TrendFinder: Automated Detection of Alarmable Trends

by

Christine L. Tsien

S.B. Computer Science and Engineering
Massachusetts Institute of Technology, 1991
S.M. Electrical Engineering and Computer Science
Massachusetts Institute of Technology, 1993

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2000

Author . . . .
Department of Electrical Engineering and Computer Science
April 28, 2000

Certified by . . . .
Peter Szolovits, Ph.D.
Professor of Computer Science and Engineering
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Departmental Committee on Graduate Students

TrendFinder: Automated Detection of Alarmable Trends

by

Christine L. Tsien

Submitted to the Department of Electrical Engineering and Computer Science
on April 28, 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

As information technology continues to permeate all areas of our society, vast amounts of data are increasingly being collected for their potential utility. This is especially true of data-rich environments, such as airplane cockpits or hospital operating rooms. These environments share in common the presence of multiple sensors whose aim is to monitor the state of affairs in that environment over time. Data from sensors, however, can only be as useful as we know how to interpret them and are able to do so in a timely manner. If we have a good understanding of the relationship between monitored sensor values and the status of the monitored process, we could create knowledge-based systems, for example, to help interpret the volumes of data that would not otherwise be possible for a human observer to do. Often, however, such relationships are not well understood. How could these data be useful then? Skillful use of machine learning is one answer.

In this thesis, we present TrendFinder, a paradigm for discovering the knowledge needed to detect events, or trends, in numerical time series. Specifically, we present a suite of data collection, preprocessing, and analysis techniques that enable effective use of existing supervised machine learning methods on time-series data. We demonstrate how these techniques can be applied to the development of 'intelligent' patient monitoring in the hospital intensive care unit (ICU), where currently as many as 86% of bedside monitor alarms are false. First, we describe application of the TrendFinder paradigm to artifact detection in a neonatal ICU. Second, we describe TrendFinder's techniques applied to detection of trends indicative of 'true alarm' situations in a medical ICU. Through illustration, we explore issues of data granularity and data compression, class labeling, multi-phase trend learning, multi-signal models, and principled time interval selection. We further introduce the idea of 'post-model threshold refinement' for adapting a machine-learned model developed for one population to use on a different population. Overall, we demonstrate the feasibility and advantage of applying TrendFinder techniques to ICU monitoring, an area that is extremely important, challenging, and in need of improvement.

Thesis Supervisor: Peter Szolovits, Ph.D.
Title: Professor of Computer Science and Engineering

To my parents

Acknowledgments

There are so many people who have contributed to my work that I hardly know where to begin. First and foremost, I would like to thank my thesis supervisor, Peter Szolovits, of the MIT Laboratory for Computer Science. I am grateful for his 'hands-off' approach to advising, his careful reading of my thesis, and his continued support of my pursuits during months when I was buried so deep in medical school, one had to wonder whether I would ever again emerge to pursue research. I would also very much like to thank my (what feels like long-time) mentor and friend, Isaac Kohane, of Harvard Medical School and Children's Hospital Informatics Program. Zak's belief in me kept me going through the good as well as the difficult times, for which I am eternally thankful. I am also grateful to Patrick Winston, of the MIT Artificial Intelligence Laboratory, whose worldly insights were instrumental in the shaping of both my written thesis and oral defense. Finally, I cannot continue without expressing my lasting gratitude to James Fackler, formerly of Children's Hospital and now of Johns Hopkins University School of Medicine. Without Jim's desire to 'close the loop' in automated mechanical ventilation (and his very personable manner), I would never have gotten involved in this work.

I would like to thank all of the MEDG members, past and present, at the MIT Laboratory for Computer Science, who have each helped in their individual ways. I would especially like to mention Hamish Fraser, for being always such a pleasure to interact with and so generously giving of his time to explain things; Jon Doyle, for his support of my research even when it was only tangential to his own; Ying Zhang, for her continued interest in doing research with me, making me feel like I was not working on this project entirely by myself; William Long, for his assistance in early decision tree work in myocardial infarction diagnosis; and Mojdeh Mohtashemi, for being especially supportive in those last weeks when I felt most time-pressured and anxious. Other MEDG members whom I would particularly like to thank include Heather Grove, Eric Jordan, Tze-Yun Leong, Yao Sun, Jennifer Wu, and Ruilin Zhao.

I am indebted to my ballroom dance partner, Seth Webster, who happened to also work at MIT Lincoln Laboratory for a group that had built LNKnet, one of the primary classification tools I ended up using extensively for my experiments. Seth could not only lead a beautiful waltz, but could also explain the obscure details of LNKnet code generation and scatter plot functionality. He even offered two days before the thesis due date to ftp fifty-some plot files to his computer, convert each one to an updated version of Postscript, and ftp them all back. Fortunately, I didn't need to take him up on that. I would additionally like to thank his colleagues, Richard Lippman and Linda Kukolich, both of Lincoln Laboratory, for assisting me in getting a working copy of LNKnet.

On a trip to Australia during my graduate school years, I had the pleasure of meeting Ross Quinlan. He has since been ever-helpful with c4.5 questions, for which I am very thankful.

I would next like to thank Martha Curley, of the Children's Hospital Multidisciplinary ICU in Boston, who continued to make annotated data collection, and therefore this work, possible after Jim Fackler had moved to Johns Hopkins. More recently, I am also grateful to Adrienne Randolph for her assistance in our continued research efforts at the MICU. I cannot begin to thank enough the MICU nursing staff for their several years of continued patience, cooperation, and assistance, which have made this work possible. I am also grateful to the students–Deborah Missal, Banny Wong, Scott Pauker, and Linda Ungsunan–who annotated the data for hours on end.

I am grateful to Neil McIntosh, of the Edinburgh Royal Infirmary in Scotland, for his assistance with the neonatal ICU data collection and annotation, and for his extraordinary patience with me during all the months I did clinical rotations rather than data analysis. I would also like to thank Peter Badger and the staff of the neonatal ICU at Simpson Memorial Maternity Pavilion at the Edinburgh Royal Infirmary for their invaluable assistance.

I would like to thank Bill McGarry, David Liimatainen, and Greg Murphy of Nellcor for their invaluable assistance such that I could both perform the study of the N-3000 and understand its technical specifications. I am also grateful to Jonathan Tien, formerly of SpaceLabs Medical, for his early assistance with data collection from the SpaceLabs monitors.

I would also like to thank Ryan Rifkin of the MIT Center for Biological and Computational Learning for performing the support vector machine experiments.

My acknowledgments would be incomplete without expressing my sincere gratitude to Paula Mickevich and Maria Sensale of the MIT LCS Reading Room. They continually amazed me with their ability to procure copies of obscure theses and references for me, tirelessly and efficiently.

As I finally plan to leave MIT 'Course VI,' after having started more than a decade ago as an undergraduate, I would like to thank Arthur Smith, my undergraduate department advisor; William Dally, my first graduate academic advisor; and Eric Grimson, my second graduate academic advisor (after William Dally had moved to Stanford). Without their collective support, and the support of Richard Mitchell and Patricia Cunningham at the Harvard-MIT Division of Health Sciences and Technology, I would never have been able to pursue my bizarrely unique educational path, non-traditional for even M.D.-Ph.D. programs. Furthermore, no Ph.D. thesis should be turned in to the Course VI department without thanking Marilyn Pierce for her years of keeping track of the things that need to be kept track of.

It is ironic that Stephen Peters, one of my closest friends during undergraduate years at MIT, who used to bail me out (just in time) when I didn't understand circuits and electronics, or diodes, or Clu programming, would now be the one to save me when it truly seemed that those archaic-style Postscript figures would just never fit within the allowable margins. To Steve, I am forever grateful.

Once upon a time I was more focused, or perhaps just more narrow-minded, and maybe could have persevered without constant support from my friends, but no longer. I am indebted to my friends and family who have continued to believe in me (year after year, after year, after year...) and who have kept me sane, entertained, and happy. I would particularly like to thank Christiana Bardon, David Barrett, Sola Cheng, Robert Fogelson, Trelawney Grenfell, Cheryl Henry, Mike Lemon, Kenneth Mandl, David Ouyang, Sara Schutzman, Annie Tan, Howard Yeon, and especially Robert Silvers.

This thesis is based upon work supported in part by a National Science Foundation Graduate Fellowship, a National Library of Medicine Medical Informatics Traineeship, a Howard Hughes Medical Institute Fellowship for Medical Students, an American Association for University Women Selected Professions Dissertation Fellowship, and DARPA contract F30602-99-1-0509.

About the Author

Christine L. Tsien was born in Minneapolis, Minnesota. After graduation in 1987 from Mounds View High School in Arden Hills, Minnesota, she attended the Massachusetts Institute of Technology where she majored in Computer Science and Engineering, with a humanities concentration in Russian language. After completing her Bachelor of Science degree in 1991, she worked on her master's thesis project at Hewlett Packard Laboratories in Palo Alto, California, earning her Master of Science degree in 1993. At that time, she joined the MIT Laboratory for Computer Science Clinical Decision Making Group ('MEDG'), where she was first introduced to the field of artificial intelligence in medicine.

In 1994, she began her medical studies at Harvard Medical School in the Division of Health Sciences and Technology. For the next six years, she divided her time amongst medical coursework, research in ICU monitoring, and clinical clerkships. During that time, she also published five peer-reviewed first-author articles, seven abstracts, and four essays, as well as competed in over 20 amateur ballroom dance competitions around the country and in England. She also supervised several MIT undergraduate research students, and served as a resident advisor for an MIT undergraduate living group.

After finishing her M.D. in June 2001, she plans to train in emergency medicine. Her long term goals are to both care for patients and continue research in the area of medical artificial intelligence. She is a member of the Massachusetts Medical Society, American Medical Association, American Medical Informatics Association, and American Association for Artificial Intelligence.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

As information technology continues to expand into all areas of our society, we need to understand how to take advantage of the data being made available. Vast amounts of data not only are being generated daily, but increasingly, are also being collected and stored for their potential utility. This is especially true of data-rich environments, such as airplane cockpits, hospital operating rooms, or manufacturing plants. These environments share in common the presence of multiple sensors whose aim is to monitor the state of affairs in that environment over time. This includes, for example, keeping watch on the proper functioning of an airborne plane, a patient undergoing surgery, or the machines in an assembly line. Data from sensors, however, can only be as helpful as we know how to use them. This requires knowing both how to interpret the data and how to do so in a timely manner.

When we have a good understanding of the relationship between monitored sensor values and the well-being of the monitored process, we might be able to create knowledge-based systems [199], for example, to help interpret the volumes of data which would not otherwise be possible for a human observer to do. In a small manufacturing plant, we might be able to apply rules within a knowledge-based system to determine whether a particular production machine has run out of parts, or has malfunctioned. Often, however, the underlying knowledge needed to write such rules in these data-rich environments is not well understood. What patterns of internet communication transfers are indicative of a 'Denial of Service' (DoS) attack on that internet site? In a space shuttle with 109 sensors [102], how does one begin to understand the interrelationships of those sensor readings? How do fluctuations in the stock market and predictions of this quarter's earnings affect the performance of a particular stock? When the relationships between events and monitored signals are not clear, as is the case in many data-intensive environments, how can data collected from those signals then be of value? Skillful use of machine learning is one answer.

Machine learning methods, such as neural networks and decision tree classifiers, are being used increasingly for knowledge discovery. Examples of their application are in loan advising, speech recognition, and robot vision [139]. In medicine as well, machine learning methods have been explored. A common target area for these methods is the classification of patients as having or not having a disease condition (e.g., myocardial infarction) based upon patient characteristics (e.g., age, gender, smoking history, and symptoms). An area of medicine that has not received as much attention for machine learning is data-intensive bedside monitoring. Patients in the operating room, intensive care unit (ICU), emergency room, labor and delivery department, coronary care unit, as well as other areas of the health care setting, are usually 'hooked up' to several lines, tubes, and probes that continuously monitor vital signs such as heart rate, blood pressure, and respiratory rate. While the classification of a patient as having a myocardial infarction or not is relatively easy to fit into the framework of machine learning, it is less clear how to formulate these bedside monitoring situations as machine learning questions.

In this thesis, we present TrendFinder, a paradigm for discovering the knowledge needed to detect events, or trends, in numerical time-series data. This includes an elaboration of data collection, preprocessing, and analysis techniques that, although currently overlooked, under-emphasized, or nonexistent, are essential to the success of knowledge discovery. Trends of interest might range from high-level events, such as a walking robot's loss of hydraulic power in one leg, to low-level events, such as sensor artifact. We then demonstrate how the TrendFinder paradigm can be applied for knowledge discovery in health care. Our target application is 'intelligent' patient monitoring in the hospital intensive care unit (ICU), where currently as many as 86% of bedside monitor alarms are false [206]. Specifically, we describe two applications of the TrendFinder paradigm: one, to detect low-level artifacts on monitored physiologic signals in the neonatal ICU; and another, to detect higher-level trends indicative of 'true alarm' situations in the medical ICU.

To summarize, the goals of this research are twofold: first, we would like to develop techniques that enable successful learning of interesting trends from numerical time-series data. Second, through our demonstration of the feasibility of these techniques in health care, we would like to gain a better understanding of the clinical problem of false alarms in the ICU, ways for their elimination, and construction of intelligent alarms.

1.2 Thesis Organization

The remainder of this thesis is organized into several major sections. We begin by describing the TrendFinder paradigm for event detection. Then, before describing two applications of TrendFinder to health care, we next set the stage by describing the existing state of patient monitoring in the hospital ICU. Following the descriptions of TrendFinder's application to detecting sensor artifacts and ICU true alarm events, we discuss in detail a couple alternative possibilities for improving patient monitoring. This is followed by a more thorough presentation of related work in the areas of time-series analysis, machine learning, monitoring and alarms, and intelligent alarm construction. Finally, we conclude with a summary of contributions.

Specifically, Chapter 2 provides a description of the TrendFinder paradigm for event discovery in time-series data. Each component of the process, including event identification, annotated data collection, annotated data preprocessing, model derivation, and performance evaluation, is discussed in detail.

Chapter 3 gives background information about hospital intensive care units, patient monitoring, and alarms. By illustrating the problems of existing monitors and their astoundingly high false alarm rates, our aim is to help the reader understand our motivation for exploring knowledge discovery techniques in the health care domain. We present the methodology and results of our prospective observational study that we performed in a pediatric hospital intensive care unit over the course of ten weeks.

Chapter 4 illustrates in detail the application of the TrendFinder paradigm to sensor artifact detection in the neonatal intensive care unit. Here, we also discuss the details of several preprocessing issues mentioned only at a high level in Chapter 2. Specifically, we explore, in the context of machine learning from time-series data, issues of data granularity and data compression, 'class labeling' of data as events or non-events, and learning of single-phase versus multi-phase trends of interest. We also compare the performance on artifact detection of several different types of classifiers, including logistic regression, decision trees, neural networks, radial basis function networks, and support vector machines.

Chapter 5 illustrates the application of the TrendFinder paradigm to another area of health care–

detection of true alarm events in the medical intensive care unit. Here, we discuss the details of several data collection and preprocessing issues that, again, were mentioned only cursorily in Chapter 2. Specifically, we explore issues of principled feature attribute selection, multi-signal versus single-signal learning, and non-traditional 'committee' classification systems. We further introduce the idea of 'post-model threshold refinement' for more effective use of a model developed for one situation in a new situation.

Chapter 6 provides a sampling of alternative approaches to decreasing false alarms in the intensive care unit. First, we turn to industry and study how well an available 'improved monitor' performs. Specifically, we compare the performance of two pulse oximeters, both from the same manufacturer, one of which is advertised to decrease false alarms due to motion artifact. The methods and results of that study are described. Secondly, we experiment with single-signal filter methods that do not involve using machine learning techniques and report on their performance at decreasing false alarms.

We then follow this detailed exploration of two alternative approaches with a more thorough survey of related work in Chapter 7. This includes work done in time-series analysis, patient monitoring and alarm algorithm development, and machine learning in various areas of health care.

Finally, Chapter 8 summarizes the contributions made in this thesis.

Chapter 2

TrendFinder Event Discovery Pipeline

Five fundamental parts comprise the pipeline for event discovery in time-series data. These are: identification of the event(s) of interest; appropriate collection of annotated data; appropriate preprocessing of annotated data; derivation of an event detection model; and performance evaluation. Emphasis is given to 'appropriate' data collection and preprocessing because these are steps that are currently under-emphasized in importance yet are crucial for successful model development. The five-step pipeline is depicted in Figure 2-1.

2.1 Event Identification

The first step in the event discovery pipeline is identification of the event or events of interest for knowledge discovery. An event in this context should be an entity that is thought to effect changes in available monitored time-series values; the exact nature of those changes is what we would

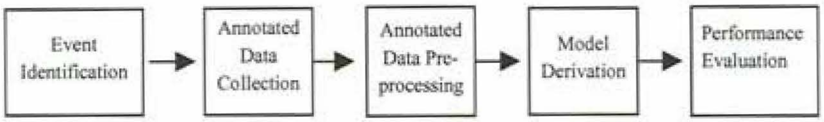


Figure 2-1: Components of the TrendFinder pipeline for event discovery in time-series data.

like to better understand. Examples of events include loss of cabin pressure on an airplane, voice commands given in the 'intelligent room' of the future [36], and raised interest rates announced by Alan Greenspan[1]. Examples of clinical events include disconnection of an electrocardiogram (ECG) lead, apnea (lack of breathing), false alarm due to patient motion, and blood pressure decrease warranting clinical attention.

