Do NOT score the low heart rate as zero (0) for babies who die during a scoring period.</i>
Respiratory rate: high	Highest respiratory rate sustained for more than one minute. Count spontaneous respirations only. If on ventilator with no breaths above the vent, score as zero (0). <i>At some site you may need to subtract the vent rate from the listed respiratory rate in order to find the number of spontaneous respirations.</i>
Temperature: low	Lowest body temperature (axillary or rectal but NOT skin probe which is the baby's temperature recorded through the isolette) recorded in Celsius.
Sodium (Na): high	Highest serum sodium (Na) during the TIME PERIOD. Normal is in the 130-145 mmol/L range.
Sodium (Na): low	Lowest serum sodium (Na) during the TIME PERIOD. Normal is in the 130-145 mmol/L range.
Potassium (K): high	Highest serum potassium (K) during the TIME PERIOD. Disregard hemolyzed potassium values. Normal values are in the 5-6.5 mmol/L range.
Potassium (K): low	Lowest serum potassium (K) during the TIME PERIOD. Disregard hemolyzed potassium values. Normal values are in the 5-6.5 mmol/L range.
HCO ₃ (bicarb): high	Highest bicarbonate (HCO ₃ , TC02) measured on electrolytes. Do NOT use calculated value derived from blood gas analysis.
HCO ₃ (bicarb):	Lowest bicarbonate (HCO ₃ , TC02) measured on electrolytes.

low	Do NOT use calculated value derived from blood gas analysis.
Calcium (total): high	Mutually exclusive with Calcium ionized: high. Highest total calcium (Ca) during TIME PERIOD. Total Ca is usually in the 1.5-2.7 range. Do NOT record ionized Ca here.
Calcium (total): low	Mutually exclusive with Calcium ionized: low. Lowest total calcium (Ca) during TIME PERIOD. Total Ca is usually in the 1.5-2.7 range. Do NOT record ionized Ca here.
Calcium (ionized): high	Mutually exclusive with Calcium total: high. Highest ionized calcium (Ca) during TIME PERIOD. Ionized Ca is usually in the 1.1-1.3 range. Ionized calcium is the same as "free calcium." Do NOT record total Ca here.
Calcium (ionized): low	Mutually exclusive with Calcium total: low. Lowest ionized calcium (Ca) during TIME PERIOD. Ionized Ca is usually in the 1.1-1.3 range. Ionized calcium is the same as "free calcium." Do NOT record total Ca here.
Glucose: high	Highest serum glucose (GLU, blood sugar, BS) or dextrostix (DS, D-STIX, DEXTRO, CHEMSTRIP, ONE-TOUCH) during the TIME PERIOD. If reagent strips are quoted as a range take the midpoint of the range. Normal range is 2.3-7.0. <i>serum glucose and dextrostix should be weighted equally in choosing high/low values.</i>
Glucose: low	Lowest serum glucose (GLU, blood sugar, BS) or dextrostix (DS, D-STIX, DEXTRO, CHEMSTRIP, ONE-TOUCH) during the TIME PERIOD. If reagent strips are quoted as a range take the midpoint of the range. Normal range is 2.3-7.0. <i>serum glucose and dextrostix should be weighted equally in choosing high/low values.</i>
Indirect bilirubin: high	Highest indirect bilirubin in the TIME PERIOD. This may be labelled as "INDIRECT" or "UNCONJUGATED" or as "I" and "U" respectively. These are equivalent, but are differentiated from direct/conjugated (recorded below). The sum of indirect and direct equals total bilirubin. If ONLY total bilirubin is available, record total value here and leave direct bilirubin as missing value. The highest direct and highest indirect do NOT need to come from the same specimen.
Direct bilirubin: high	Highest direct bilirubin in the TIME PERIOD. This may be labelled as "DIRECT" or "CONJUGATED" or as "D" and "C" respectively. These are equivalent, but are differentiated from indirect/unconjugated (recorded above). The sum of indirect and direct equals total bilirubin. If ONLY total bilirubin is available, leave direct bilirubin as missing value and score total value as indirect bilirubin. The highest direct and highest indirect do NOT need to come from the same

	specimen.
Hematocrit: high	Highest percentage hematocrit (Hct, Ht) drawn during the TIME PERIOD. If sent, but results not available, enter -9. if Hemoglobin (Hb, Hgb) available but no Hct, divide Hb by 10 and then multiply by 3 and enter this value. Values should be 30-55% (normal Hb is 80-240 g/L).
Hematocrit: low	Lowest percentage hematocrit (Hct, Ht) drawn during the TIME PERIOD. If sent, but results not available, enter -9. if Hemoglobin (Hb, Hgb) available but no Hct, divide Hb by 10 and then multiply by 3 and enter this value. Values should be 30-55% (normal Hb is 80-240 g/L).
BUN: high	Highest blood urea nitrogen (BUN) during the TIME PERIOD.
Creatinine: high	Highest creatinine (CR) during the TIME PERIOD.
Serum pH: low	Lowest pH during the TIME PERIOD. This may be obtained by ABG, CBG or VBG.
PCO2: high	Highest PCO2 recorded during the TIME PERIOD. Arterial, venous and capillary CO2 are acceptable, but non-invasive CO2 (transcutaneous, end-tidal) are not. Do not confuse with bicarbonate (HCO2), occasionally called "Total CO2."
Appropriate FiO2: high	Highest appropriate FiO2 recorded during the TIME PERIOD. <i>Appropriate FiO2 is defined at the end of the demographic section.</i> Note, the SNAP scoring period (since admission) is different than CRIB variables (12 hours of life).
Appropriate FiO2: low	Lowest appropriate FiO2 recorded during the TIME PERIOD. <i>Appropriate FiO2 is defined at the end of the demographic section.</i> Note, the SNAP scoring period (since admission) is different than CRIB variables (12 hours of life).
Blood gas with lowest pO2	<p>Select the ARTERIAL blood gas with the lowest pO2. If there are several blood gases at the same lowest pO2, record the one occurring first. Record the FiO2 (21%-100%) and MAWP (in cm-water) at the time blood was drawn, and the pH, PO2 and PCO2 from this blood gas.</p> <p>FiO2 should be recorded as the missing value if the baby was on blow-by oxygen at the time of the draw or if the FiO2 is not available. If the baby was on room air, record FiO2 as 21%.</p> <p>If on CPAP only, you may use CPAP value as MAWP if there is no MAWP listed. No distinction is made between nasal and endotracheal CPAP. If MAWP recordings do not correspond with blood gas times, assume constant MAWP between recordings. MAWP should be recorded as zero (0) if the baby was not on pressure respiratory support at the time</p>

	<p>of the draw (room air, blow-by oxygen, nasal cannula or hood). MAWP should be recorded as the missing value if the baby was on pressure support (including hand bagging) but MAWP is not available.</p>
Blood gas with highest MAWP	<p>Select the ARTERIAL blood gas with the highest mean airway pressure (MAWP). If this is the same gas recorded above in the lowest PO₂ row, select the gas with the next highest MAWP instead. If there are several blood gases as the same highest MAWP, record the one with the next lowest PO₂. If there are several gases with the same highest MAWP and the same lowest PO₂, record the gas occurring first. If no MAWPs are available, leave this row blank and enter -9 for the MAWP in the 1st and 3rd rows. If MAWP is "0" for the entire scoring period because the baby was never on respiratory support, leave this row blank and score "0" for MAWP in the 1st and 3rd rows.</p> <p>Record the FiO₂ (21%-100%) and MAWP (in cm-water) at the time blood was drawn, and the pH, PO₂ and PCO₂ from this blood gas.</p> <p>FiO₂ should be recorded as the missing value if the baby was on blow-by oxygen at the time of the draw or if the FiO₂ is not available. If the baby was on room air, record FiO₂ as 21%.</p> <p>If on CPAP only, you may use CPAP value as MAWP if there is no MAWP listed. No distinction is made between nasal and endotracheal CPAP. If MAWP recordings do not correspond with blood gas times, assume constant MAWP between recordings.</p>
Blood gas with highest FiO ₂	<p>Select the ARTERIAL blood gas with the highest FiO₂. If this is the same gas recorded above in the lowest PO₂ row OR in the highest MAWP row, select the gas with the next highest FiO₂ instead. If there are several blood gases with the same highest FiO₂, select the one with the lowest PO₂. If there are several gases with the same highest FiO₂ AND the same lowest PO₂, select the gas occurring first. Record the FiO₂ (21%-100%) and MAWP (in cm-water) at the time blood was drawn, and the pH, PO₂ and PCO₂ from this blood gas.</p> <p>FiO₂ should be recorded as the missing value if the baby was on blow-by oxygen at the time of the draw or if the FiO₂ is not available. If the baby was on room air, record FiO₂ as 21%.</p> <p>If on CPAP only, you may use CPAP value as MAWP if there is no MAWP listed. No distinction is made between</p>

	nasal and endotracheal CPAP. If MAWP recordings to not correspond with blood gas times, assume constant MAWP between recordings. MAWP should be recorded as zero (0) if the baby was not on pressure respiratory support at the time of the draw (room air, blow-by oxygen, nasal cannula or hood). MAWP should be recorded as the missing value if the baby was on pressure support (including hand bagging) but MAWP is not available.
CBC: white blood count	White Blood Count (WBC) of first, second or third complete blood count (CBC) during the TIME PERIOD. This is routinely denoted in THOUSANDS, so 10,200 is usually recorded as 10.2. record here in THOUSANDS (10.2). do not use "uncorrected" values. <i>If white blood count was not done in this CBC, record the missing value.</i>
CBC: poly count	Percent of polymorphonuclear leukocytes (PMNs, polys, P, neuts) in the first, second or third complete blood count (CBC) during the TIME PERIOD. <i>If the differential count was not done in this CBC, record the missing value.</i>
CBC: band count	Percent of Bands in the first, second or third complete blood count (CBC) during the TIME PERIOD. Bands includes the sum of ALL juvenile forms including BANDS STABS, JUVENILES as well as BLASTS, PROMYELOCYTES, METAMYELOCYTES and MYELOCYTES. These latter forms will be found only infrequently. <i>If the differential count was not done in this CBC, record the missing value.</i>
CBC: platelet count	Platelet count of the first, second or third complete blood count (CBC) during the TIME PERIOD. Platelets are usually denominated in THOUSANDS (e.g. 350, or 350K or 350,000 are equivalent). Clinically significant levels are those below 100. spuriously low values can occur in clotted specimens. If unusually low, verify in clinical notes that specimen not known to be clotted. If clotted, disregard value. If there is ONLY an unconfirmed value available, record this, otherwise use confirmed values only. Record here in THOUSANDS (350). <i>If platelet count not done in this CBC, record the missing value.</i>
Seizures	If only one seizure was confirmed, score as "single." If more than one seizure was confirmed, score as "multiple." Otherwise, check "none." Confirmed is defined as witnessed by 2 or more clinicians or diagnosed by EEG. Use of antiepileptics (phenobarbital) is not ALONE evidence of the diagnosis, but if antiepileptics are ordered by one clinician, and seizure is observed by a DIFFERENT clinician, assume the seizure is confirmed.

Apnea	<p>Apnea is defined as a 15 second or longer pause in respiration OR shorter pauses accompanied by bradycardia or cyanosis or recorded as apnea on the flow sheet.</p> <p>Score as "responsive" if:</p> <ul style="list-style-type: none"> - Apnea is recorded, but medically managed with self-stimulation, stimulation, theophylline, aminophylline, caffeine, etc. <i>Administration of theophylline WITHOUT apneic spells occurring during the scoring period does NOT count as apnea.</i> - Apnea is recorded, but medically managed with blow-by or facial CPAP - The baby has an apneic spell WHILE on CPAP or mechanical ventilation. <p>Score as "unresponsive" only when:</p> <ul style="list-style-type: none"> - It is clearly stated that the patient was STARTED on mechanical ventilation or CPAP in the delivery room IMMEDIATELY before transfer into the NICU or in the NICU FOR apnea - It is clearly stated that the patient was transferred from CPAP to mechanical ventilation in the delivery room IMMEDIATELY before transfer into the NICU or in the NICU FOR apnea. <p>Do NOT score as unresponsive unless there is clear evidence that the patient was put on the support DURING the scoring period FOR apnea.</p> <p>Score as "none" if:</p> <ul style="list-style-type: none"> - There is no recorded apnea - The patient is on skeletal muscle relaxants (pancuronium, Pavulon, succinyl choline, sux).
Stool hematest	<p>Test for occult blood in the stool (HEME+, H+, GUAIAC+). If blood is grossly evident (BLOODY, BRB=bright red blood), this should also be scored as positive. <i>This should not be confused with HEME+ results for urine.</i> If stool is not tested, described as normal or if there is no stool, score as negative.</p>
Urine output in CCs	<p>Total CCs of urine output during the TIME PERIOD. Do NOT divide by birth weight. If notes indicate that 20% or more of the total output for the TIME PERIOD was lost/unmeasured (recorded as mixed with stool, "VOID" or overflow) then record the missing value. To calculate whether 20% was lost, if all urine output values list specific numbers, determine whether the uncertain values (CCs recorded as mixed with stool or overflow) make up 20% of the total CCs. If some values are not recorded at all (recorded as "VOID")</p>

	then determine whether the uncertain values (unmeasured diapers) make up 20% of the total number of diapers.
Weight	Weight on the day for which SNAP score is abstracted. If more than one weight is listed during the TIME PERIOD, record the first weight occurring during the TIME PERIOD. If the baby was not weighed during the scoring period, record the missing value.

Appropriate FiO2/Respiratory Support

An FiO2 value is considered appropriate when one or more of the following apply:

- A. Arterial paO2 or transcutaneous TcpO2 is 50-80 torr (i.e. 6.7-10.7 kPa); OR
- B. FiO2 is more than 25% when arterial oxygen saturation by pulse oximeter (SaO2) is 88-95%; OR
- C. FiO2 is less than or equal to 25% when arterial oxygen saturation by pulse oximeter (SaO2) is more than 88%; OR
- D. FiO2 is 100% when arterial oxygen saturation by pulse oximeter (SaO2) is less than 96%.
- E. If there are no appropriate FiO2s while a baby is receiving supplemental oxygen (Or while a baby is not receiving supplemental oxygen) because saturations are continuously below 88%, score the high FiO2 as one more than the highest FiO2 listed and the low FiO2 as the missing value.