Candidate events for knowledge discovery should either occur frequently, or, if not, occur in environments amenable to prolonged monitoring and observation such that adequate numbers of those events may eventually be observed. In some domains, such as research in intrusion detection [96, 120, 121], limited amounts of actual event data may necessarily need to be supplemented by simulated data. Candidate events should also be such that it is clear to an observer when they are or are not occurring. For example, if a patient is breathing, clearly the 'apnea' event is not occurring; if the patient has not been breathing for the past minute, on the other hand, clearly the 'apnea' event is occurring.

2.2 Annotated Data Collection

Once an event of interest has been identified, the next step is to collect a large amount of time-series data along with annotations of event occurrences and event 'non-occurrences'. Time-series data can usually be stored to computer disk either in a central data repository or via a laptop computer. The granularity of the collected data, that is, how frequently the measured signal value is stored, should usually be as close as possible to that available in the actual environment. We will show in Chapter 4, however, that sometimes data can be compressed without loss of effectiveness in model-building or in subsequent testing on data of the native time granularity.

Annotations, when at all feasible, must be made prospectively, at the time of event occurrence. Retrospective speculation in some cases, for example, to determine the exact timing of Alan Greenspan's announcement of an interest rate hike, may be acceptably accurate. In medicine, however, retrospective chart review to try to determine the time at which a patient's bedside monitor sounded a false alarm, for example, is grossly inadequate. Some researchers have suggested annotating ICU data post-collection based on the nurse and physician notes (R. Mark, personal communication); a study of pulse oximetry use in general medical-surgical nursing units, however, found that multiple episodes of hypoxemia and low saturation levels (which account for the majority

[1] United States Federal Reserve Chairman

of ICU false alarms) were not reflected in the medical record (nurse or physician notes or orders to change respiratory care) [22]. When annotations can only be made retrospectively, a preliminary study can first be performed, with plans to follow later with a prospectively annotated study.

Annotations furthermore need to be 'time-stamped' for accurate correlation with the data, which often are in a separate file or files. Ideally, these time-stamps have both a start and a stop time and are of the same time granularity as the data being collected. One way to meet these criteria is by using a custom-built program in which a human observer can easily record time periods of event occurrence and event non-occurrence. An observer, if not already knowledgeable about the area, can be trained to recognize which events to look for and to obtain verbal verification of events from knowledgeable staff. An alternative to a custom-designed annotation program is a custom-formatted data entry interface for a readily available commercial spreadsheet program, such as Access (Microsoft, Redmond, WA).

An alternative method for collecting data and event information in the hospital setting could be to use documented monitors [217]; these have been used in studies of home monitoring efficacy. Documented monitors can record, for example, transthoracic impedance, ECG heart rate, hemoglobin saturation, and pulse waveform, as well as store the dates and times of alarm events.

Data collection and annotation should proceed for as long as is feasible to capture multiple occurrences of the event of interest. Typically, machine learning programs are more robust when presented with larger numbers of samples of the event of interest [59]. Initial model development can also be tried periodically, with return to data collection if inadequate models (due to insufficient event samples) result.

2.3 Annotated Data Preprocessing

The next step in the event discovery pipeline is the preprocessing of annotated data. Preprocessing is much more important than generally recognized. This is the step that enables us to apply traditional machine learning methods to less traditional application areas such as medical monitoring, and other domains with monitored sensor values. The three major components of the preprocessing step are feature attribute derivation, class labeling, and data partitioning.

2.3.1 Feature Attribute Derivation

Feature attribute derivation refers to the selection and calculation of mathematical quantities that can describe the time-series data. These mathematical quantities, such as the moving mean or median, can be any features of time-series data that are thought to be potentially different for events versus non-events. The quantities are calculated over a specified time interval (for example, 10 seconds), for successively overlapping data. The same quantities can also be calculated over multiple time intervals (for example, 10 seconds and also one minute) and then used as two different attributes of that monitored signal. Time intervals for feature attribute derivation may be chosen to reflect a very general understanding of the problem if that is all that is known. For example, very short time intervals might be chosen for spurious false alarms. An analysis of time-interval selection, that is, how to go about choosing time intervals for feature attribute derivation in a more principled manner, is described in Chapter 5.

The derived values described above are calculated not only for just one signal type, but for all available data signals being collected. The various quantities calculated for each signal, for all monitored signals of interest, together comprise the set of feature attributes that describe a time period of monitoring. When there exists only one monitored value, or in cases in which one knows with certainty that only one signal is related to the event of interest, event detection models can be developed from feature attributes of a single signal. When multiple monitored signals exist and precise relationships between each signal and the event of interest are unclear, however, it is better to derive feature attributes from all available signals that may have some correlation to the event of interest. An example analysis of single-signal versus multi-signal event detection models is presented in Chapter 5.

Feature attributes can also consist of multiple parts, or phases, corresponding to characteristics of sequential blocks of a monitored signal. For example, a two-phase feature or pattern in the data may describe the slope of the first phase and the slope of the second phase of two contiguous regions. The first phase might correspond to a 10-second time interval, for example, while the second phase might correspond to the adjacent and following 20-second time interval. Individual phases of multi-phase features are subject to the same flexibility of choice in time-interval selection as previously described for single-phase feature attributes.

These multi-phase patterns can be described either by a single attribute that encapsulates the idea of the whole pattern, or by multiple attributes that together paint the picture of the whole pattern. Say for instance that we would like to learn temporal patterns, or trends, consisting of two phases. For simplicity, consider the case in which each phase is 30 seconds in length and can be described by only one of three slope characteristics: positive (rising), zero (level), or negative (falling). There would then exist nine unique two-phase patterns, as shown in Figure 2-2. We can choose, based on the classification system to be used or on other factors, whether to represent these two-phase characteristics by a single attribute (e.g., 'two-phase pattern number 1', 'two-phase pattern number 2', 'two-phase pattern number 3', ..., 'two-phase pattern number 9') or by multiple attributes (e.g., 'phase-one slope' and 'phase-two slope'). These ideas can readily be extended for learning more complex trends, consisting of three or more phases, which may be associated with event occurrences. Figure 2-3 illustrates a data stream with three contiguous phases, indicated by adjacent rectangles, to be used for feature attribute derivation. In the example, the slopes of the phases are falling, level, and rising for the first, second, and third phases shown, respectively. In Figure 2-4, the time intervals of the three phases have been changed. As a result, the slopes of the phases are rising, falling, and rising for the first, second, and third phases shown, respectively.

Except in cases in which the number of possible multi-phase patterns is small, representing these multi-phase patterns with several attributes that describe each individual phase is preferred. First, the latter method would be computationally more tractable. From a pattern learning standpoint, moreover, the latter method provides flexibility for learning peculiarities about individual phases of a temporal pattern. Also, pattern types, although feasible as inputs into a decision tree induction system, for example, are not amenable to neural network learning. Chapter 4 provides an example of learning multi-phase patterns.

2.3.2 Class Labeling

Each set of feature attributes–whether it be multi-signal or single-signal, multi-phase or single-phase, consisting of attributes derived over several time intervals or just one–can be thought of as a feature vector and is given a class label of 'event' or 'non-event' according to the recorded annotations. This is easiest to do when annotations and data share the same time granularity and when annotations have clear start and stop points. A method of 'windowing' to compensate for lack of either or both of these desired annotation features is described further in Chapters 5 and 6.

Class labeling can be varied on two separate axes: 'location' and 'strictness.' One axis, location, concerns the notion of 'front-labeling' versus 'end-labeling.' This is only an issue when more than one time interval is used for feature attribute derivation. For example, let's say we would like to calculate feature values over three data points, five data points, and ten data points. For any

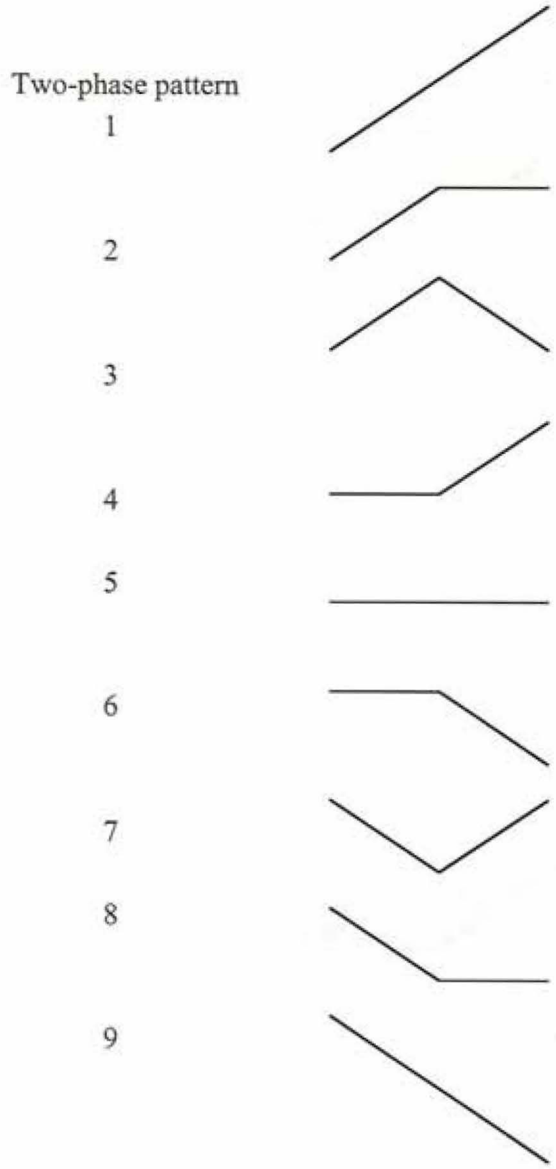


Figure 2-2: Nine possible two-phase patterns in which each phase is represented by one of three slopes (rising, level, falling).

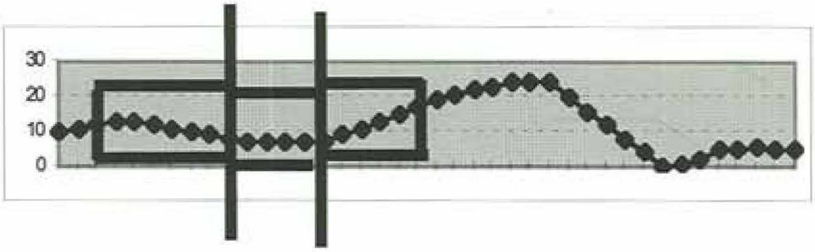


Figure 2-3: Multi-phase feature attributes: each phase is indicated by a rectangle. Vertical bars indicate transitions between two phases. Slopes are falling, level, and rising for the first, second, and third phases shown, respectively.

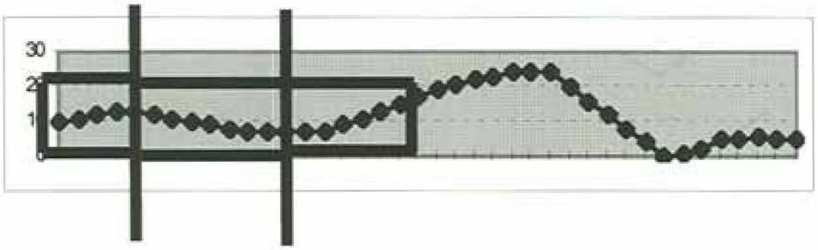


Figure 2-4: Multi-phase feature attributes: each phase is indicated by a rectangle. Vertical bars indicate transitions between two phases. Slopes are rising, falling, and rising for the first, second, and third phases shown, respectively.

particular interval of ten points, then, there are several options for which five of those ten points can be used for the five-point time interval calculations. Similarly, there are several possible three-point time intervals within that ten-point interval. In such cases, we might choose to use the annotations of the shortest time interval to determine the class label for the collection of all features derived from all of the chosen time intervals. Front-labeling then means that the time interval from which the feature vector's label is determined occurs at the 'front' of a stream of data, or at the earliest temporal region. Longer time intervals thus start with the same raw values as the entire shortest time interval, then additionally include however many more raw values are needed that follow the shortest time interval. Contrast this method with end-labeling, which places the shortest time interval at the 'end' of the longest time interval (at the latest position, temporally-speaking). These labeling methods are depicted in Figure 2-5. Front-labeling might be useful in applications such as learning the result of giving a patient a particular intravenous drug; the interest lies in the temporal patterns following drug delivery. End-labeling might be useful in applications such as trying to predict a stock's plummet in value; the interest lies in learning temporal patterns indicative of a crash that precede the crash (for example, to allow investors time to move their money). Chapter 4 presents a comparison of front-labeling and end-labeling in artifact detection.

The other axis along which class labeling can be varied is 'strictness' of class inclusion or exclusion. For example, consider data in which raw data values are each individually marked as 'event' or 'non-event.' Feature attributes derived from multiple raw values may span regions of different 'event' or 'non-event' markings. One way in which to translate these individual raw value event annotations to class labels is by taking the average value over the time interval under consideration. 'Event' average values would therefore range from 0 to 1, inclusive. The most strict class labeling technique would require that only feature vectors whose attributes are derived from data with event average equal to one be labeled 'event'; similarly, only feature vectors whose attributes are derived from data with event average equal to zero would be labeled 'non-event.' Time intervals spanning a transition from an event to a non-event or vice versa can thus electively be disregarded for model development experiments. Alternatively, these transition periods can themselves become the event of interest that we wish to learn how to detect.

One interpretation of the least strict labeling method would be to give any feature vectors whose attributes are derived from data with event average greater than zero the 'event' class label. This in some sense makes for a more challenging classification problem since monitored periods at the very edge of an event's start or finish are also included in the event class even though they may

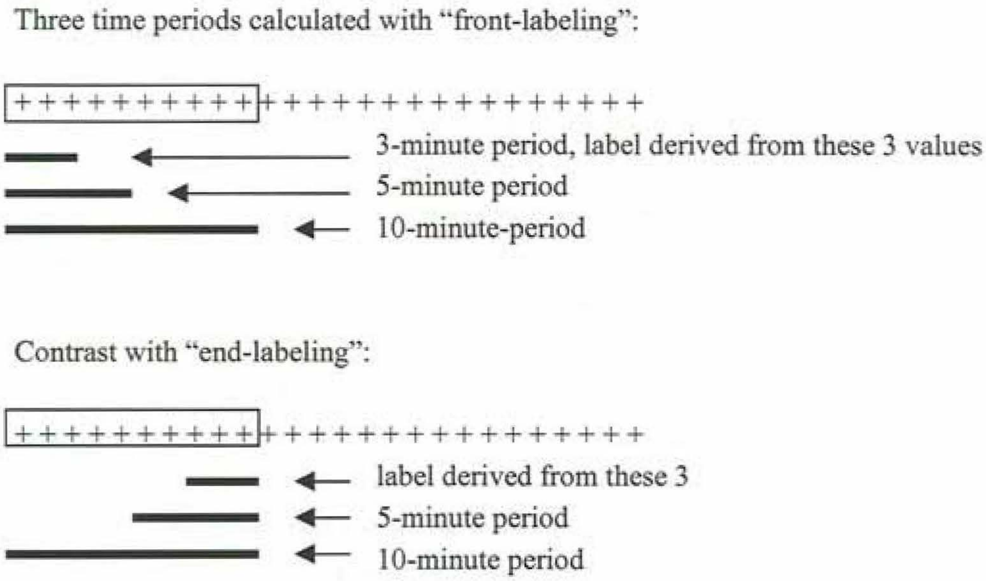

Three time periods calculated with “front-labeling”:
+++++++++++++++++++++++++++
3-minute period, label derived from these 3 values
5-minute period
10-minute-period
Contrast with “end-labeling”:
+++++++++++++++++++++++++++
label derived from these 3
5-minute period
10-minute period

Figure 2-5: Front-labeling versus end-labeling of temporal data feature vectors spanning more than one time interval.

not belong. Other thresholds for class inclusion or exclusion are also possible. Chapter 4 illustrates and compares four different methods (levels of strictness) of class labeling based upon the general methods described here.

Class labeling with multi-phase features presents the additional option of choosing from which phase or part of a phase to derive the feature vector's label.

2.3.3 Data Partitioning

All collected data are thus transformed into sets of class-labeled multi-signal feature attributes. Before machine learning methods can be applied to preprocessed data, we must randomly partition our available data into, ideally, three sets. These three sets consist of a training data set, an evaluation data set, and a test set. The training set is used for deriving candidate event detection models. The evaluation set is used to determine how well candidate models perform relative to each other; this enables experimentation with model development while avoiding the potential problem of fitting a model to the actual test data. Once a final model is selected, it is then run on the reserved test set to determine the model's performance. A training set consisting of approximately 70%-80% of the available data is often chosen, while the remaining data are further split to create the other two data sets.

2.4 Model Derivation

For the techniques described thus far, 'supervised' machine learning methods, such as neural networks or decision trees, can now easily be employed. These machine learning methods facilitate development of models, from labeled training data, which can then be used to classify unseen data as events or non-events.

Quite an array of classification methods exist. One way in which to think about these methods at a high level is by broadly categorizing them as linear or non-linear approaches (Tom Mitchell, personal communication). An example of a linear method is logistic regression (LR) in which there are no interaction terms. Probably the best-known non-linear method is neural networks.

There has been much debate about which classifiers work better than others. We believe that in this area, event discovery from time-series data, the choice of the classification system is of only minimal importance. What is more important is the proper collection, annotation, and preprocessing of the data to be used for learning. While there may be large differences between linear and non-linear classifiers, classifiers within the same category are likely to perform relatively similarly, with each having its own advantages and disadvantages.

A brief description of several classification systems is nonetheless presented; the purpose is to acquaint the reader with classifiers that are used in experiments described in Chapters 4 and 5.