General FiO2 Scoring Information:

- If none of the FiO2s listed meet any of the definitions above, then score the high and low FiO2 as the missing value.
- If you ever find an appropriate FiO2 of 100, even if the rest of the FiO2s are unknown, score the high appropriate FiO2 as 100 (this is always true even if one of the conditions below states something contradictory).
- If you ever find an appropriate FiO2 of 21, even if the rest of the FiO2s are unknown, score the low appropriate FiO2 as 21 (this is always true even if one of the conditions below states something contradictory).
- If both pre and post TcPO2s are listed, use the post TcPO2.
- If both post-TcPO2 and saturations are listed during the same time block, do it once using each and record the one that gives the higher FiO2.
- Do not count FiO2s associated with transient oxygen saturations (desats). Only include FiO2s associated with sustained (continuous) oxygen saturations.

Moribund babies:

- Oxygen given for comfort care only should count towards CRIB and SNAP FiO2s, supplemental oxygen on NTISS, last day on oxygen on the Diagnoses and Procedures (D&P) screen and last day on oxygen at next hospital on the Post-Transfer screen.

- For moribund babies who receive no supplemental oxygen, you should score appropriate FiO₂s as the missing value.

CRIB:

- You may use other hospital records up to 12 hours of age in order to score CRIB FiO₂.
- If you are missing information for part or all of the scoring period because the baby was outborn and the birth hospital chart was not sent or was incomplete, score high and low appropriate FiO₂ as the missing value. If you do have other appropriate FiO₂ values, score using these values.
- If you do not have a complete scoring period because the baby died within the first 12 hours of life, score based on what information you have and note in the comments box that CRIB was scored on an incomplete scoring period.
- If oxygen was given before admission to the NICU (e.g. Oxygen given for an extended period of time in the delivery room), but you do not know how much, score as the missing value. "Extended period of time" is defined as greater than 3 minutes. This assumes that you have no other FiO₂ values available which are appropriate. If you have other values in the NICU (i.e., room air), ignore incomplete delivery room info.

Room air/no supplemental oxygen:

- When a baby is not receiving supplemental oxygen, consider room air appropriate (unless SATs are below 88%) and score high and low appropriate FiO₂ as 21%.
- For moribund babies who receive no supplemental oxygen, you should score appropriate FiO₂s as the missing values.
- MAWP should be listed as zero (0) if a blood gas is being drawn while a baby is in room air.

Blow-by oxygen:

- FiO₂s listed while a baby is on blow-by oxygen should NOT be used towards appropriate FiO₂ and should be ignored when trying to find high and low appropriate FiO₂.
- FiO₂ should be listed as the missing value in the ABG box if a blood gas is drawn while a baby is on blow-by oxygen.
- MAWP should be listed as zero (0) if a blood gas is drawn while a baby is on blow-by oxygen.
- Blow-by oxygen values do NOT count towards supplemental oxygen on NTISS, last day on oxygen on the D&P screen, oxygen at discharge on the Discharge screen or last day on oxygen at next hospital on the Post-Transfer screen.
- When a baby is both on nasal cannula and blow-by oxygen, score the FiO₂ as 5, not the missing value.

Bagging:

- FiO₂s listed while “bagging” should be used if appropriate.
- MAWP should be listed as the missing value if a blood gas is drawn while a baby is being “bagged.”
- “Bagging” should be used (if FiO₂ is greater than 21%) to score supplemental oxygen on NTISS, last day on oxygen on D&P, oxygen at discharge on the Discharge screen and lsat day on oxygen at next hospital on the Post-Transfer screen.

Facial CPAP:

- FiO₂s while on facial CPAP should be used if appropriate.
- MAWP should be recorded in the ABG box if a blood gas is drawn while a baby is on facial CPAP.
- Facial CPAP should NOT be scored as CPAP on NTISS and should NOT be used to score last day on CPAP/bent.
- Facial CPAP should be used (if FiO₂ is greater than 21%) to score supplemental oxygen on NTISS, last day on oxygen on D&P, oxygen at discharge on the Discharge screen and lsat day on oxygen at next hospital on the Post-Transfer screen.

Nasal cannula:

- If a baby is on nasal cannula for the entire CRIB or SNAP scoring period (regardless of what is listed in the FiO₂ column), you should enter 5 in the high FiO₂ box and 5 in the low FiO₂ box.
- If a baby is on nasal cannula for part of a CRIB or SNAP scoring period and some other form of respiratory support (Hood, CPAP, IMV or HiFi) for the other part of that period and there is at least one appropriate FiO₂ from the non-nasal cannula part of the period, you should score:
 - o The highest appropriate FiO₂ from the non-nasal cannula part of the period in the high FiO₂ box and 5 in the low FiO₂ box if the appropriate FiO₂ is greater than 21%.
 - o 555 in the high FiO₂ box and 21 in the low FiO₂ box if the only appropriate FiO₂ from the non-nasal cannula part of the scoring period is 21%.
- FiO₂s should be listed as 5 in the ABG box if a gas is drawn while a baby is on nasal cannula.
- MAWPs should be listed as zero (0) if a blood gas is being drawn while a baby is on nasal cannula.
- When a baby is both on nasal cannula and blow-by oxygen, score the FiO₂ as 5, not the missing value.
- When a baby is on nasal cannula at the beginning of day 28 or week 36, record the FiO₂ that is listed on the flow sheet in the “oxygen %” box. This is the only case where you would NOT record a 5 or 555 when a baby is on nasal cannula during your scoring period. If this information is missing, record the missing value.

- Nasal cannula oxygen for feeds only does count as supplemental oxygen (if FiO₂ is greater than 21%) EXCEPT for oxygen support at discharge on the Discharge screen.

Hood/CPAP/Mechanical ventilation/HiFi ventilation:

- FiO₂s associated with any of these forms of respiratory support should be used if appropriate.

Unknown/ambiguous values:

- If you know that supplemental oxygen was given, but you do not know the amount (e.g. Oxygen given for an extended period of time in the delivery room), score FIO₂ as the missing value.
- If you are missing all or part of a flow sheet, score appropriate FiO₂ as the missing value unless you can verify from other sources that the baby was always on room air (e.g. Profess notes state that the baby never received supplemental oxygen).
- If you are missing the day 28 or week 36 flow sheet and you can get accurate information from other sources, score based on these other sources. Otherwise, use the closest complete 24 hour scoring period (it is okay to adjust time of day).
- If you know the FiO₂, but not the SAT, ignore this value and select from the remaining FiO₂s.
- If you know the SAT, but not the FiO₂, select from the remaining FiO₂s.
- If all of the FiO₂s are either listed without SATs or are inappropriate, score both the high and low FiO₂ as the missing value.
- If the SAT is listed as a range (e.g. 90-100), score the FiO₂ if it is appropriate according to any of the values in the range.
- If the SAT is listed as an ambiguous range (e.g. 90s), and you cannot tell whether the FiO₂ is appropriate, do not score the FiO₂ as appropriate.
- If the FiO₂ is listed as a range (e.g. 24-28 with a SAT of 96), score any values in the range that are appropriate (24 and 25 in this example).
- If the FiO₂ is listed as a range, score the lowest appropriate FiO₂ value in that range for the FiO₂ in the ABG box (24 in the above example).
- FiO₂s that are listed as a decimal value should be rounded (up for 0.5 and down for 0.4) and then scored appropriately.
- If on CPAP and NP and no appropriate FiO₂s, score high as -9 and low as 5.