2.4.1 Logistic Regression

Logistic regression (LR) is a classification method that uses a set of samples of known classification to derive coefficients for an equation that calculates the probability that a new case is of a certain class. The equation below shows the form of the logistic regression equation.

$$Probability = \frac{1}{1+e^{-(\beta_o+\sum_i \beta_i X_i)}}$$

where β_o is a constant term, β_i terms are the derived coefficients, and X_i terms are the values of the attributes used to determine the case's classification (0 or 1 for dichotomous attributes; integers, for example, for continuous attributes).

In general, the time-consuming and difficult aspect of building an LR model is deciding which attributes to include in the model and which to exclude. In many classification problems in which the relationship between attributes and class is not well understood, many attributes are available for model-building. Choosing a small but meaningful subset, however, can pose a daunting task.

The approach taken here has been to allow a decision tree induction system (described below) to select the optimum variables. This simplifies the job of building an LR model using the JMP statistical package (SAS Institute, Carey, NC).

2.4.2 Neural Networks

Neural networks are classification systems originally inspired by biological learning systems in which numerous neurons are interconnected, as in the brain. These artificial classifiers use the output values of internal calculation nodes to estimate the posterior class probabilities of the patterns seen at their input nodes. Like other supervised learning techniques, they are a method of modeling an input-output relationship based on example data for which the input-output relationships are already known.

The neural network model consists of 'neuron'-like processing units linked together in layers. The output of a single processing unit is a weighted sum of its inputs passed through a sigmoid function. These weights are successively modified, or trained, using a back propagation algorithm to perform a gradient descent which minimizes the error seen at the outputs.

The equation below gives the form of the sigmoid function used at each calculation node.

$$Output = \frac{1}{1+e^{-(\sum_{i=1}^{m} w_i x_i - \delta)}}$$

In the equation, m equals the number of neuron inputs; w_i equals the weighting factor for input i, x_i equals the ith input value, and δ equals the neuron offset.

Multi-layer perceptrons (MLP) are a special kind of neural network in which the neurons are arranged in layers and the outputs of one layer are used as the inputs to the next layer. Furthermore, every neuron output is fully interconnected to the inputs of the subsequent layer. In this study, we specifically mean multi-layer perceptrons when referring to experiments using neural networks.

In our studies, we use the neural networks available in the LNKnet classification system (Lincoln Laboratory, Lexington, MA) [113]. Options to experiment with when developing neural network models include use of data normalization; use of re-sampling of training data when class probabilities in the training data are not uniform; selection of the number of hidden nodes and hidden layers in the network; selection of the 'step size' by which to modify network weights during training; and selection of the number of 'epochs' through which the network should back-propagate errors and make weight changes accordingly.

2.4.3 Decision Trees

Decision trees, or classification trees as they are sometimes called, can be generated by machine learning algorithms used to classify new data using a tree structure derived from a sample of training data of known classification. Each 'datum' consists of several attributes (e.g., characteristics of chest pain), and one class label (e.g., myocardial infarction or non-myocardial infarction). The trees are built by looking for regularities in the data with which to separate the data by class.

The decision trees built in this project use either Quinlan's c4.5 program[166] or the LNKnet system [113], both written for Unix systems. The input to c4.5 is a file specifying the available attributes and their value type, as well as the possible classifications for each sample. Training data cases are provided in a separate file, and optional test items are provided in a third file. LNKnet uses analogous input files with slightly different format for the data.

In c4.5, several options are available for modifying tree-building behavior. The options experimented with include: 'm', the minimum number of cases needed in at least two outcomes of a tree node in order to include that node while creating the tree; and 'c', the 'confidence level' of the predicted error rate on each leaf and each subtree, used to find the upper limit on the probability of error at a leaf or subtree during pruning. Increasing m values and decreasing c values often result in smaller trees. LNKnet has similar options that allow the user to specify when to stop tree growth and how many nodes to use during testing (essentially, how much to prune).

When possible, expert knowledge can also assist in selection of a final tree model from those with the best numerical results on training data. In medicine, for example, clinical judgment can be used to determine whether proposed attributes are themselves consistent with the goal; whether attributes within a given branch make sense with respect to each other; and in cases where an attribute is repeated more than once in the same branch, whether there is clinical plausibility for this.

2.4.4 Radial Basis Function Networks

Radial basis function networks are similar to multi-layer perceptrons except that instead of using sigmoids for calculation at each node, they use local Gaussian functions. The LNKnet User's Guide [113] provides an excellent brief summary of these networks.

> Radial Basis Function classifiers calculate discriminant functions using local Gaussian functions instead of sigmoids of hidden node sums. They may perform better than MLP classifiers if input features are normalized so a Euclidean distance is meaningful and if class distributions exhibit radial symmetries. Network outputs are weighted sums of the outputs of Gaussian hidden nodes or basis functions. Hidden node outputs are normalized to sum to one. Weights are trained using least-squares matrix inversion to minimize the squared error of the output sums given the basis function outputs for the training patterns. These basis functions can include a constant bias node.

Some of the options available for training a radial basis function classifier are: selection of the number of 'clusters' corresponding to Gaussian hidden nodes, selection of whether the clusters should be created by class or without respect to class; selection of whether all network weights should be updated during each iteration of training; and selection of a clustering algorithm.

All experiments performed for these studies using radial basis function classifiers are done using the LNKnet system.

2.4.5 Support Vector Machines

A support vector machine (SVM) is a kind of classifier relatively new to the field. A support vector machine, developed by Vapnik and his team at AT&T Bell Laboratories [20, 27, 42, 71, 162], is a pattern classification algorithm that can be seen as a way to train polynomial, neural network, or radial basis function network classifiers. Training an SVM, that is, finding the decision surface that best separates samples by their classification, is equivalent to solving a linearly constrained quadratic programming problem. The number of variables in the equivalent quadratic programming problem is equal to the number of data points in the given training set.

Support vector machines are an approximate implementation of the 'structural risk minimization' induction principle, which means that they try to minimize an upper bound on the generalization error of a classifier rather than on the training error. To get an intuitive sense for support vector machines, Figure 2-6a shows in two dimensional space samples of two different classes separated by a decision surface line (hyperplane in higher dimensional space) with its associated 'margin' size. The margin is defined as the sum of the distance from the decision surface to the closest point of one class and the distance from the decision surface to the closest point of the other class. Figure 2-6b shows the same data with a different separating decision surface that has an associated margin of larger size. The larger margin size is expected to have better generalization capability in classifying new samples. The idea behind the support vector machine is that the location of the separating decision surface only depends on those samples closest to the decision surface; these are 'support' samples. In multi-dimensional space, each sample is represented by a vector quantity, and hence these points are called 'support vectors,' as shown in Figure 2-6c.

The example given in Figure 2-6 is of course overly simplistic, as it shows two linearly separable classes. For non-separable classes, the decision surface that maximizes the margin and minimizes the number of misclassification errors is desirable. This tradeoff is represented mathematically by a positive constant, C, that the user of a support vector machine must choose before training.

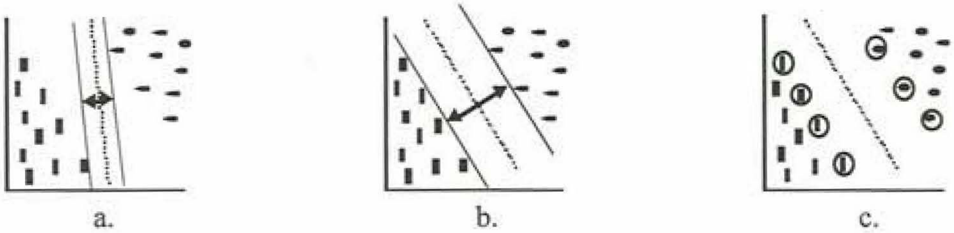


Figure 2-6: Understanding support vector machines: (a) The separating decision surface (dotted line in 2D) has a small margin. (b) The separating decision surface has a larger margin. (c) Placement of the decision surface depends only on those samples closest to the boundary between the classes; these samples are called 'support vectors' (shown circled).

The solution to finding the decision surface (linear classifier in this case) has been shown to be equivalent to the solution of an appropriately formulated quadratic programming problem. Each training data sample is associated with a term in the mathematical equation, but only those data points corresponding to the support vector points have non-zero coefficients. Thus, only the support vectors are relevant to the solution of the problem.

This approach is then further extended to allow for nonlinear decision surfaces by projecting the original set of variables of the linear classifier into a higher dimensional feature space, such that the classification problem in feature space is linear, but in the original input space is nonlinear. With well-chosen mathematical manipulations [153], the resulting quadratic programming problem that must be solved to determine a classifier's decision surface becomes almost exactly like that of the original linear classifier. The user of a support vector machine must choose the 'kernel function,' K, for the mathematical form of the SVM classifier. Vapnik showed that choosing particular kernel functions makes the SVM classifier equivalent to well-known classifiers, such as the Gaussian radial basis function, polynomial of degree d, or multi-layer perceptron [42].

2.4.6 Committee Classifiers

Efforts by several researchers have also been focused upon use of groups, or committees, of classifiers to decide upon class inclusion or exclusion. These committees typically use a majority voting scheme, or an averaging scheme, for example. We propose a variation on the traditional committee classifier in Chapter 5, Section 5.3.4.

2.5 Performance Evaluation

2.5.1 Performance Metrics

Relevant performance metrics include sensitivity, specificity, positive predictive value (PPV), accuracy, and area under the receiver operating characteristic (ROC) curve [81]. Sensitivity measures the number of correct model-labeled event cases out of the total number of actual event cases, while specificity measures the number of correct model-labeled non-event cases out of the total number of actual non-event cases. Positive predictive value is calculated by the number of correct model-labeled events, divided by the number of all model-labeled events (correct and incorrect). Accuracy is calculated by the number of correct model-labeled sets of derived values (event or non-event) divided by the total number of sets of derived values evaluated. To further illustrate these metrics, Figure 2-7 shows a contingency table in which the two classes are called '0' (non-event) and '1' (events). The vertical groupings of cases are gold standard-labeled cases belonging to each class. The horizontal groupings are model-labeled cases belonging to each class. For example, 'a' cases and 'c' cases were actual class '0' cases, though 'a' cases and 'b' cases were labeled by a particular model as class '0'. Similarly, 'b' cases and 'd' cases were actual class '1' cases, while 'c' cases and 'd' cases were labeled by a particular model as class '1'. The relevant performance metrics in equation form are then as follows:

$$Sensitivity = \frac{d}{b+d} = \frac{number\ of\ correct\ model-labeled\ event\ cases}{total\ number\ of\ event\ cases}$$

$$Specificity = \frac{a}{a+c} = \frac{number\ of\ correct\ model-labeled\ non-event\ cases}{total\ number\ of\ non-event\ cases}$$

$$PPV = \frac{d}{c+d} = \frac{number\ of\ correct\ model-labeled\ event\ cases}{number\ of\ model-labeled\ event\ cases\ (correct\ and\ incorrect)}$$

$$Accuracy = \frac{a+d}{a+b+c+d} = \frac{number\ of\ correct\ model-labeled\ event\ and\ non-event\ cases}{total\ number\ of\ cases}$$

The ROC curve is a plot of sensitivity versus one minus specificity. Because sensitivity and specificity can be inversely varied by altering the threshold at which to categorize a case as one class or the other, the area under the ROC curve more effectively describes a model's discriminatory

	Actual class:	
	0	1
Classified as:		
0	a	b
1	c	d

Figure 2-7: Contingency table to illustrate performance metrics. (See text for details.)

ability.

For decision trees, ROC curves are determined by first assigning to each tree leaf the probability of being an event for a set of derived values that percolates to that point. These probabilities are based upon the ratio of events to non-events that fall into each leaf during training. The threshold for considering a case to be event or non-event is then set at each leaf probability value. The resulting sensitivity-specificity pairs, when plotted on a grid of sensitivity versus (1-specificity), gives the ROC curve.

For logistic regression models, various threshold values, ranging from 0 to 100% inclusive, can be used for determining whether to label a case as event or not. The resulting sensitivity-specificity pairs can then similarly be plotted. The area under each ROC curve can be calculated by trapezoidal method to avoid over-estimation of the actual area.

ROC curves can similarly be plotted for neural networks, radial basis function classifiers, and support vector machine classifiers; this functionality is provided by LNKnet.

2.5.2 Testing

Final models should ideally be tested by three methods. First, models (after experimentation with the training and evaluation data are completed) should be tested on their own reserved test sets.

When possible, a model should additionally be tested on a completely different data set, collected from a different place or time. This provides a better measure for how robust a model might be.

Finally, before any model should be used in real-life situations, prospective evaluation in the setting in which it would be used allows for better assessment of actual performance in detecting events of interest.

Chapter 3

Background: Intensive Care Unit Monitoring

This chapter gives an introduction to intensive care unit patient monitors and alarms by presenting, first, some general information about ICU alarms, and then, the methodology and results of an observational study that we performed to further investigate this domain for potential application of the TrendFinder paradigm. We conclude the chapter with a discussion aimed at providing motivation for the TrendFinder applications described in Chapters 4 and 5.

3.1 Overview

The modern intensive care unit is teeming with bedside monitoring devices that generate voluminous amounts of data per patient. While few would regard these data as completely unnecessary and most believe these devices to be useful for improving patient safety [203], such monitors have more recently been considered a source of too much information, or data overload [211]. Data overload in the setting of increased patient numbers and nursing staff shortages contributes to the impossibility for any one patient to be continually and closely followed by a care provider. This means that patient interventions (e.g., therapy changes) necessarily occur intermittently and possibly infrequently; soundings of alarms therefore become crucial indicators of a patient's deteriorating condition or need for assistance. In actuality, however, how useful are these ICU alarms? To answer this question, we performed the following study.

3.2 Observational Study of ICU Alarms

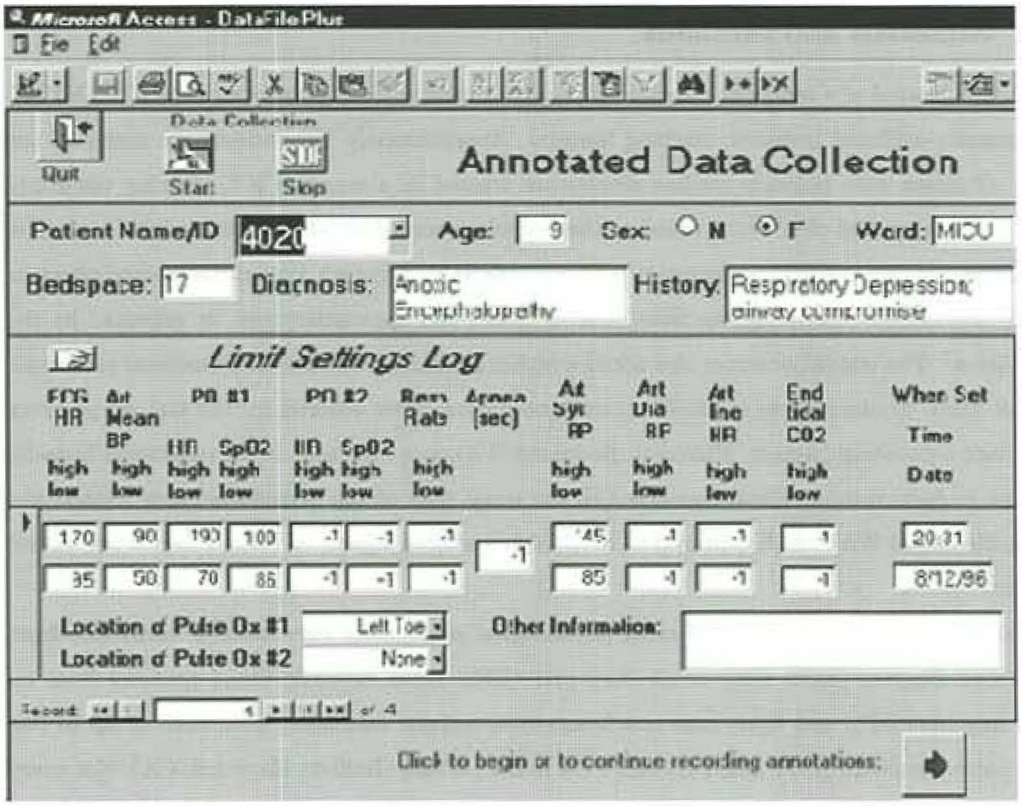
This prospective, observational study was carried out to facilitate a better understanding of ICU alarms in a pediatric intensive care unit. Specifically, our goal was to make an accurate assessment of the positive predictive value of different monitoring devices, as well as the common causes for false positive alarms in the ICU.

3.2.1 Materials and Methods

The study consisted of a ten-week monitoring period in the multidisciplinary medical ICU (MICU) of a university-affiliated pediatric teaching hospital. Approximately 120 patients are admitted per month. Children with primary cardiac disease are treated in a separate ICU. During the study period, a single trained observer situated at the bedside recorded all alarm occurrences for devices being tracked within a single bedspace at a time in the ICU. For each alarm, the trained observer recorded annotations, including the time, source, cause, and appropriateness, as validated by the bedside nurse. The trained observer also noted whether an alarm was silenced by medical personnel. Monitors likely to alarm were studied. Alarm annotations were entered by the trained observer directly into a database (Access, Microsoft, Redmond WA) on a computer at the monitored bedside. The same bedside computer was connected to the serial port of the bedside monitor (SpaceLabs Medical, Redmond WA) which captured all available monitored signals on disk in five to ten second intervals.

Annotations for alarms of the following physiological parameters were tracked: heart rate from an electrocardiogram (ECG HR); respiratory rate (RR); mean systemic blood pressure from an arterial line (Art BP); and heart rate and hemoglobin oxygen saturation from each of up to two Nellcor pulse oximeters (PO1 HR, PO1 sat, PO2 HR, PO2 sat) (Nellcor, Hayward, CA). Not every device was in use at all times; the devices that were present during each session were also recorded into the collection program. Threshold limits for each alarm were determined and set by medical personnel without regard that the monitor was being monitored; these limits were recorded as well. Figure 3-1 shows the computer interface that allows recording of these limits along with basic patient information. The interface used for recording event annotations was a slightly earlier version of the interface shown in Chapter 5, Figure 5-1.

Each recorded alarm was classified, at the time of its occurrence, into one of the following three categories, in which the terms 'relevance' and 'irrelevance' refer to whether or not an alarm was



Microsoft Access - DataFilePlus

Annotated Data Collection

Patient Name/ID: 4020

Age: 9

Sex: ○ M ● F

Ward: MICU

Bedspace: 17

Diagnosis: Anoxic Encephalopathy

History: Respiratory Depression; airway compromise

Limit Settings Log

ECG HR	Art Mean BP	PO #1 HR	PO #1 SpO2	PO #2 HR	PO #2 SpO2	Resp Rate	Apnea (sec)	Art Sys BP	Art Dia BP	Art Ine HR	End tidal CO2	When Set
high low	high low	high low	high low	high low	high low	high low		high low	high low	high low	high low	Time Date
170	90	190	100	-1	-1	-1	-1	145	-1	-1	-1	20:31
95	50	70	85	-1	-1	-1		85	-1	-1	-1	8/12/96

Location of Pulse Ox #1: Left Toe

Location of Pulse Ox #2: None

Other Information:

Record: 1 of 4

Figure 3-1: Computer interface for recording limit settings and basic patient information in the MICU.

indicative of a patient condition that required prompt medical attention:

1. 'True Positive, Clinically Relevant' (TP-R or TPR). TP-R was used to indicate the monitoring device sounded an alarm, the alarm was appropriate given the actual data value as compared to the set threshold value, and the patient's condition required prompt attention. For example, a patient suddenly develops a dysrhythmia with a heart rate of 200 beats per minute (bpm), the ECG measures 200 bpm, the monitor is set with an upper threshold at 160 bpm, and the monitor sounds an alarm.

2. 'True Positive, Clinically Irrelevant' (TP-I or TPI). TP-I was used to indicate the monitor sounded an alarm, the alarm was appropriate given the input data value as compared to the set threshold value, but the patient's condition had not changed in a way that required additional medical attention. The sounding of the alarm was thus irrelevant. For example, a patient's systolic blood pressure transiently crosses the set upper threshold during endotracheal suctioning. The alarm accurately reflects the reading, but the patient's systolic blood pressure requires no medical intervention.

3. 'False Positive' (FP). FP was used to indicate that the monitor sounded an alarm, but the alarm was inappropriate given the input data value. For example, a patient has a heart rate of 80 bpm. The ECG electrodes are manipulated. Although the patient's heart rate stays at 80 bpm throughout this period, an alarm sounds. The alarm was false because the reported value did not reflect the patient condition.

The single trained observer additionally annotated any situations in which an acute patient condition occurred but no alarm was triggered (false negatives, FN), or time periods of appropriate monitor silence when no alarms were expected (true negatives, TN). The latter was carried out by having the trained observer, during alarm-free moments, record a start time for a potential true negative period; if no alarms had sounded after 10 to 15 minutes had elapsed, then the trained observer recorded an end time and marked the period as a true negative. In the case that an alarm did sound during the observation period, the potential true negative annotation was discarded, while the interrupting alarm occurrence was recorded.

In a separate, retrospective classification, all alarms were categorized where possible as either 'patient intervention' alarms or 'non-patient intervention' alarms. 'Patient intervention' alarms refer to those alarms clearly associated with the administration by a care giver of some treatment or diagnostic test, such as endotracheal tube suctioning. 'Non-patient intervention' alarms, on the other hand, refer to those alarms clearly not associated with such interactions. Alarms which did not

Table 3.1: Tracked signals: type and duration of monitoring.

type of signal	monitored time (mins)	monitored time (hrs)	% of total time
ECG HR	17877	298	100%
Art BP	14530	242	81%
PO1 HR	13583	226	76%
PO1 sat	16932	282	95%
PO2 HR	1489	25	8%
PO2 sat	1489	25	8%
RR	6011	100	34%

fall obviously into one of these two categories were labeled as 'difficult to classify.' An example of a difficult-to-classify alarm is one that occurs because a probe falls off of a patient whose movements may or may not have been caused by the presence of a care giver intending to administer some treatment.

3.2.2 Results

A total of 2942 alarms were recorded during 298 hours of monitoring. All monitoring was performed on weekdays from approximately 9:00 am to 5:30 pm, with approximately equal representation of any hour in between. Table 3.1 lists the individual lengths of time for which each tracked parameter was followed.

The largest contributor of the 2942 total alarms was the oxygen saturation signal from the primary pulse oximeter (PO1 sat), which accounted for 43% of all recorded alarms. Since not all devices were tracked for an equal number of hours, alarm occurrences were normalized to frequency of alarms per 100 hours of device monitoring. The normalized occurrence rate for the oxygen saturation signal is 32%, which means that PO1 sat was still the most frequently alarming device of those tracked. Table 3.2 lists the breakdown of the 2942 alarms by signal type, along with the relative percentages of occurrence, the count of alarms per 100 hours, and the corresponding normalized percentages of occurrence. Of the 2942 total alarms observed during the study, 86% were false positives, while an additional 6% were classified as clinically irrelevant true alarms. Only 8% of all alarms tracked during the study period were determined to be true alarms with clinical significance. Nearly all monitored signals had false positive alarm rates exceeding 90%; the two exceptions were the respiratory rate signal, which had a false positive rate of 75%, and the arterial

Table 3.2: Frequencies of occurrence for different alarm types.

type of signal	count of alarms	% of total monitored alarms	count per 100 hours	normalized % of occurrence
ECG HR	464	16%	156	11%
Art BP	400	14%	165	12%
PO1 HR	597	20%	264	19%
PO1 sat	1270	43%	450	32%
PO2 HR	21	<1%	84	6%
PO2 sat	32	1%	128	9%
RR	158	5%	158	11%

Table 3.3: Categorization of all alarms.

type of signal	TP-R count	TP-R %	TP-I count	TP-I %	FP count	FP %
ECG HR	19	4%	15	3%	430	93%
Art BP	151	38%	69	16%	180	46%
PO1 HR	1	<1%	0	0%	596	100%
PO1 sat	61	5%	51	4%	1158	91%
PO2 HR	0	0%	1	5%	20	95%
PO2 sat	0	0%	0	0%	32	100%
RR	6	4%	33	21%	119	75%
Totals	238	8%	169	6%	2535	86%

line mean blood pressure signal, which had a false positive rate of 46%. The arterial line mean blood pressure signal also had the highest rate (38%) of clinically significant true positive alarms, while all other monitored signals had clinically significant true positive rates of 5% or lower. Table 3.3 and Figure 3-2 show the breakdown of each alarm type into these three categories: TP-R (true positive, clinically relevant), TP-I (true positive, clinically irrelevant), and FP (false positive).

Of the total 2942 alarms, 536 (18%) were associated with patient interventions, while 2165 (74%) occurred when no such interventions were taking place. The relationships of the remaining 241 alarms (8%) to patient interventions were deemed ambiguous. Table 3.4 shows for each signal type the alarms classified into one of these three categories: 'patient intervention' alarms, 'non-patient intervention' alarms, and 'difficult to classify' alarms. Patient intervention alarms had an overall false positive rate of 82% and an overall clinically significant true positive rate of 2%, while non-patient intervention alarms had an overall false positive rate of 86% and an overall clinically significant true positive rate of 11%. Table 3.5 presents the breakdown of patient intervention alarms

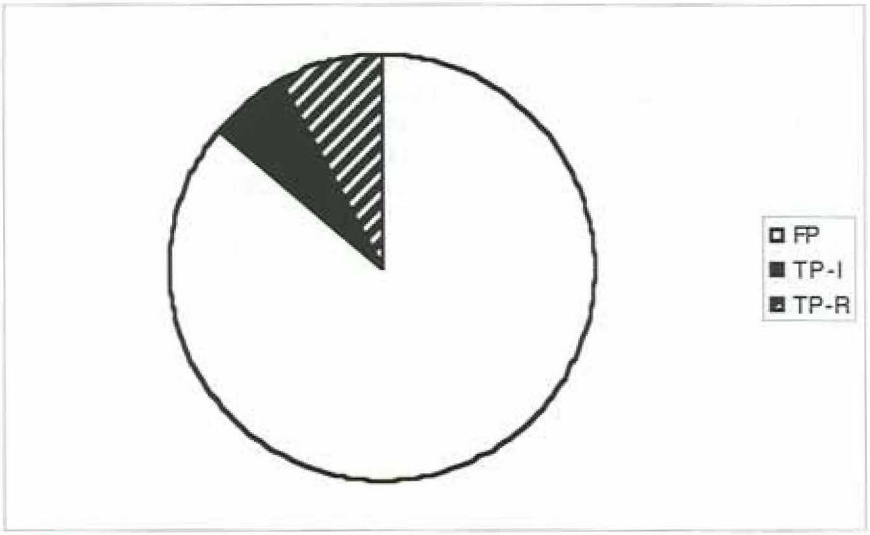


Figure 3-2: Breakdown of all alarms by type: 86% false positive alarms indicated by solid white (FP), 6% clinically-irrelevant true positive alarms indicated by solid black (TP-I), 8% clinically-relevant true positive alarms indicated by black stripes on white (TP-R).

for each signal type into the TP-R, TP-I, and FP categories, while Table 3.6 similarly shows this information for non-patient intervention alarms.

For each monitored signal, the positive predictive value was determined for both patient intervention alarms and non-patient intervention alarms. The positive predictive value was calculated by dividing the number of true positives by the number of all positives for each alarm type:

$$PPV = \frac{TP-R + TP-I}{TP-R + TP-I + FP} = \frac{number\ of\ true\ positives}{number\ of\ all\ positives}$$

$$clinically - relevant\ PPV = \frac{TP-R}{TP-R + TP-I + FP} = \frac{number\ of\ clinically-relevant\ true\ positives}{number\ of\ all\ positives}$$

The 'clinically relevant positive predictive value' was also calculated for each alarm type. This

Table 3.4: Alarms classified by relationship to patient interventions.

type of signal	count of total alarms	number of patient intervention alarms	number of non-patient intervention alarms	number alarms difficult to classify
ECG HR	464	132	328	4
Art BP	400	189	211	0
PO1 HR	597	49	451	97
PO1 sat	1270	92	1073	105
PO2 HR	21	4	7	10
PO2 sat	32	5	17	10
RR	158	65	78	15
Totals	2942	536 (18%)	2165 (74%)	241 (8%)

Table 3.5: Categorization of patient intervention alarms.

type of signal	TP-R count	TP-R %	TP-I count	TP-I %	FP count	FP %
ECG HR	0	0%	8	6%	124	94%
Art BP	8	4%	50	26%	131	69%
PO1 HR	0	0%	0	0%	49	100%
PO1 sat	1	1%	15	16%	77	83%
PO2 HR	0	0%	0	0%	4	100%
PO2 sat	0	0%	0	0%	5	100%
RR	0	0%	15	23%	50	77%
Totals	9	2%	88	16%	440	82%

Table 3.6: Categorization of non-patient intervention alarms.

type of signal	TP-R count	TP-R %	TP-I count	TP-I %	FP count	FP %
ECG HR	19	6%	6	2%	303	92%
Art BP	143	68%	14	7%	54	26%
PO1 HR	1	<1%	0	0%	450	>99%
PO1 sat	60	6%	32	3%	980	91%
PO2 HR	0	0%	1	14%	6	86%
PO2 sat	0	0%	0	0%	17	100%
RR	6	8%	14	18%	58	74%
Totals	229	11%	67	3%	1868	86%

Table 3.7: Positive predictive values of patient intervention alarms.

type of signal	positive predictive value	clinically relevant positive predictive value
ECG HR	6%	0%
Art BP	31%	4%
PO1 HR	0%	0%
PO1 sat	17%	1%
PO2 HR	0%	0%
PO2 sat	0%	0%
RR	23%	0%
Overall	18%	2%

value groups the clinically-irrelevant alarms with the false alarms rather than with the true alarms since TP-I alarms, although true, are nevertheless irrelevant as far as the patient's condition is concerned and thus do not contribute to an alarm's useful predictive value. Tables 3.7 and 3.8 display these calculated predictive values for patient intervention and non-patient intervention alarms, respectively. All of the monitored signals during patient interventions have clinically relevant positive predictive values of less than 5%. For periods not associated with patient interventions, all but one of the monitored signals have clinically relevant positive predictive values of less than 9%; the exception is the mean arterial blood pressure signal, with a clinically-relevant positive predictive value of 68%.

The causes of the observed false positive alarms have been grouped into ten categories, with

Table 3.8: Positive predictive values of non-patient intervention alarms.

type of signal	positive predictive value	clinically relevant positive predictive value
ECG HR	8%	6%
Art BP	74%	68%
PO1 HR	<1%	<1%
PO1 sat	9%	6%
PO2 HR	14%	0%
PO2 sat	0%	0%
RR	26%	8%
Overall	14%	11%

Table 3.9: Common causes of all observed false positive alarms.

cause of false positive alarm	Count	%
PO bad format/connection	1136	45%
PO poor contact	493	19%
ECG wire movement	264	10%
Motion artifact	201	8%
Art line clamp/flush	118	5%
Probe disconnect	61	2%
Unannotated	41	2%
Ventilator disconnect	27	1%
Equipment malfunction	25	1%
(Other causes)	169	7%

almost half of the total 2535 false alarms due to 'bad data format/bad connection' of the pulse oximeter. Motion artifact includes patient movement as well as movement of wires or tubes, for example, by medical personnel. The 'unannotated' group refers to those alarms which were recorded by the trained observer but which were not able to be immediately verified by a study-participating nurse. Only 2% of all false alarms in the study could not be verified. Table 3.9 lists in descending frequency of occurrence the common causes of all false positive alarms that were recorded in the pediatric ICU.

After 298 hours of monitoring, a total of 362 true negative periods, each lasting from 10 to 15 minutes, were recorded. The total amount of time recorded as true negative periods amounted to 5224 minutes, or 87 hours. Not a single false negative entry was recorded by the trained observer.

Table 3.10: Commonly recorded reasons for alarm silencing.

Nurse drawing blood gas
Nurse suctioning
Patient moving
Respiratory signal not picking up
Probe off of patient
Doctor examining patient
Dialysis circuit being changed
Nurse recalibrating machine
Patient being extubated
Leads being disconnected
Leads being reconnected
Medications being changed
Respirator being changed
Arterial line being moved
Cables being changed
Other procedure being performed

During the study period, 325 alarms were noted as being silenced. Of these, 304 entries corresponded to alarms silenced for a short enough duration of time that both the start and the end time of the silencing was recorded. The sum of the known durations was 1173 minutes, or 20 hours. For the remaining 21 silencing episodes, the duration was for an "indefinite" period of time (each episode lasting longer than several minutes). The lower bound on the percentage of total monitoring time for which the tracked signals had their alarms silenced is thus: 1173 min/17877 min = 7%. The commonly recorded reasons for an alarm being silenced are listed in approximately descending order of frequency in Table 3.10. Procedures such as drawing blood gases and suctioning, followed by the category of patient movement, were found to be the most common reasons for alarms being silenced. These three reasons accounted for slightly over half of all alarm silencing occurrences.

3.2.3 Discussion of Study Results

The purpose of this study was to prospectively assess the effectiveness of ICU alarms in alerting medical personnel to clinically significant changes in patient condition. In brief, false alarm rates were found to be extraordinarily high and positive predictive values were found to be extraordinarily low.

This study classified all alarms into three categories for the purpose of better understanding the role each type plays in the ICU. Weese-Mayer and Silvestri stress the importance of such a classification for alarms [217]. Our results extend the findings of other studies: Koski *et al.* found 10.6% of alarms to be significant [109], and Lawless found 5.5% of alarms to be significant [118]. In addition, this study found that alarms not associated with any patient intervention occur more than four times as often as those associated with interventions, at least during the 9:00 am to 5:30 pm period on weekdays. False positive rates are very similar for patient intervention alarms (82%) versus non-patient intervention alarms (86%), whereas most true alarms associated with interventions are clinically irrelevant, but most true alarms not associated with interventions are clinically significant. This observation about true alarms seems reasonable since true alarms caused by administration of some treatment by a care giver are more likely to be clinically insignificant. Interestingly, though, false positive rates are approximately equal in both situations; the implication is that monitors unfortunately do not better alert medical personnel, when they are not attending to the patient, of situations requiring attention. Indeed, overall positive predictive values were found to be equally low (in the teens) for both patient intervention and non-patient intervention alarms.

This study found that 45% of all observed false positive alarms were due to bad data format or bad connection of the pulse oximeter. This particular problem may have been due to either a hardware or software incompatibility between the pulse oximeter (Nellcor) and bedside monitor (SpaceLabs), emphasizing the importance of ascertaining that ICU equipment is functional, updated, and compatible with other devices. It is interesting to note that even if this particular problem had been eliminated before the study period, the overall false positive rate for monitored ICU alarms would only decrease by 9% from 86% (2535/2942) to 77% (1399/1806), and the leading cause of observed false positive alarms would still be due to the pulse oximeter since 'PO poor contact' would comprise 35% (493/1399) of these. In fact, to explore this further, another 32 hours of annotations were collected by the same trained observer several months after the 10 weeks of data collection had ended and after the apparent equipment incompatibility had been resolved. Not a single 'bad data format/bad connection' was observed during these subsequent 32 hours of monitoring and 262 recorded alarms, yet the overall false positive alarm rate was strikingly 86% (225/262) and the pulse oximeter ('PO poor contact') accounted for 48% (107/225) of all observed false positive alarms.