Appendix C

**Technical Report for Neonatal Intensive Care
Unit Database: Canadian Neonatal Network**

Technical report commenced: March 10, 2000

Working with Canadian Neonatal Network database with 20112 patient cases from January 8, 1996 to October 31, 1997.

D:\Ennett\NICU\BCv.3\BCsrc_0813_1999\snap1_3.sav

Based on Yanling Tong's Master's thesis, the following variables were extracted from the original NICU database (syntax code for SPSS):

```
save outfile='d:\Ennett\Nicu\BCv.3\BCsrc_0813_1999\snap1_2.sav'
/keep=hbloodp hheartr hrespr hsodium hpotass hhco3 hhema hpcO2
lbloodp lheartr ltemp lsodium lpotass lhco3 lhema lserum lpo2fio2
lpo2mawp lpo2po2 mawpfio2 mawpmawp mawppo2 fio2fio2 fio2mawp
fio2po2 cbc1wbc cbc2wbc cbc3wbc cbc1pol cbc2pol cbc3pol cbc1ban
cbc2ban cbc3ban cbc1plt cbc2plt cbc3plt seizure apnea guaiac urine bthwt
hcalctot hcalcion hgluc hidbili hdbili hbun hcreat lcalctot lcalcion lgluc
caselink.
```

The above variables can be used to calculate the appropriate input variables for the SNAP (Score for Neonatal Acute Physiology) score which contains 37 input variables. The following is a discussion on how the outliers were treated. There were 20112 cases (this review does not include output variables).

HRESPR: highest respiratory rate sustained for more than one minute

Frequency analysis identified 1051 cases (5.2 percent of the cases) which recorded 0 as the highest respiratory rate. At first glance, one would assume that there was an error, but taking a closer look at the SNAP Data Collection Guidelines, we see that the records are correct.

Directions according to SNAP Guidelines:

Count spontaneous respirations only. If on ventilator with no breaths above the vent, score as zero (0). *At some sites you may need to subtract the vent rate from the listed respiratory rate in order to find the number of spontaneous respirations.*

There were five cases of extreme outliers at the other end of the spectrum for this variable. The cases recorded highest respiratory rates of:

300 (caselink 3377)

420 (caselink 9718)

600 (caselink 5647)

6037 (caselink 15921)

7637 (caselink 9627)

The next highest value for this variable is 180, therefore, these cases were replaced with values of 180.

HHEMA: highest percentage hematocrit (Hct, Ht) drawn during TIME PERIOD

Directions according to SNAP Guidelines:

If sent, but results not available, enter -9. If Hemoglobin (Hb, Hgb) available but no Hct, divide Hb by 10 and then multiply by 3 and enter this value. Values should be 30-55 percent (Normal Hb is 80-240 g/L).

Here, there were no values recorded as -9. However, there were values above 100 (given that this is supposed to be a percentage, these are out of range). Because it is impossible to determine what the true values were here because it was supposed to be a percentage recorded, these values were replaced as missing.

380 (caselink 18284)

484 (caselink 6305)

526 (caselink 6354)

534 (caselink 6375)

2670 (caselink 19549)

HPCO2: highest PCO2 recorded during the time period

Directions according to SNAP Guidelines:

Arterial, venous and capillary CO2 are acceptable, but non-invasive CO2 (transcutaneous, end-tidal) are not. Do not confuse with bicarbonate (HCO3), occasionally called "total CO2".

Although there is no mention of recording missing values as '-9', there were 225 cases recorded as '-9'. Based on the assumption that '-9' was intended to indicate missing values, these cases were replaced as missing values.

LBLOODP: lowest mean arterial pressure (MAP), also called mean blood pressure (MBP)

This note is made here to bring attention to the fact that 3 cases had LBLOODP recorded as zero. There were no directions for this value in the SNAP Guidelines, so the values were left as is.

LHEARTR: lowest heart rate during the TIME PERIOD sustained for more than one minute continuously

Directions according to SNAP Guidelines:

Do not include transient heart rate values that reflect bradycardia associated with apnea/dsats. *Do NOT score the low heart rate as zero (0) for babies who die during a scoring period.*

Despite the directions to not record 0 for babies who die, there were 19 cases recorded as zero.

caselink 338 – TBLDISC_ = 7, patient died

caselink 2498 – TBLDISC_ = 8, moribund
 caselink 7500 – TBLDISC_ = 7, patient died
 caselink 8625 – TBLDISC_ = 7, patient died
 caselink 8952 – TBLDISC_ = 7, patient died
 caselink 10022 – TBLDISC_ = 7, patient died
 caselink 10436 – TBLDISC_ = 7, patient died
 caselink 10464 – TBLDISC_ = 7, patient died
 caselink 10476 – TBLDISC_ = 7, patient died
 caselink 10480 – TBLDISC_ = 7, patient died
 caselink 15437 – TBLDISC_ = 8, moribund
 caselink 15621 – TBLDISC_ = 8, moribund
 caselink 15904 – TBLDISC_ = 8, moribund
 caselink 15905 – TBLDISC_ = 8, moribund
 caselink 16510 – TBLDISC_ = 7, patient died
 caselink 17212 – TBLDISC_ = 8, moribund
 caselink 17213 – TBLDISC_ = 8, moribund
 caselink 17324 – TBLDISC_ = 7, patient died
 caselink 19790 – TBLDISC_ = 8, moribund
 Nothing was done to these cases.

LPOTASS: lowest serum potassium during the TIME PERIOD

Directions according to SNAP Guidelines:

Disregard hemolyzed potassium values. Normal values are in the 5-6.5 mmol/L range.

One case that was recorded as 0.1, and next lowest is 1.4. Replace 0.1 (caselink 9070) with minimum value 1.4.

LHEMA: lowest hematocrit (Hct, Ht) drawn during the time period

This note is made here to bring attention to the fact that 2 cases had LHEMA recorded as zero. There were no directions for this value in the SNAP Guidelines, so the values were left as is.

One case had LHEMA recorded as 210, when the next largest value was 80, therefore this case (caselink 18284) was replaced with the maximum value of 80.

LSERUM: lowest pH during the time period

There were two outlier values for this variable (values must be between 0 and 14.0 for the pH scale): 43 (caselink 17507) and 721 (caselink 3033). These cases were replaced with missing values.

Nasal Cannula:

Scores regarding oxygen values were scored '5' or '555' as appropriate to indicate if the infant was on nasal cannula.