This study has both extended the findings of others and presented new findings, yet some differences do exist between the results of the current investigation and of others: O'Carroll found 75% of alarms from the ECG monitor to be false alarms [150], whereas the false positive rate for the ECG monitor in the present study was 93%. Koski *et al.* found heart rate alarms to be more reliable than the other parameters monitored (arterial, pulmonary, and venous pressures) [109], whereas in this study, ECG heart rate alarms were only better than the pulse oximeter heart rate alarms and otherwise worse than all other monitored signals. Koski *et al.* also found the alarms monitored in the intensive care unit to have true positive rates ranging from 5% to 19% [109]. The overall true positive rates determined in the current study ranged from 0% to 54%. Because all alarms for the tracked bedspace were recorded and only 2% of these were 'unannotated,' the results of the current study are believed to be accurate. Despite such differences, the various studies performed on ICU alarms agree when it comes to concluding that present threshold alarms are oversensitive to false alarms and that there are several problems related to having high false positive alarm rates.

3.3 Discussion of ICU Alarms

An overview of some of the problems with ICU alarms and suggestions for how to remedy some of these is given by Meredith [131]. In practice, wider limits are often used to decrease the frequency of false alarms at the price of not detecting adverse events as quickly. High false positive rates also lead medical personnel to disable these alarm systems [129]. Some find alarms to be annoying and distracting [180] or feel that the false positive alarm rate is too high for the procedure at hand [129] and thus silence the alarms. In our ICU alarm study, it was difficult to determine accurately the amount of time for which alarms were silenced since alarm silencing for 'indefinite' periods of time were difficult to capture. McIntyre asked anesthetists whether they had "ever deactivated an alarm" and found the response to be "Yes" 58% of the time (460/789) [129]. This finding, however, also does not help to determine the length of monitoring time for which alarms become silenced. Chui and Gin describe a potential hazard of the existing alarm system for anesthesia: allowance of the audible warning alarm to sound continuously actually disables the sounding of other alarms [34].

Momtahan *et al.* found that auditory alarms in critical care areas are poorly designed [140]. For example, when alarm soundings are similar or identical, already overburdened medical personnel are easily confused. Moreover, Finley and Cohen [60], as well as Kerr [103], found there to be a "poor match between the importance of the alarm and how urgent it sounds." A specific example of this issue from our ICU alarm study involves the ECG heart rate alarm and the arterial line blood pressure alarm during periods not associated with patient interventions: why do alarms with a positive predictive value of 8% (ECG HR) have the same auditory sound, and thus the same perceived urgency, as those with a positive predictive value of 74% (Art BP)? Westenskow *et al.* found that intelligent alarms can reduce the response time needed by anesthesiologists to detect and correct faults [221], while McIntyre and Nelson suggest the use of automated human voice messages instead of existing alarm sounds [128].

Not surprisingly, noise-induced stress has been found to be a predictor of burnout in critical care nurses [204]. Moreover, the top two out of twelve noises ranked most stressful for nurses were "continuous beeping of patient monitoring devices" and "alarms on equipment" [204]. Effects of noise-induced stress cited include: deficiencies in sustained attention, rapid detection, multiple signal tasks, and incidental memory; decreased altruistic behavior and sensitivity to others; negative affect and interpersonal behavior; and more extreme and premature judgments. Monitor alarms may also be stressful for patients [136].

With so many shortcomings of current ICU alarms, what can be done? The situation cries out for the development of an 'intelligent' monitoring system that not only provides more accurate alerts to care givers, but also has the potential to provide more fine-tuned therapy changes to patients. For example, a child with difficulty breathing may show an increase in heart rate along with decreases in respiratory rate and arterial oxygen saturation (S_pO_2), while a child whose pulse oximeter sensor has fallen off the finger may show a steady heart rate and respiratory rate but a sudden drop in arterial oxygen saturation. A third child may simply have rolled over in his bed, causing no change in heart rate or respiratory rate, but causing an artifactual oxygen saturation reading below the lower threshold alarm limit. Current ICU monitors would sound the same alarm in all of the described cases, thus giving no indication of relative urgency or appropriateness. An intelligent monitor, on the other hand, by recognizing such patterns of changes across multiple physiological signals, could tailor its course of action appropriately: In the first case, the intelligent monitor could set off an urgent alarm and, if the child is being mechanically ventilated, could also make adjustments in ventilator settings such as increasing F_iO_2 (fractional inspired oxygen) and/or $PEEP$ (positive end expiratory pressure). In the second scenario, the intelligent monitor could set off a less urgent alert to indicate that the child appears to be stable but that the oximeter probe has become disconnected. And finally, in the last scenario, the intelligent monitor could recognize that the transient signal change is due to motion artifact and thus no alarm should be given.

While this description of how an intelligent monitor might work is rather straightforward, the development of the 'intelligence' is less so. The following two chapters explore the application of the TrendFinder paradigm to the ICU monitoring domain. The goal is to develop machine-learned multi-signal algorithms that have potential for improving patient monitoring and alarms in the ICU.

The approach is to correlate patterns of physiological signals with clinical events through the application in novel ways of traditional machine learning techniques to time-series clinical data. The primary data signals to be analyzed include pulse oximeter arterial oxygen saturation, electrocardiogram (ECG) heart rate, respiratory rate, and arterial line systolic, diastolic, and mean blood pressures from a multidisciplinary medical ICU; and ECG heart rate, mean blood pressure, and partial pressures of oxygen and of carbon dioxide from a neonatal ICU. These signals are notoriously in need of improvement [62, 104, 206] as well as typically available. Moreover, clinicians usually assess a patient's state by observing heart rate and systemic arterial blood pressure signals [111]. By gleaning knowledge of the physiologic changes in a patient that are associated with particular clinical events, data-assisted development of multi-signal prediction algorithms could serve as a source of intelligence for improved monitors of the future.

Chapter 4

TrendFinder to Detect Artifacts in the Neonatal ICU

4.1 Overview

In the previous chapter, we have seen that the high incidence of false alarms in the intensive care unit necessitates the development of improved alarming techniques. This chapter describes efforts aimed at detecting artifact patterns across multiple physiologic data signals from a neonatal ICU (NICU) using the TrendFinder paradigm for event discovery. Approximately 324 hours of bedside data were analyzed. Artifacts in the data streams were visually located and annotated retrospectively by an experienced clinician. Derived values were calculated for successively overlapping time intervals of raw values, and then used as feature attributes for the induction of models trying to classify 'artifact' versus 'non-artifact' cases. The results are very promising, indicating that integration of multiple signals by applying a classification system to sets of values derived from physiologic data streams may be a viable approach to detecting artifacts in neonatal ICU data.

The remainder of this chapter presents first the methodology and then the results for each set of experiments, including final models and relevant performance metrics. The chapter concludes with a discussion of interesting issues in the area of pattern learning from time-series data and limitations of the work.

4.2 Methods

4.2.1 Event Identification

The event of interest chosen for pattern detection in this set of experiments is signal artifact in monitored signals in the neonatal ICU. To be more precise, four types of artifacts are considered separately within this general category. These correspond to artifacts in each of the four types of signals being monitored: electrocardiogram (ECG) heart rate (HR), arterial line mean blood pressure (BP), partial pressure of carbon dioxide (CO_2), and partial pressure of oxygen (O_2).

4.2.2 Data Collection and Annotation

Data from bedside monitors in the neonatal ICU at Simpson Memorial Maternity Pavilion in Edinburgh, Scotland were collected during 1995 and 1996. Signals analyzed include ECG heart rate, measured in beats per minute; mean blood pressure from an indwelling arterial line, measured in millimeters of mercury; partial pressure of carbon dioxide, collected transcutaneously and measured in kilopascals; and partial pressure of oxygen, also collected transcutaneously and measured in kilopascals. A sample tracing showing three of the signals (heart rate, partial pressure of carbon dioxide, and mean blood pressure) is depicted in Figure 4-1.

Two different sets of data were used for the experiments: Set A consisted of approximately 200 hours of data values occurring at a frequency of one value per minute (1-minute granularity), while Set B consisted of approximately 74 hours of data values occurring at a frequency of one per second (1-second granularity). To produce Set A, approximately three hours of data were first recorded for each of 123 patients. Monitored data not having all four data signals present were not included in further analysis, leaving available for use approximately 200 hours of four-signal data from more than 100 patients. Set B was derived from two different patients, consisting of approximately 24 hours from one patient and 50 hours from a different patient. These data were unrelated to the data used in Set A.

Raw data values were available from bedside monitors at a frequency of one value per second. One-second granularity data, therefore, simply recorded all values coming from the monitors. To collect one-minute granularity data, on the other hand, the arithmetic mean value was calculated for each 60 raw values, for any given signal. Only this mean value was then recorded for that minute's worth of bedside monitoring of that signal.

Artifact occurrences in each of the data streams from both sets of data were visually located and annotated retrospectively by an experienced clinician[1] working constantly with the data collection system in a NICU environment. Artifact annotation consisted of marking the relevant portions of a data stream on a visual system [86]; these markings translated into association of asterisks with those regions deemed to be artifact, one asterisk per signal measurement. A sample of the text data for annotated mean blood pressure values of 1-minute granularity is shown in Figure 4-2; asterisks correspond to the annotations of artifact.

4.2.3 Data Preprocessing

Single-Phase Experiments

We performed experiments to learn both 'single-phase' and 'multi-phase' patterns. In this section, we discuss several issues relevant to both, though specific details refer to the single-phase studies.

Attribute Derivation Data preprocessing included a 'flattening' (abstraction) of the raw temporal data streams into time series features; this consisted of calculating (for each of the four signals) eight quantities that were thought to be potentially clinically useful for ICU event detection. These included moving mean ('avg'), median ('med') , maximum value ('high'), minimum value ('low'), range ('range'), standard deviation ('std_dev'), linear regression slope ('slope'), and absolute value of the linear regression slope ('abs_slope'). These eight quantities were calculated for each successively overlapping set of raw values. Moving mean and median were chosen because of their potential usefulness described in earlier work [123, 205], while the remainder of the derived values were chosen based on perceived clinical usefulness. These derived values comprised the inputs that were later given to machine learning systems.

Each derived value was calculated over a specified time 'window.' For example, a five-minute window would calculate the moving mean for each sequentially overlapping five values. Initially, only data Set A was available. At that time, the number of raw values with which to derive feature attributes had been chosen arbitrarily to be three, five, and ten values (abbreviated as '3-5-10'). That corresponded to time intervals of three minutes, five minutes, and ten minutes, respectively, given the 1-minute granularity nature of the data. Had data with finer granularity (e.g., seconds) been available initially, time windows on the order of seconds would have been preferred due to the presumed fleeting nature of artifacts in ICU data. In all experiments on detecting artifacts in

[1] Neil McIntosh

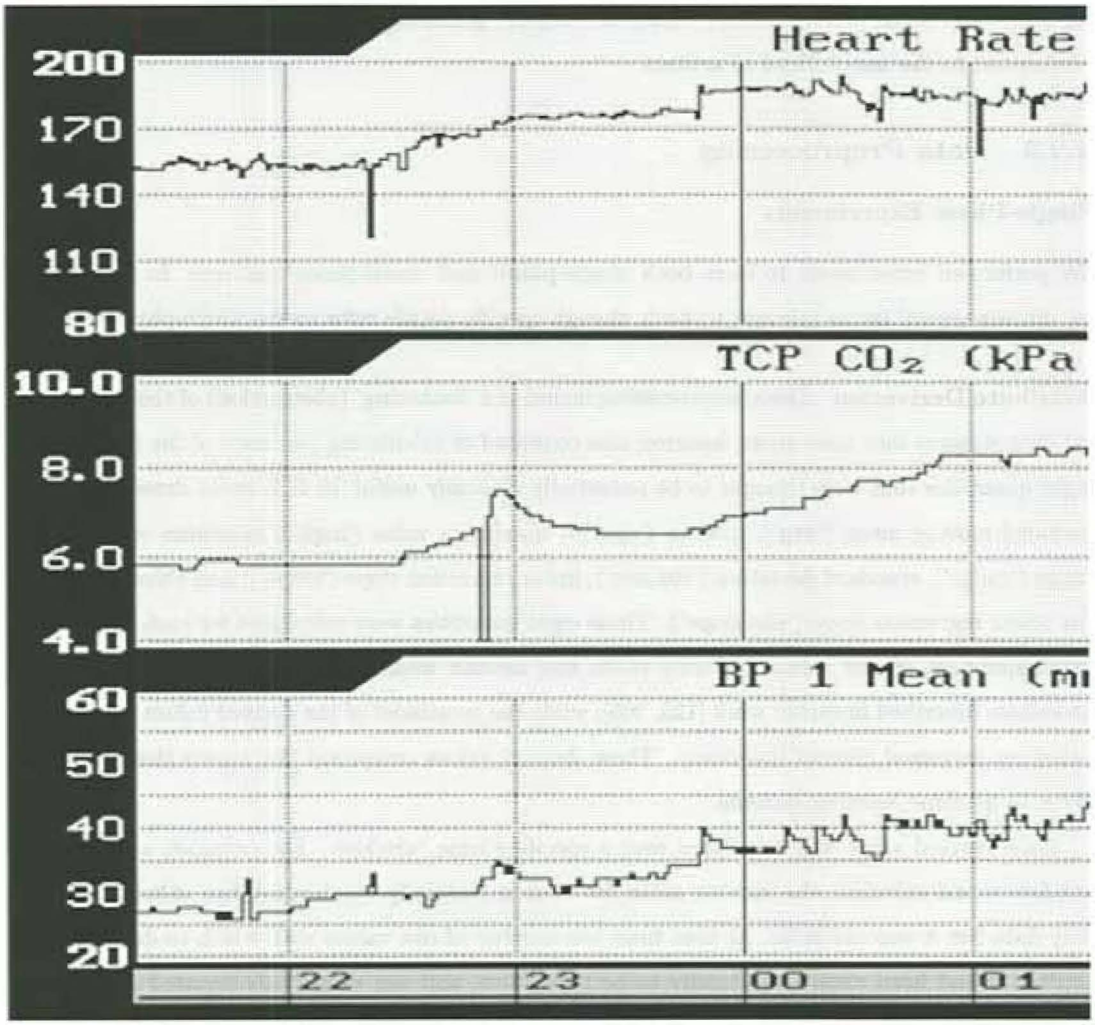


Figure 4-1: Sample tracing from the neonatal ICU.

```
63
62
59
56
55 *
67 *
74 *
54 *
54 *
53 *
55
55
56
57
57
58
60
62
62
```

Figure 4-2: Sample annotated mean blood pressure values.

neonatal ICU data, the approach taken to time interval selection is the general approach of using only vague understanding of the domain to choose the time intervals. An example of a more principled approach to time interval selection is presented in Chapter 5.

Four variations of time intervals were employed on the data from Set B. Each experiment, for both Sets A and B, consisted of feature vectors consisting of attributes derived from each of three different time intervals. This was felt to give some flexibility in searching for useful time intervals, while keeping the number of feature attributes to a computationally reasonable one. Two of the variations were chosen to allow study of data granularity issues. First, in 'Experiment 1', the numbers of raw values with which to calculate feature attributes were chosen to be 180, 300, and 600 ('180-300-600'), equal to three, five, and ten minutes, such that with 1-second granularity data, the identical quantities (in terms of time interval length) would be calculated. The eight derived features, calculated for each of three time intervals and for each of the four signals, resulted in multi-signal feature vectors of size 96. These feature vectors comprised the inputs to machine learning programs which then learned how to classify artifacts on one of those signals. In addition, the 180-300-600 choice allows the 1-minute artifact detection models (developed on data from Set A) to be directly compared to artifact detection models developed from the 1-second data (Set B). Furthermore, the Set B data could be used as a different test set than the Set A test set to evaluate model robustness of the 1-minute models.

In 'Experiment 2,' the numbers of raw values with which to calculate the feature attributes were chosen to be three, five, and ten values ('3-5-10'), corresponding to three second, five second, and ten second intervals. This was chosen such that the same *numbers* of raw values were used as in the 1-minute experiments, though of course the amount of actual *time* represented by each interval is much shorter (one sixtieth). Feature vectors derived for three, five, and ten seconds were not used to develop classification models; these were only for testing of how the 3-5-10 1-minute models would perform when run on 3-5-10 1-second data.

The other two variations of time intervals allowed experimentation with 'class labeling' techniques (to be discussed in more detail in Section 4.2.3). These experiments used only 24 hours of data from Set B (because they were performed before the additional 50 hours of 1-second granularity data became available). The time intervals used were five seconds, 15 seconds, and 60 seconds ('5-15-60'); and two seconds, three seconds, and five seconds ('2-3-5'). These sets of values were chosen with the knowledge that artifacts tend to occur very briefly, on the order of seconds. The actual numbers were otherwise chosen arbitrarily.

In any given experiment, the eight mathematically-derived features, calculated for each of three time intervals and for each of the four signals, resulted in multi-signal feature vectors of size 96 (8 x 3 x 4). These feature vectors comprise the inputs to machine learning programs, which then try to learn how to classify artifacts on one of those signals. (Again, in the case of the 3-5-10 1-second data granularity experiment, Experiment 2, these feature vectors were only to be run through already-developed models.)

Class Labeling Preprocessing also included assignment of class labels ('artifact', 'non-artifact', or 'transition') to each set of derived values. Class labels were given to each of the four data signals so that learning could in turn be focused upon artifacts in each type of data signal. A class label was assigned based upon the corresponding annotations of the raw data values comprising the smallest time interval being labeled. As described previously, any signal measurement that was considered to be artifact was marked by one asterisk; collectively, the presence or absence of these asterisks comprised the annotations. For example, for a time interval of five minutes, between zero and five asterisks, inclusive, may be associated with those five time points. The average number of asterisks per number of time interval minutes was then calculated to give the 'artifact average.'