LPO2FIO2 (555)

Possibly incorrect values: 0.21 (caselink 16475), 1 (caselink 6329, 18199), 7.31 (caselink 16445), 7.36 (caselink 16446), 20 (caselink 14003) – replace as missing values

Nasal cannula: 5 (113), 555 (caselink 16818, 16816, 10811, 10489, 10227) – replace as missing values

LPO2MAWP

Possibly incorrect values: 0 (1220)

MAWPFIO2

Nasal cannula: 5 (caselink 214, 2173, 7654, 9803, 14098, 14149, 14467, 14576, 14577, 15084, 16606, 19287) – replace as missing values

FIO2FIO2

Nasal cannula: 5 (29), 555 (caselink 2074) – replace as missing values

FIO2PO2

Nasal cannula: 5 (caselink 305) – replace as missing values

CBC1WBC: White Blood Count of complete blood count (CBC) during the time period

Directions according to SNAP Guidelines:

This is routinely denoted in thousands, so 10,200 is usually recorded as 10.2. record her in thousands. Do not use “uncorrected” values. If white blood count was not done in this CBC, record the missing value.

The values that appeared out of range were:

976 (caselink 864)
1013 (caselink 471)
1019 (caselink 480)
1152 (caselink 75)
1158 (caselink 192)
1225 (caselink 19130)
1442 (caselink 20287)
1493 (caselink 9948)
1959 (caselink 385)
1973 (caselink 20469)

These values were replaced with the next largest value of 335.

CBC1POL: percent of polymorphonuclear leukocytes (PMNs, polys, P, neut) in the CBC during the time period

There was a value recorded as -953 (caselink 15036). This case was replaced as a missing value.

Also, since this is supposed to be a percentage, the values over 100 were replaced as missing values.

177.33 (caselink 6596)
 200 (caselink 7821)
 220.78 (caselink 6599)
 300 (caselink 7820)
 400 (caselink 7822)
 606.99 (caselink 7479)
 2803.28 (caselink 19243)
 4400 (caselink 7859)

CBC3POL: percent of polymorphonuclear leukocytes (PMNs, polys, P, neutrs) in the CBC during the time period

A case was recorded as -99 (caselink 11161), therefore, it was replaced as a missing value.

CBC1BAN: percent Bands in the CBC during the time period

A case was recorded as -8 (caselink 14830), therefore, it was replaced as a missing value.

Also, since this is supposed to be a percentage, the values over 100 were replaced as missing values.

182.93 (caselink 6321)
 208.33 (caselink 6356)
 236.53 (caselink 19243)
 1400 (caselink 7916)

CBC1PLT: platelet count of CBC in the time period

Directions according to SNAP Guidelines:

Platelets are usually denominated in thousands (i.e., 350, 350k, or 350,000). Clinically significant levels are those below 100. Spuriously low values can occur in clotted specimens. If unusually low, verify in clinical notes that specimen not known to be clotted. If clotted, disregard value. If there is only an unconfirmed value available, record this, otherwise use confirmed values only. Record here in thousands. If platelet count not done in this CBC, record the missing value.

Extreme outliers here were replaced with the next highest value (4314).

9999 (caselink 16873)
 10055 (caselink 14202)

CBC2PLT: platelet count of CBC in the time period

Outlier was replaced with the next highest value (672).

6449 (caselink 9468)

CBC3PLT: platelet count of CBC in the time period

Outlier was replaced with minimum value (5).

0.29 (caselink 14698)

cc: total CCs of urine output during the time period

Directions according to SNAP Guidelines:

Do NOT divide by birth weight. If notes indicate that 20 percent or more of the total output for the time period was lost/unmeasured (recorded as mixed with stool, “void”, or overflow) then record the missing value. To calculate whether 20 percent was lost, if all urine output values list specific numbers, determine whether the uncertain values (CCs recorded as mixed with stool or overflow) make up 20 percent of the total CCs. If some values are not recorded at all (recorded as “void”), then determine whether the uncertain values (unmeasured diapers) make up 20 percent of the total number of diapers.

Given the information from the SNAP Guidelines, we can see that the data collectors were instructed to enter the missing value, however, there are many cases with the values -9 (10551 or 52.5 percent of the database), and one case with -93300 and two more with -8. These cases will be replaced with the missing value.

-93300 (caselink 7678)

-8 (caselinks 4309, 14947)

HCALCTOT: highest total calcium (Ca) during time period (mutually exclusive with Ca ionized)

Extreme outliers were replaced with the next highest value (4.2).

10 (caselink 8058)

10.5 (caselink 15112)

12.7 (caselink 17626)

LCALCTOT: lowest total calcium (Ca) during time period (mutually exclusive with Ca ionized)

Extreme outlier was replaced with the next highest value (4.2).

10.5 (caselink 15112)

HCALCION: highest ionized calcium (Ca) during time period (mutually exclusive with Ca total)

Leave as is.

LCALCION: lowest ionized calcium (Ca) during time period (mutually exclusive with Ca total)

Leave as is.

DOND

Dates listed as Jan 01, 1911 should be recoded as missing values.

Caselink 67, 120, 276, 292, 294, 298, 317, 323, 324, 350, 408, 418, 658, 1368, 4084, 4176, 4206, 4271, 6144, 6270, 6556, 6594, 6595, 6618, 6631, 6645, 6650, 6720, 6809, 6814, 6867, 7033, 9797, 9826, 9839, 9848, 9928, 10004, 10249, 10329, 10350, 10384, 10385, 10526, 10604, 10866, 11562, 11674, 11773, 11947, 12037, 12165, 12170, 12466, 12717, 12750, 12822, 13050, 13058, 13066, 13068, 13072, 15248, 15249, 15250, 15508, 15574, 15814, 15815, 15822, 15865, 15880, 15881, 15903, 15908, 15911, 15942, 15980, 16048, 16075, 16076, 16105, 16134, 16412, 16465, 16529, 16596, 16619, 16672, 16675, 16708, 16919, 16939, 16976, 17112, 17144, 17164, 17216, 17231, 17252, 17253, 17282, 17375, 17376, 17426, 18703, 18975, 19091, 19239, 19268, 19322, 19324, 19580, 19593, 19743, 19856, 19960, 20019, 20055, 20108, 20232, 20353, 20459.

OI: oxygenation index (calculated value)

One case recorded as 1100 (caselink 20166 – replaced with missing value).

HBUN: highest blood urea nitrogen (BUN) during the TIME PERIOD.

Blood urea nitrogen (BUN) is a metabolic by product (in the liver) from the breakdown of blood, muscle and protein. Blood urea nitrogen can be measured from a simple venipuncture specimen. Abnormal elevation in the blood urea nitrogen can indicate renal disease, dehydration, congestive heart failure, gastrointestinal bleeding, starvation, shock or urinary tract obstruction (by tumour or prostate gland). Low BUN level can indicate liver disease, malnutrition or a low protein diet. Normal BUN levels should be between 7 and 20 mg/dl (milligrams per decilitre).

<http://www.graylab.ac.uk/ogi-bin/omd?blood+urea+nitrogen>

Given this information, replace 136 (caselink 15036), 203 (caselink 14090), 367 (caselink 139) with next highest value 84.