Two types of class labeling experiments were studied. One explores 'strictness' of class labeling, while the other explores 'location' of class labeling. These notions were introduced in Chapter 2. Only data from Set B (the 1-second granularity data) were used for specifically studying these class labeling issues. Set A data, of course, were also labeled.

Location of class labeling The two variations of class labeling location studied include 'front-labeling' and 'end-labeling,' as described previously in Section 2.3.2. To review, in front-labeling, the time interval from which the feature vector's label is determined occurs at the 'front' of a stream of data. Longer time intervals thus consist of the raw values from the entire shortest time interval plus however many more raw values are needed that follow the shortest time interval. In contrast, 'end-labeling' places the shortest time interval at the 'end' of the longest time interval. These labeling methods were depicted in Figure 2-5. In these experiments, models–one using front-labeling and one using end-labeling–are derived for each of the four signals. This is done for both 2-3-5 preprocessed feature vectors, as well as 5-15-60 feature vectors.

Strictness of class labeling In the experiments for exploring class labeling strictness, four schemes were compared. In each of these, the thresholds for labeling 'artifact' versus 'non-artifact' were varied. Method 1 uses the most 'strict' criteria for 'artifact' and 'non-artifact' class inclusion. In this method, only feature vectors with artifact average exactly equal to one are labeled 'artifact,' while only feature vectors with artifact average exactly equal to zero are labeled 'non-artifact.' (This is the scheme that is used for the 1-minute model for heart rate.) Method 2 uses 'less strict' criteria for class inclusion: feature vectors with artifact average value greater than 0.8 are labeled artifact, while feature vectors with artifact average less than 0.2 are labeled non-artifact. Method 3 uses an even less strict labeling technique: feature vectors with artifact average greater than 0.5 are labeled artifact, while those with artifact average less than or equal to 0.5 are labeled non-artifact. (This is the scheme that is used for the 1-minute models for blood pressure, carbon dioxide, and oxygen artifact detection models.) Method 4 represents the least strict, or perhaps most difficult, labeling method: all feature vectors with artifact average greater than zero are labeled artifact, while only those feature vectors with artifact average exactly equal to zero are labeled non-artifact. This method is perceived to be most difficult because even those feature vectors which span only one raw value associated with an asterisk will become labeled artifact class. In these class labeling strictness experiments, models are derived from 5-15-60 feature vectors.

For the 1-minute Set A experiments, two different strictness labeling schemes were used, as mentioned. For the blood pressure, carbon dioxide, and oxygen signals, all cases with artifact average greater than 0.5 were labeled artifact, while all cases with artifact average less than or equal to 0.5 were labeled non-artifact. No cases were labeled transition for these three signals. For the heart rate signal, cases with artifact average equal to 1.0 were labeled artifact, cases with artifact average equal to 0 were labeled non-artifact, and all other cases were labeled transition. For all four signals, only artifact and non-artifact cases were subsequently used for model development. All Set A experiments used front-labeling.

Multi-Phase Experiments

Preprocessing for multi-phase experiments is very similar to that for single-phase experiments, except that the target feature attributes to be calculated are not necessarily the same ones, and there is added flexibility in class labeling. These issues are discussed in the following paragraphs.

Attribute Derivation For these experiments, we chose to implement a method for discovering two-phase patterns that could be indicative of signal artifact. The feature attributes we chose to derive are the slope of the first phase and the slope of the second phase, over three possible time intervals for each phase. That is, we calculated linear regression slopes for non-overlapping adjoining regions for each feature vector. This is illustrated in Figure 4-3. We also determined several possible two-phase, or 'biphase,' patterns based on the calculated slopes (as was shown in Figure 2-2). These were given as categorical (symbolic) data to only the c4.5 decision tree classifier system; they were not used as input data for the neural network system or for LNKnet's decision tree classifier. While there could be a maximum of nine possible two-phase patterns based upon our calculated slopes, we chose only to determine a subset of those biphase patterns. Specifically, we determined the biphase patterns for the combinations in which the first phase and the second phase are of the same time interval. This resulted in three possible biphase patterns for each physiological signal type. The other previously used single-phase feature attributes–maximum, minimum, range, average, median, standard deviation, and absolute value of linear regression slope–were not used as inputs for these experiments to essentially force learning to use the slope data. The time intervals chosen for this set of experiments were three, five, and 30 seconds (3-5-30).

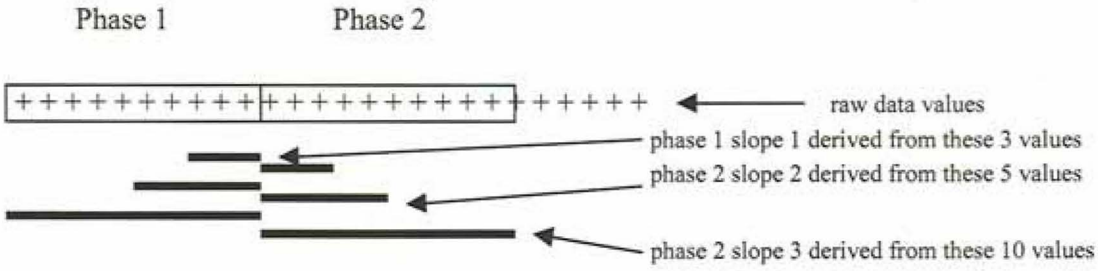

Phase 1
Phase 2
+++++++++++++++++++++++++++++
raw data values
phase 1 slope 1 derived from these 3 values
phase 2 slope 2 derived from these 5 values
phase 2 slope 3 derived from these 10 values

Figure 4-3: Derivation of feature attribute slopes for multi-phase experiments. For example, the first slope of the first phase ('phase 1 slope 1') is calculated from the three raw data values indicated by the dark bar underneath them.

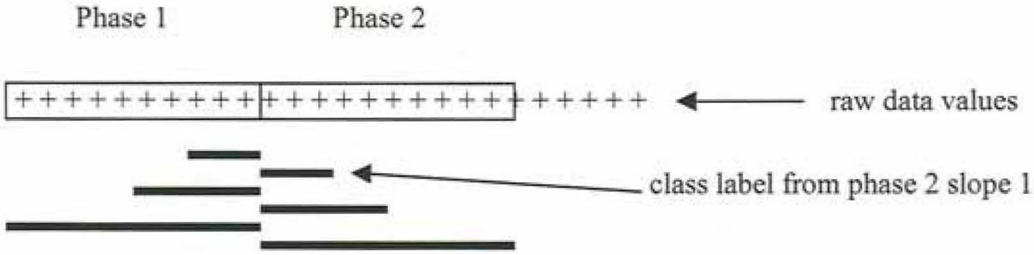


Figure 4-4: Class labeling for multi-phase experiments.

Class Labeling Class labeling for multi-phase pattern learning gives the additional flexibility of specifying from which phase, part of a phase, phases, or parts of phases one wishes to derive the class label. We have chosen, for this set of experiments, to use the label derived from the shortest time interval of the second phase. This corresponds to looking at the qualities of what is going on before the class of interest (during phase one), and then at the class of interest (during phase two). The region from which the class label is derived for our experiments is depicted in Figure 4-4.

Data Partitioning

Set B data that have been preprocessed as described were then split randomly into a training, evaluation, and test set. The training set consisted of 70% of the processed data, while the remaining 30% was further divided into evaluation and test sets. The test set consisted of 70% of the remaining data (21% of the total processed data), while the evaluation set consisted of the other 30% of the remaining data (9% of the total). Again, the evaluation set enables experimentation with model development while avoiding the potential problem of fitting a model to the actual test data. Once experimentation with the training data and evaluation data are completed and a final model is chosen, the test set can then be used to determine model performance.

Data partitioning for Set A was as follows: Set A data were first randomly split into three groups of patients, with roughly the same 70%, 21%, 9% proportions as previously described for training, test, and evaluation sets, respectively. Data for each group of patients were then preprocessed as described.

4.2.4 Model Derivation

Single-Phase Experiments

Comparison of Classifiers The Set A (1-minute) training data were first given to c4.5, a decision tree induction system [166] (described in Section 2.4.3). Decision trees were chosen for this step for both the understandability of their models (e.g., which attributes are important and in what manner) and for their ability to select from amongst a (potentially very large) set of candidate attributes a (potentially much smaller) subset of attributes believed sufficient to classify new cases. Of the 96 derived feature attributes, the decision tree system found those that best divide the remaining data at each level into homogeneous groups of classes according to class label. Systematic experiments in altering tree induction were tried at this step. Two major factors were experimented with: tree-building and tree-pruning. For tree-building, the minimum number of cases, 'm', required in the outcome branches of a candidate test (in order for that test to become additional structure in the tree) was systematically increased until model performance on the evaluation set either improved and then worsened, or just worsened. For tree-pruning, the 'confidence' level, 'c', of pruning was systematically increased until model performance on the evaluation set either improved and then worsened, or just worsened. Improvement was seen as either a decrease in total number of errors on the evaluation set, or a decrease in the size of the tree with little or no increase in the number of errors on the evaluation set. For each level of tree-building tried, tree-pruning was experimented with as described above. Those decision trees with a combination of best performance on the evaluation set and smallest structure were chosen as the final decision tree models (one each for detecting artifact in HR, BP, CO_2, and O_2). For candidate trees with similar performance, the tree that appeared more 'clinically reasonable' was chosen.

The next step in comparing models for detection of artifacts was the development of logistic regression models, again, one for each of the four monitored signals. Each LR model was designed to include exactly those attributes which were present in each final decision tree model. The reason for this approach was twofold. First, decision trees by nature can only divide a multi-dimensional space with plane-parallel boundaries; this may or may not be limiting for describing particular classification domains. LR models do not have this same plane-parallel rigidity. Secondly, previous work in LR has indicated that the difficulty in deriving a good LR model is in choosing which attributes to include or exclude from the model [100]; some have suggested that initially building a decision tree model may be a way of selecting a subset of feature attributes for then building an LR model [208, 219]. In this study, all LR models were built using the JMP statistical package (SAS Institute, Carey, NC) with attributes selected in the decision tree-guided manner described.

Neural network models were also developed for each of the four signals. This was performed using the LNKnet system, described in Section 2.4.2. Numbers of hidden nodes were varied to compare performance of candidate neural networks on the evaluation data set. Best performance on the evaluation set was preferred; in cases of equal performance, a model with fewer hidden nodes was preferred.

To further compare different classification systems, blood pressure artifact detection was chosen for additional model development beyond that already described for all four signals. First, the LNKnet system was used to build a radial basis function classifier. Training, evaluation, and test sets were also given to a colleague[2] at the MIT Center for Biological and Computational Learning, who then used custom-designed software to develop a support vector machine classifier. For the RBF classifier, the numbers of clusters used in the model were varied to compare performance of candidate RBF models. For SVMs, linear, Gaussian, and polynomial kernel functions, with various constant C values (as described in Section 2.4.5), were tried.

Data Granularity Experiments Decision tree models, as described in Section 4.2.4, were developed from both 1-minute data (3-5-10) and 1-second data (180-300-600 and 3-5-10) to perform the data granularity experiments.

Class Labeling Experiments Class labeling experiments were performed by deriving decision tree models from Set B (1-second) data, preprocessed as 2-3-5 attributes and as 5-15-60 attributes.

[2] SVM experiments were performed by Ryan Rifkin at the MIT Center for Biological and Computational Learning.

Criteria for choosing a final model were as described in Section 4.2.4.

Multi-Phase Experiments

Both decision tree models and neural network models were created for the multi-phase experiments. In these experiments, LNKnet was used for development and performance evaluation of both decision trees and neural networks. LNKnet experiments were given only slopes as input feature attributes to enable fair comparisons with their counterpart neural networks. Criteria for choosing a decision tree model were based on performance of candidate trees on the evaluation set. The parameters varied were whether to stop growth of the tree before all training cases were correctly classified, and how much to prune the tree for use in testing; these are similar to the options available in c4.5. Criteria for choosing a neural network model were based upon the same as previously described for neural networks, that is, varied performance on the evaluation set from varying the number of hidden nodes in candidate networks. The c4.5 tree induction system was additionally used to get a better idea for the patterns detected when using both slopes and phase patterns as feature attributes. These models were meant only for qualitative study.

4.2.5 Performance Evaluation

Performance metrics used for comparing different models include sensitivity, specificity, positive predictive value, accuracy, and area under the ROC curve. Sensitivity was calculated by the number of correct model-labeled artifact cases, divided by the number of gold-standard artifact cases. Specificity was calculated by the number of correct model-labeled non-artifact cases, divided by the number of gold-standard non-artifact cases. Positive predictive value was calculated by the number of correct model-labeled artifacts, divided by the number of all model-labeled artifacts (correct and incorrect). Accuracy was calculated by the number of correct model-labeled sets of derived values (artifact or non-artifact) divided by the total number of sets of derived values evaluated. ROC curves and areas were determined as described in Chapter 2. For each LR model, twelve threshold values (0, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100%) were used for determining whether to label a case as artifact or not; the (sensitivity, one-minus-specificity) pairs were then plotted.

The decision tree and logistic regression models, where appropriate, were implemented as computer programs written in the C language to facilitate performance evaluation (i.e., running the models on the reserved test set).

Decision trees that had been developed on Set A (1-minute) data, as described in Section 4.2.4, were not only run on test data derived from the 1-minute data, but were also run on a completely different test data set derived from the 1-second data. For all other classifiers, models developed from 1-minute data were run on test sets processed from 1-minute data, and likewise, models developed from 1-second data were run on test sets processed from 1-second data.

4.3 Results

4.3.1 Data Collection, Annotation, and Preprocessing

Data were collected and annotated in the neonatal ICU at Simpson Memorial Maternity Pavilion as described in the methodology section. Preprocessing, including feature attribute derivation and class labeling for both single-phase and multi-phase learning, was also performed as described in the methodology.

For the 3-5-10 experiments using Set A data (1-minute granularity data), the blood pressure training data set consisted of 275 cases labeled as BP artifacts and 11,096 cases labeled as non-BP artifacts. For carbon dioxide, there were 590 CO_2 artifacts and 10,781 non-CO_2 artifacts. The heart rate training data set consisted of 520 cases labeled as HR artifact and 10,849 cases labeled as non-HR artifact. For oxygen, there were 176 O_2 artifacts and 11,195 non-O_2 artifacts. The blood pressure evaluation data set consisted of 43 BP artifacts and 1404 non-BP artifacts. For CO_2, there were 68 CO_2 artifacts and 1379 non-CO_2 artifacts. The heart rate evaluation data set consisted of 74 HR artifacts and 1377 non-HR artifacts. For O_2, there were 20 O_2 artifacts and 1427 non-O_2 artifacts. The test sets consisted of 78, 183, 130, and 64 artifacts for BP, CO_2, HR, and O_2, respectively, and 3341, 3236, 3287, and 3355 non-artifacts for BP, CO_2, HR, and O_2, respectively. These numbers are tabulated in Table 4.1.

The 1-second data of Set B were preprocessed as described in the methodology section. In the 3-5-10 experiments, for the blood pressure signal, there were 157,210 training cases, consisting of 441 BP artifact class cases and 156,769 non-BP artifact class cases. The blood pressure evaluation set of 20,077 cases contained 50 BP artifact class cases and 20,027 non-BP artifact class cases. The blood pressure test set consisted of 47,419 cases, comprised of 132 BP artifact class cases and 47,287 non-BP artifact class cases. These numbers are tabulated in Table 4.2, along with the numbers for the other three data signals.

The numbers of cases of artifact class and non-artifact class vary slightly for different experiments using Set B data depending on both the time intervals selected for feature derivation and the class

Table 4.1: Breakdown of data cases by class label by each signal type for Set A (3-5-10).

Signal	Class label	Training set	Evaluation set	Test set
Blood pressure	non-artifact	11,096	1404	3341
	artifact	275	43	78
	total	11,371	1447	3419
Carbon dioxide	non-artifact	10,781	1379	3236
	artifact	590	68	183
	total	11,371	1447	3419
Heart rate	non-artifact	10,849	1377	3287
	artifact	520	74	130
	total	11,369	1451	3417
Oxygen	non-artifact	11,195	1427	3355
	artifact	176	20	64
	total	11,371	1447	3419

labeling strictness method used; this is because transition cases are not included in these experiments for the machine learning step. The numbers of cases in each class for 180-300-600 experiments will be similar to the numbers described above for the 1-second 3-5-10 experiment and thus are not presented here. The numbers of cases in each class for 2-3-5 and 5-15-60 are similar to each other but are not similar to the numbers shown in Table 4.2 (3-5-10) because these experiments only used data from 24 hours of Set B. Table 4.3 displays the numbers of cases by class for each data signal artifact type for the 2-3-5 experiment.

4.3.2 Single-Phase Models: Comparison of Classifiers

In this section we present the final single-phase models derived from neonatal ICU data Set A (1-minute granularity) using time intervals of three, five, and 10 minutes. First we present the decision tree models, one for each type of signal artifact. Then we present the logistic regression models for detecting artifacts in each of those four signals. After that, we describe the network structure of the final neural network models for each signal. For the blood pressure artifact detection data, we additionally describe a radial basis function classifier and a support vector machine classifier. Performance of each model on its test set is also described.

Table 4.2: Breakdown of data cases by class label by each signal type for Set B (3-5-10).

Signal	Class label	Training set	Evaluation set	Test set
Blood pressure	non-artifact	156,769	20,027	47,287
	artifact	441	50	132
	total	157,210	20,077	47,419
Carbon dioxide	non-artifact	153,083	19,557	46,178
	artifact	4,127	520	1,241
	total	157,210	20,077	47,419
Heart rate	non-artifact	153,862	19,656	46,419
	artifact	474	76	139
	total	154,336	19,732	46,558
Oxygen	non-artifact	151,646	19,362	45,723
	artifact	5,564	715	1,696
	total	157,210	20,077	47,419

Table 4.3: Breakdown of data cases by class label by each signal type for subset of Set B containing 24 hours of data (2-3-4).

Signal	Class label	Training set	Evaluation set	Test set
Blood pressure	non-artifact	60,600	7712	18,303
	artifact	51	4	10
	total	60,651	7716	18,313
Carbon dioxide	non-artifact	60,625	7714	18,310
	artifact	26	2	3
	total	60,651	7716	18,313
Heart rate	non-artifact	60,025	7640	18,119
	artifact	626	76	194
	total	60,651	7716	18,313
Oxygen	non-artifact	60,500	7692	18,261
	artifact	151	24	52
	total	60,651	7716	18.313

```
bp_med3 <= 4 : 1 (114.0/3.0)
bp_med3 > 4 :
|   bp_range3 <= 7 : 0 (10959.0/72.5)
|   bp_range3 > 7 :
|   |   bp_med10 > 46 : 0 (126.0/23.7)
|   |   bp_med10 <= 46 :
|   |   |   bp_std_dev3 <= 5.51 : 0 (78.0/28.5)
|   |   |   bp_std_dev3 > 5.51 :
|   |   |   |   co2_low10 <= 5.3 : 1 (46.0/10.1)
|   |   |   |   co2_low10 > 5.3 :
|   |   |   |   |   hr_high5 <= 157 : 0 (27.0/12.8)
|   |   |   |   |   hr_high5 > 157 : 1 (21.0/8.2)
```

Figure 4-5: Blood pressure artifact detection decision tree model from 1-minute data.

Decision Tree Models

The final decision trees selected are shown in Figures 4-5 through 4-8. A final label (a single digit following a colon in the text version of a tree model) of '1' corresponds to 'artifact,' while a final label of '0' corresponds to 'non-artifact.' Attribute names are a concatenation of the signal type, abbreviated name of the derived value, and time interval length in minutes. For example, 'ox_std_dev10' refers to the standard deviation of the partial pressure of oxygen signal calculated on time intervals using 10 values (in this case, equal to 10 minutes), while 'hr_med3' refers to the moving median value of heart rate over three-minute time intervals. The numbers in parentheses after a final label indicate first, the number of training cases that reached that leaf and thus were given that label, followed by the number of training cases at that leaf that are not appropriately of that label. For example, in the HR decision tree model, shown in Figure 4-7, the second line is "hr_range5 > 78: 1 (180.0/3.0)." This means that if for a set of derived values the range over five minutes of HR raw values is greater than 78, then this set becomes labeled 'artifact' ('1'); of the cases in the training set, 180.0 cases met this inclusion criteria and thus were labeled artifact. Of these 180.0 cases, 3.0 were actually not artifacts, while the other 177.0 cases were correctly labeled as artifacts. Fractional cases can come about due to pruning of a predecessor tree.

The final decision tree model for BP artifact detection consisted of 13 nodes and six attributes. The final tree was created with c4.5 parameters 'm' = 15 and 'c' = 5. Of 3419 test cases, it correctly classified 3336 non-artifacts and 45 artifacts. It incorrectly classified five non-artifacts and 33 artifacts. Sensitivity for the BP decision tree model was therefore 57.7%; specificity was 99.9%,

```
co2_med5 <= 0.7 : 1 (207.0)
co2_med5 > 0.7 :
|   ox_high3 > 14 : 1 (166.0/34.0)
|   ox_high3 <= 14 :
|   |   co2_range3 <= 0.6 : 0 (10686.0/102.0)
|   |   co2_range3 > 0.6 :
|   |   |   co2_low5 <= 4.5 : 1 (117.0/22.0)
|   |   |   co2_low5 > 4.5 :
|   |   |   |   co2_slope3 <= 0.5 : 0 (150.0/26.0)
|   |   |   |   co2_slope3 > 0.5 : 1 (45.0/17.0)
```

Figure 4-6: Carbon dioxide artifact detection decision tree model from 1-minute data.

```
hr_low3 <= 113 :
|   hr_range5 > 78 : 1 (180.0/3.0)
|   hr_range5 <= 78 :
|   |   hr_low10 <= 30 : 1 (60.0/1.0)
|   |   hr_low10 > 30 :
|   |   |   hr_med5 <= 121 : 0 (145.0/21.0)
|   |   |   hr_med5 > 121 :
|   |   |   |   ox_low10 > 6 : 1 (41.0)
|   |   |   |   ox_low10 <= 6 :
|   |   |   |   |   hr_range3 <= 38 : 0 (30.0/10.0)
|   |   |   |   |   hr_range3 > 38 : 1 (37.0/8.0)
hr_low3 > 113 :
|   hr_std_dev3 <= 8.14 : 0 (10565.0/66.0)
|   hr_std_dev3 > 8.14 :
|   |   hr_range3 > 36 : 1 (41.0/6.0)
|   |   hr_range3 <= 36 :
|   |   |   hr_low5 > 129 : 0 (160.0/28.0)
|   |   |   hr_low5 <= 129 :
|   |   |   |   ox_low10 <= 4 : 0 (35.0/6.0)
|   |   |   |   ox_low10 > 4 :
|   |   |   |   |   bp_abs_slope10 <= 0.21 : 1 (38.0/5.0)
|   |   |   |   |   bp_abs_slope10 > 0.21 : 0 (37.0/15.0)
```

Figure 4-7: Heart rate artifact detection decision tree model from 1-minute data.

```
ox_med3 > 20 : 1 (90.0/1.6)
ox_med3 <= 20 :
|   ox_low3 > 0 : 0 (11026.0/1.6)
|   ox_low3 <= 0 :
|   |   ox_med5 <= 0 : 0 (154.0/5.4)
|   |   ox_med5 > 0 :
|   |   |   ox_med3 <= 0.5 : 1 (79.0/1.6)
|   |   |   ox_med3 > 0.5 : 0 (22.0/6.3)
```

Figure 4-8: Oxygen artifact detection decision tree model from 1-minute data.

positive predictive value was 90.0%, and overall accuracy was 98.9%. The area under the ROC curve for the BP decision tree model was calculated as 89.4%. Figure 4-9 shows this ROC curve. (Recall that sensitivity and specificity are inversely correlated and depend on the threshold at which a model calls a case artifact or not. The sensitivity and specificity values reported here are for the default threshold used by c4.5 during testing, which is approximately 50%.)

The decision tree model for detecting artifacts in CO_2 contained 11 nodes and five attributes. The final tree was created with c4.5 parameters 'm' = 45 and 'c' = 30. The model correctly classified 3209 non-artifacts and 151 artifacts, while incorrectly classifying 27 non-artifacts and 32 artifacts. Sensitivity for the model was calculated to be 82.5%, while specificity, positive predictive value, and accuracy were calculated to be 99.2%, 84.8%, and 98.3%, respectively. The ROC curve for this model is shown in Figure 4-10; the area under the ROC curve was 93.3%.

The decision tree for detecting heart rate artifacts consisted of 23 nodes and included nine attributes. The final tree was created with c4.5 parameters 'm' = 25 and 'c' = 10. On the test set of 3417 cases, it correctly classified 3279 non-artifacts, and incorrectly classified eight non-artifacts as artifacts. It correctly identified 85 artifacts, while not identifying 45 artifacts. Sensitivity for the HR decision tree model was therefore 65.4%; specificity was 99.8%, positive predictive value was 91.4%, and overall accuracy was 98.5%. The area under the ROC curve, displayed in Figure 4-11, was 92.8%.

The final decision tree model for detecting O_2 artifacts contained nine nodes, consisting of three attributes. The final tree was created with c4.5 parameters 'm' = 20 and 'c' = 20. This model classified all 3355 non-artifacts correctly. Of 64 artifacts, it correctly classified 56, missing eight. Sensitivity for this model was 87.5%. Both specificity and positive predictive value were 100%, while accuracy was 99.8%. On the test set, the O_2 decision tree model achieved an area under the ROC

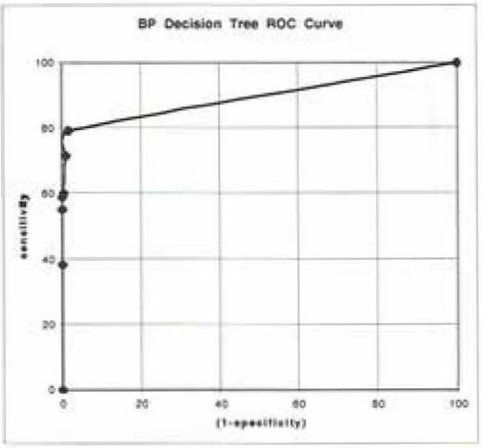


Figure 4-9: NICU blood pressure artifact detection decision tree (3-5-10) ROC curve. Area = 89.4%.

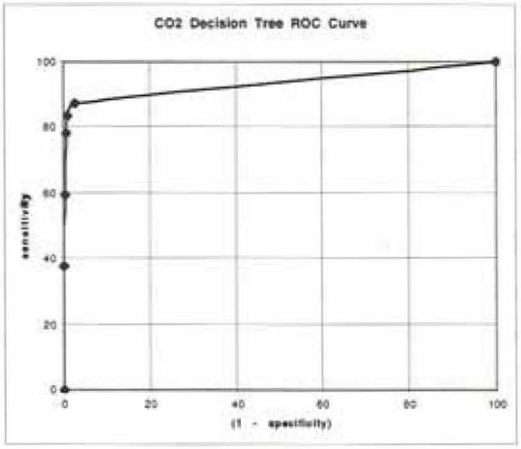


Figure 4-10: NICU carbon dioxide artifact detection decision tree (3-5-10) ROC curve. Area = 93.3%.

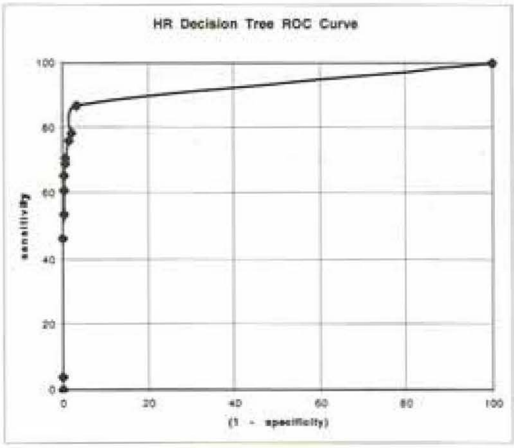


Figure 4-11: NICU heart rate artifact detection decision tree (3-5-10) ROC curve. Area = 92.8%

curve of 99.9%. The ROC curve is shown in Figure 4-12.

Logistic Regression Models

Logistic regression models were derived from the identical training sets used for decision tree development. As described in the methodology, these LR models were designed to include exactly those attributes that are in the corresponding decision tree models. For each of the LR models, a list of the included feature attributes along with their parameter estimates are presented in Tables 4.4 through 4.7.

The areas under the ROC curve for the LR models were 16.7%, 7.8%, 8.9%, and 26.8% for BP, CO_2, HR, and O_2 signals, respectively. The ROC curves are shown in Figures 4-13 to 4-16. An example of how the numbers for sensitivity, specificity, positive predictive value, and accuracy looked are as follows: for the O_2 LR model with a threshold set at 80%, values were 85.9%, 0.1%, 1.6%, and 1.8%, respectively.

After the results of the LR models were determined, an additional LR model was created for O_2 artifact detection to determine the effect of 'dichotomizing' one of the attributes from the original O_2 LR model. The attribute 'ox_med3' was split into two new variables, 'high_ox_med3' and 'low_ox_med3,' where high_ox_med3 was set to 1 if and only if the value of ox_med3 was greater than

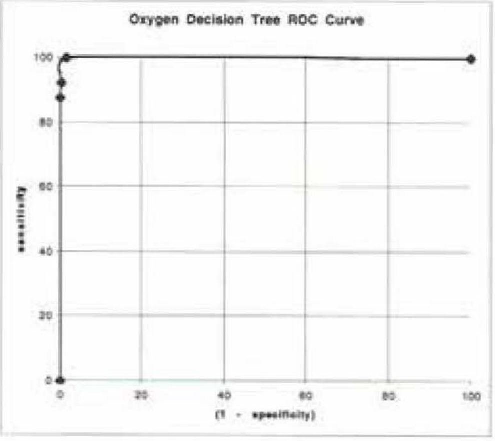


Figure 4-12: NICU oxygen artifact detection decision tree (3-5-10) ROC curve. Area = 99.9%.

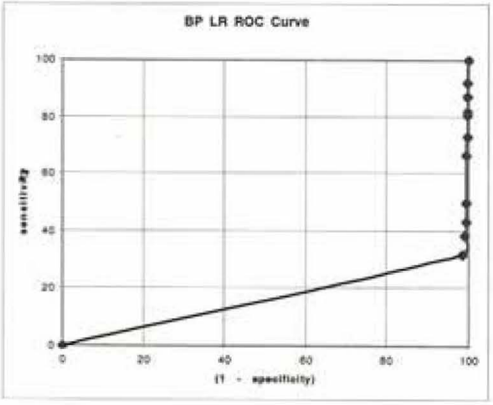


Figure 4-13: NICU blood pressure artifact detection LR (3-5-10) ROC curve. Area = 16.7%.

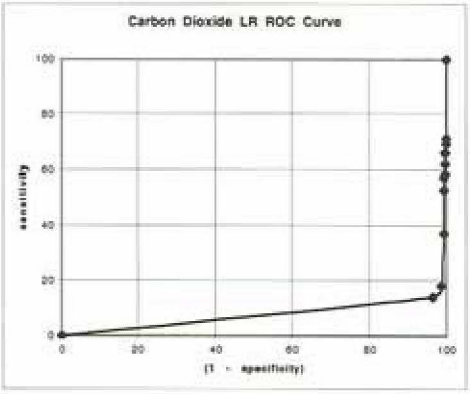


Figure 4-14: NICU carbon dioxide artifact detection LR (3-5-10) ROC curve. Area = 7.8%.

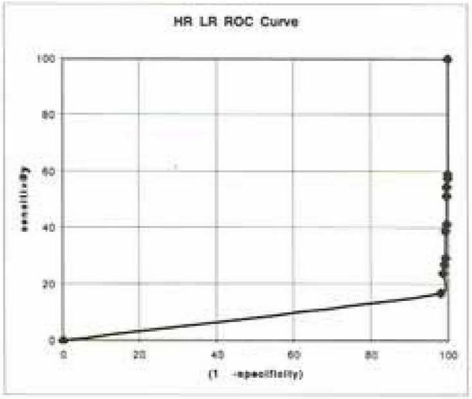


Figure 4-15: NICU heart rate artifact detection LR (3-5-10) ROC curve. Area = 8.9%

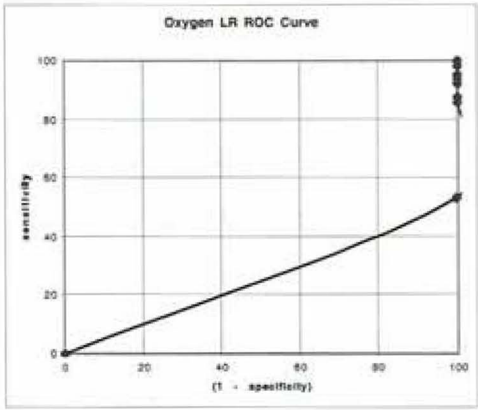


Figure 4-16: NICU oxygen artifact detection LR (3-5-10) ROC curve. Area = 26.8%.

Table 4.4: Blood pressure artifact detection logistic regression model.

Attribute	Value
intercept	-2.085
bp_med3	0.147
bp_range3	-0.675
bp_med10	0.009
bp_std_dev3	0.506
co2_low10	0.059
hr_high5	0.013

20, and low_ox_med3 was set to 1 if and only if the value of ox_med3 was less than or equal to 0.5. The thresholds (20 and 0.5) at which to dichotomize this attribute were the same as those present in the O_2 decision tree model. With this dichotomization, the new LR model achieved an area under the ROC curve of 86.0%. The attributes and their parameter estimates for the O_2 LR model with dichotomization are presented in Table 4.8.

Neural Network Models

Four neural network (multi-layer perceptron) models were created, one for detection of artifacts of each signal type. The neural network for detecting blood pressure artifacts had 96 inputs, 30

Table 4.5: Carbon dioxide artifact detection logistic regression model.

Attribute	Value
intercept	0.769
co2_med5	0.007
ox_high3	0.005
co2_range3	-3.136
co2_low5	0.727
co2_slope3	-2.595

Table 4.6: Heart rate artifact detection logistic regression model.

Attribute	Value
intercept	0.198
hr_low3	0.110
hr_range5	-0.007
hr_low10	0.009
hr_med5	-0.098
ox_low10	-0.102
hr_range3	0.291
hr_std_dev3	-0.641
hr_low5	0.019
bp_abs_slope10	0.072

Table 4.7: Oxygen artifact detection logistic regression model.

Attribute	Value
intercept	6.078
ox_med3	-0.303
ox_low3	0.223
ox_med5	-0.118

Table 4.8: Oxygen artifact detection logistic regression model with dichotomization.