FILE SAVED AS: *SNAP with missing values & fixed outliers day1.sav*

Appendix D

Derivation and Statistics of SNAP Variables

Table D-1. SPSS variable name and transformation formulae for SNAP variables

Parameter	Units	Variable name and transformation (if necessary)	Percent missing
Highest mean blood pressure	mmHg	Hbloodp	12.8
Lowest mean blood pressure	mmHg	Lbloodp	16.1
Highest heart rate	Beats/min	Hheartr	0.2
Lowest heart rate	Beats/min	Lheartr	0.3
Highest respiratory rate	Breaths/min	Hrespr	0.4
Lowest temperature	°F	Ltempf=ltemp*9/5+32	0.4
Lowest pO ₂	%	Lpo2po2	63.3
Lowest pO ₂ /FiO ₂ ratio	%/%	Po2fio2r=MIN(lpo2po2/lpo2fio2, mawppo2/mawpfio2, fio2po2/fio2fio2)	63.7
Highest pCO ₂	%	Hpco2	40.4
Highest oxygenation index	%/%	Oi=MAX(lpo2mawp/lpo2po2*lpo2fio2, mawpmawp/mawppo2*mawpfio2, fio2mawp/fio2po2*fio2fio2)/100	72.8
Highest hematocrit	%	Hhema	18.8
Lowest hematocrit	%	Lhema	33.9
Lowest white blood count	10 ³ cells	Lwbc=MIN(cbc1wbc, cbc2wbc, cbc3wbc)	17.6
Highest immature/total neutrophil ratio	10 ³ cells	Hitnrat=MAX(cbc1ban/(cbc1ban+cbc1pol), cbc2ban/(cbc2ban+cbc2pol), cbc3ban/(cbc3ban+cbc3pol))	
Lowest absolute neutrophil count	10 ³ cells	Lanc=MIN(cbc1wbc*(cbc1pol+cbc1ban), cbc2wbc*(cbc2pol+cbc2ban), cbc3wbc*(cbc3pol+cbc3ban))	
Lowest platelet count	10 ³ cells	Lplt=MIN(cbc1plt, cbc2plt, cbc3plt)	28.7
Highest blood urea nitrogen		Hbun	87.3
Highest creatinine	mg/dl	Hcreat	84.7
Lowest urine output	cc/kg/24 hr	Lurine=urine/bthwt/24*1000	
Highest indirect bilirubin - bili for BW>2kg - bili/kg for BW<2kg		Hidbili	77.3
Highest direct bilirubin		Hdbili	82.5
Highest sodium	mmol/L	Hsodium	59.3
Lowest sodium	mmol/L	Jsodium	66.1
Highest potassium	mmol/L	Hpotass	62.4
Lowest potassium	mmol/L	Lpotass	68.1
Highest total calcium		Hcalctot	74.7
Lowest total calcium		Lcalctot	80.1
Highest ionized calcium		Hcalcion	89.6
Lowest ionized calcium		Lcalcion	90.8
Highest glucose		Hgluc	8.9
Lowest glucose		lgluc	14.9
Highest serum bicarbonate		Hhco3	85.6
Lowest serum bicarbonate		Lhco3	86.0
Lowest serum pH		Lserum	40.0
Presence of seizures		Seizure	0.0
Presence of apnea		Apnea	0.0
Stool guaiac		Guaiaac	0.0

Appendix E

Neural Network Parameter Settings

Table E-1. Default network settings for
Step 1 experiments to extract ANN weights
for use as CBR match weights

Parameter	9 inputs and 9 inputs (modified order)
No. of layers	2
Hidden nodes	0
Weight-elimination	yes
RNG seed	1
Initial weights ^a	W1=rands()*1, B1=rands()*1 W2=rands()*0.1, B2=rands()*0.1
Error ratio	1.02
Lambda	0.0001 changing to 0.0005 at 200 epochs
Lambda decrement	0.999
Lambda increment	1.001
Learning rate	0.00001
Learning rate decrement	0.999
Learning rate increment	1.01
Momentum	0.8
Weight scale factor	0.1
Float value	0

^aWeights were initialized the same way for all experiments.

Table E-2. Optimal network settings for
Step 2 experiments for imputed data

Parameter	ANN weights	Clinician weights	Uniform weights	Mean	Random
No. of layers	2	2	2	2	2
Hidden nodes	0	0	0	0	0
Weight-elimination	Yes	Yes	Yes	Yes	Yes
RNG seed	1	1	1	1	1
Error ratio	1.02	1.02	1.0188	1.02	1.02
Lambda	0.005	0.005	0.005	0.0001	0.0001
Lambda decrement	0.999	0.999	0.999	0.999	0.999
Lambda increment	1.001	1.001	1.001	1.001	1.001
Learning rate	0.0005	0.0004	0.0005	0.0006	0.0006
Learning rate decrement	0.8118	0.9683	0.999	0.999	0.999
Learning rate increment	1.01	1.01	1.01	1.01	1.01
Momentum	0.99	0.99	0.99	0.99	0.99
Weight scale factor	0.01	0.01	0.01	0.01	0.01
Float value	0	0	0	0	0

Table E-3. Optimal network settings for
Step 3 experiments for horizontal
expansion

Parameter	Set A	Set B	Set C
No. of layers	2	2	2
Hidden nodes	0	0	0
Weight-elimination	Yes	Yes	Yes
RNG seed	1	1	1
Error ratio	1.05	1.04	1.01
Lambda	0.0005	0.00005	0.00001
Lambda decrement	0.99	0.999	0.999
Lambda increment	1.01	1.1	1.01
Learning rate	0.00007	0.0001	0.0007
Learning rate decrement	0.999	0.995	0.999
Learning rate increment	1.03	1.1	1.01
Momentum	0.2	0.6	0.94
Weight scale factor	0.6	0.15	0.3
Float value	0	0	0

Table E-4. Optimal network settings for
Step 3 experiments for variable reduction
following horizontal expansion

Parameter	30 inputs	22 inputs	16 inputs	14 inputs	13 inputs
No. of layers	2	2	2	2	2
Hidden nodes	0	0	0	0	0
Weight-elimination	Yes	Yes	Yes	Yes	Yes
RNG seed	1	1	1	1	1
Error ratio	1.01	1.005	1.005	1.005	1.007
Lambda	0.00001	0.00005	0.00005	0.0005	0.0005
Lambda decrement	0.999	0.999	0.999	0.999	0.999
Lambda increment	1.01	1.001	1.001	1.001	1.001
Learning rate	0.0007	0.0008	0.0008	0.0008	0.0008
Learning rate decrement	0.999	0.9032	0.9032	0.9032	0.9032
Learning rate increment	1.01	1.004	1.004	1.004	1.001
Momentum	0.94	0.99	0.99	0.99	0.99
Weight scale factor	0.3	0.01	0.0017	0.00017	0.00017
Float value	0	0	0	0	0

Table E-5. Optimal network settings for
Step 3 experiments for new Neonatal
Mortality Model with 13 inputs

Parameter	Hybrid model
No. of layers	3
Hidden nodes	2
Weight-elimination	Yes
RNG seed	1
Error ratio	1.02
Lambda	0.001
Lambda decrement	0.999
Lambda increment	1.001
Learning rate	0.00001
Learning rate decrement	0.999
Learning rate increment	1.0805
Momentum	0.4
Weight scale factor	0.01
Float value	0

Appendix F

Step 1 Statistical Analysis

Results of *t*-test for Significance of Difference Between True Values and Imputed Data

Table F-1. Statistics for *lbloodp* data

	Mean	N	Std. deviation	Std. error mean
True values	32.41	286	8.080	0.478
Hybrid: ANN weights	32.55	286	5.401	0.319
Hybrid: Clinician weights	32.71	286	5.568	0.329
Hybrid: Uniform weights	32.69	286	5.504	0.325
Mean (N=5102, complete only)	33.11	286	0.00000	0.00000
Mean (N=19427, entire database)	37.88	286	0.00000	0.00000
Random (N=5102, complete only)	33.7736	286	7.97339	0.47148
Random (N=19427, entire database)	38.6794	286	9.60591	0.56801

Table F-2. Paired samples correlations for *lbloodp* data

Pair		N	Correlation	Sig.
1	True values & Hybrid: ANN weights	286	0.645	0.000
2	True values & Hybrid: Clinician weights	286	0.648	0.000
3	True values & Hybrid: Uniform weights	286	0.653	0.000
4	True values & Mean (N=5102, complete only)	286	.	.
5	True values & Mean (N=19427, entire database)	286	.	.
6	True values & Random (N=5102, complete only)	286	0.027	0.651
7	True values & Random (N=19427, entire database)	286	0.027	0.651

Table F-3. Paired samples *t*-test results for *lbloodp* data (degrees of freedom=285)

Paired Differences							
Pair	Mean	Std. deviation	Std. error mean	95% CI ^a of the difference		t	Sig. (2- tailed)
				Lower	Upper		
1	-0.14	6.180	0.365	-0.86	0.58	-0.383	0.702
2	-0.30	6.167	0.365	-1.01	0.42	-0.815	0.416
3	-0.28	6.122	0.362	-0.99	0.43	-0.773	0.440
4	-0.6974	8.08015	0.47779	-1.6379	0.2430	-1.460	0.145
5	-5.4674	8.08015	0.47779	-6.4079	-4.5270	-11.443	0.000
6	-1.3610	11.19842	0.66218	-2.6644	-0.0576	-2.055	0.041
7	-6.2668	12.38526	0.73236	-7.7084	-4.8253	-8.557	0.000

^aCI = confidence interval

Table F-4. Statistics for *lserum* data

	Mean	N	Std. deviation	Std. error mean
True values	7.2982	689	0.11897	0.00453
Hybrid: ANN weights	7.3024	689	0.05168	0.00197
Hybrid: Clinician weights	7.3022	689	0.05054	0.00193
Hybrid: Uniform weights	7.3039	689	0.05146	0.00196
Mean (N=5102, complete only)	7.300	689	0.0000	0.0000
Mean (N=19427, entire database)	7.3200	689	0.00000	0.00000
Random (N=5102, complete only)	7.3047	689	0.11179	0.00426
Random (N=19427, entire database)	7.3183	689	0.10726	0.00409

Table F-5. Paired samples correlations for
lserum data

Pair		N	Correlation	Sig.
1	True values & Hybrid: ANN weights	689	0.267	0.000
2	True values & Hybrid: Clinician weights	689	0.258	0.000
3	True values & Hybrid: Uniform weights	689	0.294	0.000
4	True values & Mean (N=5102, complete only)	689	.	.
5	True values & Mean (N=19427, entire database)	689	.	.
6	True values & Random (N=5102, complete only)	689	0.004	0.917
7	True values & Random (N=19427, entire database)	689	0.004	0.917

Table F-6. Paired samples *t*-test results for
lserum data (degrees of freedom=688)

Paired Differences							
Pair	Mean	Std. deviation	Std. error mean	95% CI ^a of the difference		t	Sig. (2- tailed)
				Lower	Upper		
1	-0.0042	0.11635	0.00443	-0.0129	0.0045	-0.953	0.341
2	-0.0040	0.11663	0.00444	-0.0128	0.0047	-0.911	0.362
3	-0.0057	0.11492	0.00438	-0.0143	0.0029	-1.309	0.191
4	-0.0018	0.11897	0.00453	-0.0107	0.0071	-0.403	0.687
5	-0.0218	0.11897	0.00453	-0.0307	-0.0129	-4.816	0.000
6	-0.0065	0.16293	0.00621	-0.0187	0.0057	-1.051	0.294
7	-0.0201	0.15987	0.00609	-0.0321	-0.0082	-3.306	0.001

^aCI = confidence interval

Table F-7. Statistics for *lurine* data

	Mean	N	Std. deviation	Std. error mean
True values	0.8505	919	0.75138	0.02479
Hybrid: ANN weights	0.850	919	0.2909	0.0096
Hybrid: Clinician weights	0.8355	919	0.28719	0.00947
Hybrid: Uniform weights	0.8461	919	0.29502	0.00973
Mean (N=5102, complete only)	0.8400	919	0.00000	0.00000
Mean (N=19427, entire database)	0.9400	919	0.00000	0.00000
Random (N=5102, complete only)	1.1439	919	0.67006	0.02210
Random (N=19427, entire database)	1.2185	919	0.71374	0.02354

Table F-8. Paired samples correlations for *lurine* data

Pair		N	Correlation	Sig.
1	True values & Hybrid: ANN weights	919	0.261	0.000
2	True values & Hybrid: Clinician weights	919	0.199	0.000
3	True values & Hybrid: Uniform weights	919	0.202	0.000
4	True values & Mean (N=5102, complete only)	919	.	.
5	True values & Mean (N=19427, entire database)	919	.	.
6	True values & Random (N=5102, complete only)	919	0.005	0.887
7	True values & Random (N=19427, entire database)	919	0.005	0.887

Table F-9. Paired samples *t*-test results for *lurine* data (degrees of freedom=918)

Paired Differences							
Pair	Mean	Std. deviation	Std. error mean	95% CI ^a of the difference		t	Sig. (2- tailed)
				Lower	Upper		
1	0.0006	0.73159	0.02413	-0.0468	0.0479	0.023	0.981
2	0.0150	0.74899	0.02471	-0.0335	0.0635	0.607	0.544
3	0.0044	0.74979	0.02473	-0.0441	0.0529	0.178	0.859
4	0.0105	0.75138	0.02479	-0.0382	0.0591	0.422	0.673
5	-0.0895	0.75138	0.02479	-0.1382	-0.0409	-3.613	0.000
6	-0.2934	1.00440	0.03313	-0.3585	-0.2284	-8.857	0.000
7	-0.3680	1.03384	0.03410	-0.4349	-0.3011	-10.791	0.000

^aCI = confidence interval

Table F-10. Statistics for *po2fio2r* data

	Mean	N	Std. deviation	Std. error mean
True values	1.834	1066	1.2959	0.0397
Hybrid: ANN weights	1.8806	1066	0.52673	0.01613
Hybrid: Clinician weights	1.8794	1066	0.50351	0.01542
Hybrid: Uniform weights	1.8886	1066	0.54703	0.01675
Mean (N=5102, complete only)	1.8500	1066	0.00000	0.00000
Mean (N=19427, entire database)	1.900	1066	0.0000	0.0000
Random (N=5102, complete only)	2.1116	1066	1.21789	0.03730
Random (N=19427, entire database)	2.1619	1066	1.24700	0.03819

Table F-11. Paired samples correlations for
po2fio2r data

Pair		N	Correlation	Sig.
1	True values & Hybrid: ANN weights	1066	0.292	0.000
2	True values & Hybrid: Clinician weights	1066	0.255	0.000
3	True values & Hybrid: Uniform weights	1066	0.304	0.000
4	True values & Mean (N=5102, complete only)	1066	.	.
5	True values & Mean (N=19427, entire database)	1066	.	.
6	True values & Random (N=5102, complete only)	1066	0.011	0.712
7	True values & Random (N=19427, entire database)	1066	0.011	0.712

Table F-12. Paired samples *t*-test results for
po2fio2r data (degrees of freedom=1065)

Paired Differences							
Pair	Mean	Std. deviation	Std. error mean	95% CI ^a of the difference		t	Sig. (2- tailed)
				Lower	Upper		
1	-0.0464	1.24812	0.03823	-0.1215	0.0286	-1.215	0.225
2	-0.0453	1.26520	0.03875	-0.1213	0.0308	-1.169	0.243
3	-0.0544	1.24391	0.03810	-0.1292	0.0203	-1.429	0.153
4	-0.0159	1.29591	0.03969	-0.0938	0.0620	-0.400	0.689
5	-0.066	1.2959	0.0397	-0.144	0.012	-1.660	0.097
6	-0.2774	1.76832	0.05416	-0.3837	-0.1712	-5.123	0.000
7	-0.3277	1.78823	0.05477	-0.4352	-0.2203	-5.984	0.000

^aCI = confidence interval

Appendix G

List of Publications

Articles Published in Refereed Journals

1. Ennett CM, Frize M, Charette E. Improvement and automation of artificial neural networks to estimate medical outcomes. *Medical Engineering and Physics*. (Accepted with revisions Mar 2003)
2. Ennett CM, Frize M. Weight-elimination neural networks for mortality prediction in coronary artery surgery. *IEEE Transactions in Information Technology in Biomedicine* 2003 Jun;7(2):86-92.
3. Frize M, Ennett CM, Stevenson M, Trigg HCE. Clinical decision-support systems for intensive care units using artificial neural networks. *Medical Engineering and Physics* 2001 Apr;23(3):217-225.

Papers Published in Refereed Conference Proceedings

1. Ennett CM, Frize M. Validation of hybrid approach for imputing missing data, *Proc IEEE-EMBS* 2003 Sep 2003.
2. Ennett CM, Frize M, Scales N. Evaluation of the logarithmic-sensitivity index as a neural network stopping criterion for rare outcomes. *Proc IEEE-ITAB* 2003 Apr 2003.
3. Ennett CM, Frize M, Scales N. Logarithmic sensitivity index as a stopping criterion for neural networks. *Proc IEEE-EMBS/BMES* 2002.

4. Ennett CM. An overview of artificial neural networks applied to medicine. *Proc ICWES12 (International Conference of Women Engineers and Scientists)* 2002.
5. Ennett CM, Baniassad E, Miller-Young J, Sordi A, Le Clair B, Furlong J. Student branch discussion group for women in science and engineering organizations. *Proc ICWES12* 2002.
6. Ennett CM, Frize M, Walker CR. Influence of missing values on artificial neural network performance. *Medinfo* 2001;10(Pt 1):449-53.
7. Frize M, Ennett CM, Hebert P. Improving the efficiency and effectiveness of clinical decision-support systems using artificial neural networks. *Medinfo* 2001.
8. Walker CR, Ennett CM, Frize M. Use of an artificial neural network to estimate probability of mortality and duration of ventilation in neonatal intensive care unit patients. *Medinfo* 2001.
9. Frize M, Ennett CM, Charette E. Automated optimization of the performance of artificial neural networks to estimate medical outcomes. *Proc 3rd IEEE-ITAB Conference (Information Technology Applications in Biomedicine) and ITIS (International Telemedical Information Society)* 2000:168-73.
10. Ennett CM, Frize M. Selective sampling to overcome skewed a priori probabilities with neural networks. *Proc AMLA Symp* 2000;(20 Suppl):225-9.

11. Frize M, Ennett CM. Improving the potential clinical significance of decision-support systems using artificial neural networks. *Proc AMLA Symp* 2000:1011.
12. Ennett CM. Reengineering the Ottawa-Carleton WISE chapter. *Proc CCWEST (Canadian Coalition for Women in Engineering, Science and Technology)* 2000.
13. Ennett CM, Frize M, Shaw RE. Methodologies for predicting coronary surgery outcomes. *Proc IEEE-EMBS (Engineering in Medicine and Biology Society) and BMES (BioMedical Engineering Society)* 1999.
14. Adeney KM, Ennett CM, Frize M, Korenberg MJ. Prediction of patient outcomes in an intensive care unit. *Proc IJCNN (International Joint Conference on Neural Networks) Symp* 1999:7-12.
15. Ennett CM, Frize M. An investigation into the strengths and limitations of artificial neural networks: an application to an adult ICU patient database. *Proc AMLA Symp* 1998:998.
16. Tong Y, Frize M, Ennett CM. Discussion on the use of information theory to classify medical outcomes. *Proc AMLA Symp* 1998:1086.
17. Frize M, Wang L, Ennett CM, Nickerson B, Solven FG, Stevenson MH. New advances and validation of knowledge management tools for critical

care using classifier techniques. *Proc AMLA (American Medical Informatics Association) Symp* 1998:553-558.

Abstracts Published (of Papers Presented at Conferences)

1. Walker RC, Ennett CM, Frize M. Use of artificial neural networks to identify the contribution of different physiologic variables to neonatal mortality in newborn infants admitted to neonatal intensive care units. *Pediatr Res* 2003; 53:444A.
2. Walker RC, Ennett CM, Frize M. Potential utility of an artificial neural network in evidence-based ethical decision-making in the neonatal intensive care unit. *Pediatr Res* 2002; 51:374A. Also in *Paediatr Child Health* 2002; 7:57A.
3. Walker CR, Frize M, Ennett CM. Will artificial intelligence systems assist planning of resource allocation in the neonatal intensive care unit? *Paediatr Child Health* 2001; 6:28A.
4. Walker CR, Ennett CM, Frize M. Will artificial intelligence systems help counseling of parents in the neonatal intensive care unit? *Paediatr Child Health* 2001; 6:28A.

5. Walker RC, Ennett C, Frize M. Use of an artificial neural network to estimate probability of mortality in neonatal intensive care unit patients. *Pediatr Res* 2001; 49:310A.
6. Ennett CM, Frize M. Artificial neural network-based ICU data prediction. *Proc Ottawa Life Sciences Council Annual Conference* 1998.