Attribute	Value
intercept	-0.002
high_ox_med3	0.988
low_ox_med3	0.353
ox_low3	-0.007
ox_med5	0.007

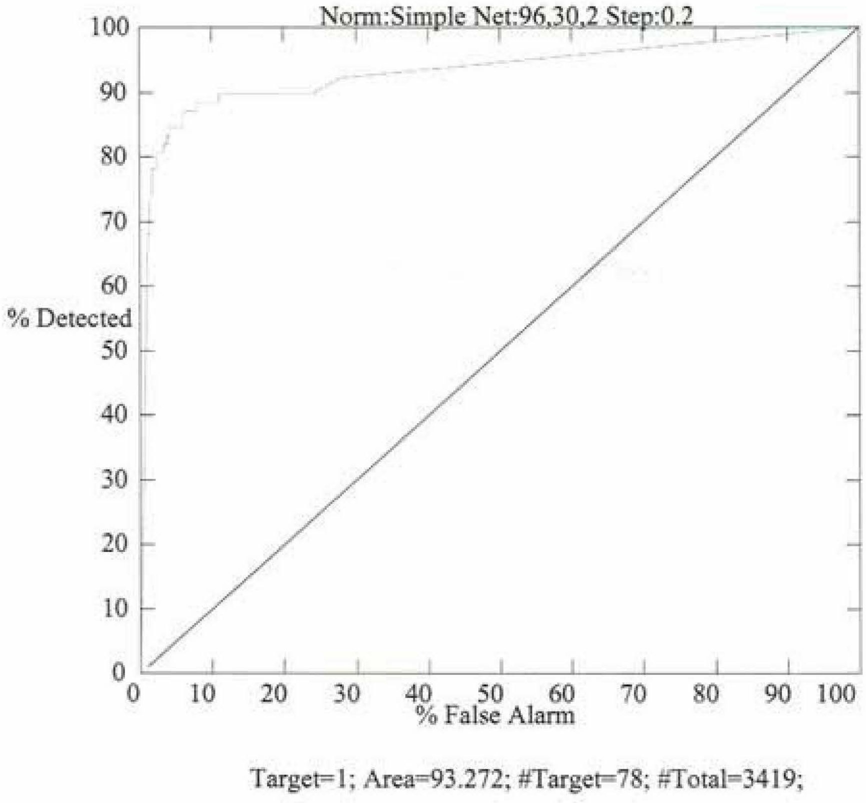


Figure 4-17: ROC curve for blood pressure artifact detection neural network model (1-minute data).

hidden nodes in a single layer, and 2 output nodes. Its area under the ROC curve was 93.27%; the ROC curve is shown in Figure 4-17. The neural network for detecting CO_2 artifacts had 96 inputs, 15 hidden nodes in a single layer, and 2 outputs. It achieved an area under the ROC curve of 97.46%, shown in Figure 4-18. The heart rate neural network had 96 input nodes, one hidden layer containing 30 nodes, and 2 output nodes. The resulting area under the curve, shown in Figure 4-19, was 96.62%. Finally, the oxygen neural network had 96 inputs, one hidden layer containing 20 nodes, and 2 output nodes. The area under the ROC curve for the oxygen artifact detection network was 99.31%; this is shown in Figure 4-20.

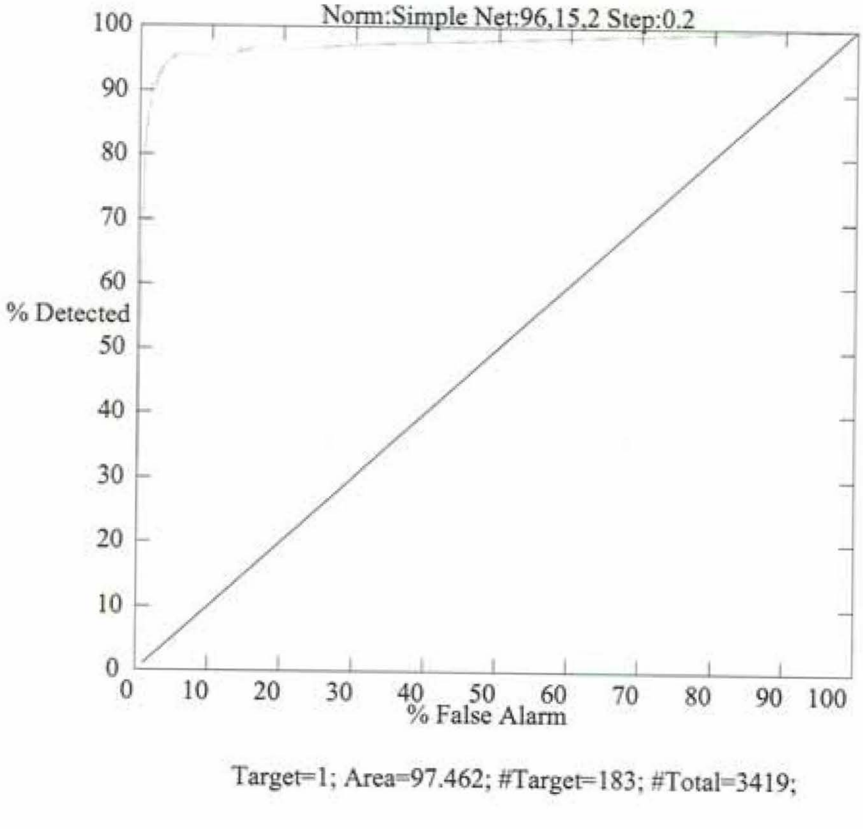


Figure 4-18: ROC curve for carbon dioxide artifact detection neural network model (1-minute data).

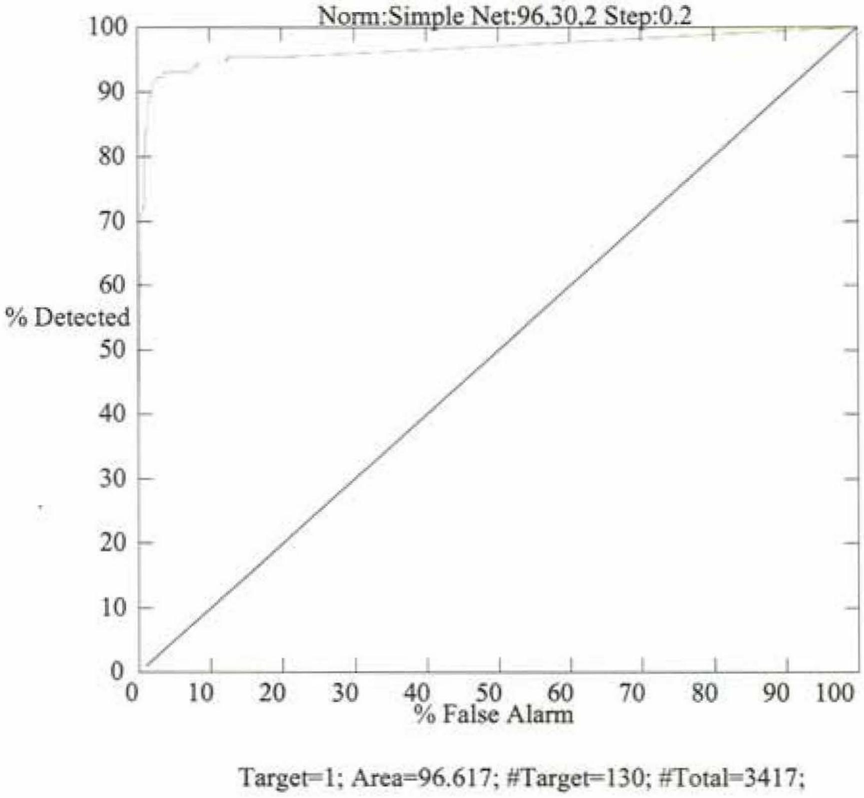
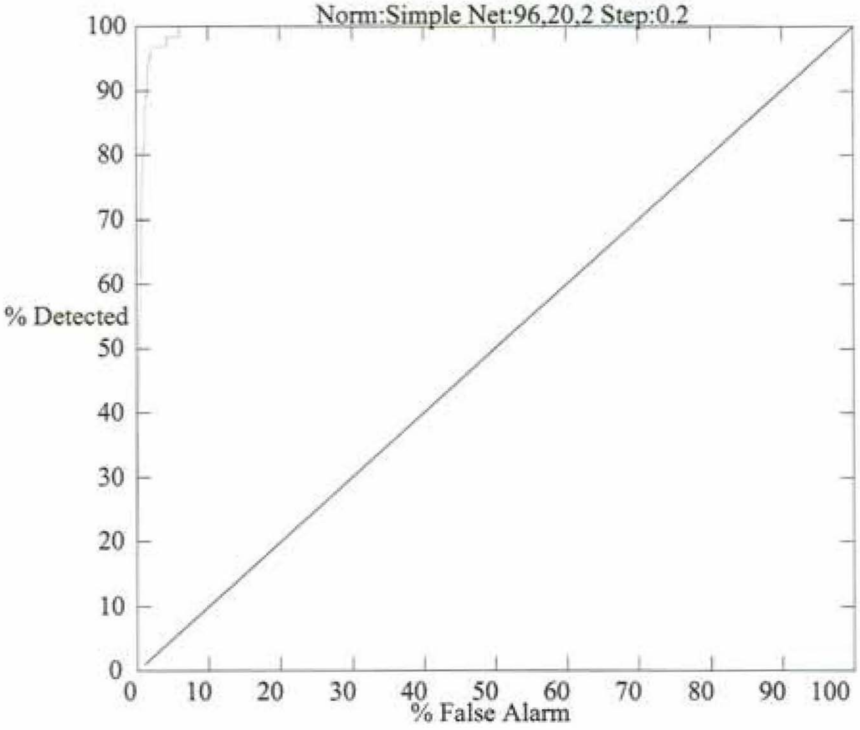


Figure 4-19: ROC curve for heart rate artifact detection neural network model (1-minute data).



Target=1; Area=99.307; #Target=64; #Total=3419;

Figure 4-20: ROC curve for oxygen artifact detection neural network model (1-minute data).

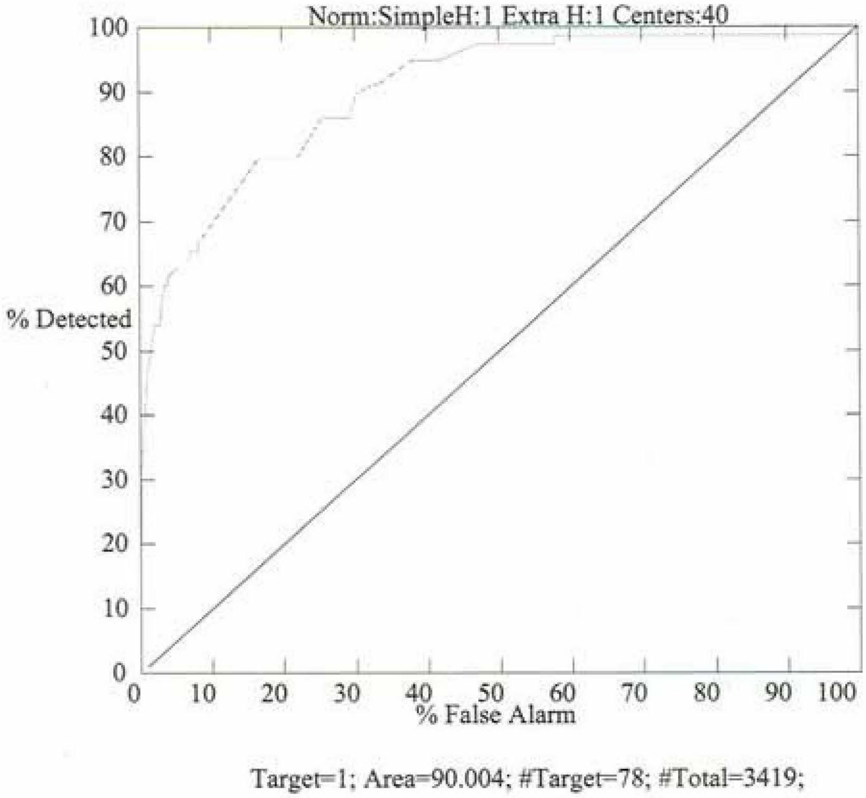


Figure 4-21: ROC curve for blood pressure artifact detection radial basis function network model (1-minute data).

Radial Basis Function Model for BP

More than twenty experiments with creating a radial basis function classifier for BP artifact detection were tried. The model that performed best on the evaluation set was chosen as the final RBF model. It had 20 clusters per class to give a total of 40 clusters; these were formed by K-means clustering. The resulting area under the ROC curve for the RBF classifier was 90.00%; this is shown in Figure 4-21.

```
bp_med180 <= 0 : 1 (441.0)
bp_med180 > 0 : 0 (156769.0)
```

Figure 4-22: Blood pressure artifact detection decision tree model from 1-second data (180-300-600).

```
co2_med180 <= 0.2 : 1 (4053.0/3.0)
co2_med180 > 0.2 :
|   ox_low180 > 16 : 1 (30.0/3.0)
|   ox_low180 <= 16 :
|   |   co2_avg180 > 0.4 : 0 (153040.0/9.0)
|   |   co2_avg180 <= 0.4 :
|   |   |   hr_range180 <= 51 : 1 (40.0)
|   |   |   hr_range180 > 51 : 0 (47.0/1.0)
```

Figure 4-23: Carbon dioxide artifact detection decision tree model from 1-second data (180-300-600).

Support Vector Machine Model for BP

More than forty experiments with creating an SVM classifier for BP artifact detection were performed. Linear, Gaussian, and polynomial kernel functions were tried with various values for the real-valued parameters to the SVM system. The best results on the evaluation data set came with using a Gaussian kernel function. On the test set, this SVM classifier achieved an area under the ROC curve of 95.95%.

4.3.3 Single-Phase Models: Data Granularity Experiments

Figures 4-22 through 4-25 show the final decision trees for artifact detection developed from 1-second training data preprocessed as described for Experiment 1. Again, class labels are represented by '1' for artifact class and '0' for non-artifact class. Parentheses after a class label indicate the number of training cases that arrived at that node, shown where applicable as the total number of training cases that arrived at that node followed by the number of training cases that were incorrectly classified at that node. Attribute names are a concatenation of the abbreviation of the signal name, an abbreviation of the derived feature name, and the number of values over which the derived feature was calculated.

Final decision trees developed from 1-minute training data were previously shown in Figures

```
hr_high180 <= 0 : 1 (474.0)
hr_high180 > 0 : 0 (153862.0)
```

Figure 4-24: Heart rate artifact detection decision tree model from 1-second data (180-300-600).

```
ox_med180 <= 0.5 : 1 (4725.0)
ox_med180 > 0.5 :
|   ox_avg180 <= 15.7 :
|   |   ox_med180 <= 15.9 : 0 (150362.0/57.0)
|   |   ox_med180 > 15.9 :
|   |   |   ox_std_dev180 <= 8.67 : 0 (638.0/61.0)
|   |   |   ox_std_dev180 > 8.67 :
|   |   |   |   hr_avg600 <= 161.8 : 0 (100.0/6.0)
|   |   |   |   hr_avg600 > 161.8 :
|   |   |   |   |   co2_std_dev300 <= 2.2 : 1 (190.0/5.0)
|   |   |   |   |   co2_std_dev300 > 2.2 : 0 (80.0/17.0)
|   ox_avg180 > 15.7 :
|   |   ox_med180 <= 20 : 0 (601.0)
|   |   ox_med180 > 20 : 1 (514.0/1.0)
```

Figure 4-25: Oxygen artifact detection decision tree model from 1-second data (180-300-600).

Table 4.9: ROC curve areas for each model run on the test set of its own granularity, and for the 1-minute models run on two different types of 1-second test sets (Experiment 1 used 180, 300, and 600 values for feature derivation; Experiment 2 used 3, 5, and 10 values for feature derivation).

Signal	1-minute model run on 1-minute test set	1-second model run on 1-second test set (5-15-60)	1-minute model run on 1-second test set (Experiment 1)	1-minute model run on 1-second test set (Experiment 2)
Blood pressure	89.41%	100.00%	100.00%	100.00%
Carbon dioxide	93.29%	99.70%	99.25%	99.70%
Heart rate	92.83%	100.00%	96.30%	99.95%
Oxygen	99.93%	99.40%	98.81%	99.92%

```
bp_med3 <= 2 : 1 (45.0)
bp_med3 > 2 : 0 (60581.0)
```

Figure 4-26: Blood pressure artifact detection decision tree model using 1-second data (2-3-5) with front-labeling.

4-5 to 4-8. Table 4.9 displays the areas under the ROC curve for each model run on test set data of its same granularity, plus results of running the 1-minute models on the two different types of preprocessed 1-second test sets (180-300-600 and 3-5-10).

4.3.4 Single-Phase Models: Class Labeling Experiments

Location

Decision trees were created for both 5-15-60 and 2-3-5 second time intervals for each of the four signals using both front- and end-labeling. Models were quite small, usually consisting of a single attribute node, and they performed with almost no errors. No ROC curves were therefore generated and we simply report the number of true positives (TP), true negatives (TN), and, if applicable, false positives (FP) or false negatives (FN) from running each model on its respective test set.

Blood Pressure Artifacts Using 2-3-5 feature attributes, both the front-labeling and end-labeling BP artifact detection decision tree models had perfect results. The front-labeling model, shown in Figure 4-26, classified 7709 true negatives and four true positives. The end-labeling model, shown in Figure 4-27, classified 7707 true negatives and six true positives.

```
bp_high2 <= 2 : 1 (43.0)
bp_high2 > 2 : 0 (60583.0)
```

Figure 4-27: Blood pressure artifact detection decision tree model using 1-second data (2-3-5) with end-labeling.

```
bp_low5 <= 2 : 1 (34.0)
bp_low5 > 2 : 0 (59643.0)
```

Figure 4-28: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with front-labeling.

Results for the 5-15-60 trees were similar: the front-labeling model, shown in Figure 4-28, classified 7583 true negatives and three true positives, while the end-labeling model, shown in Figure 4-29, also classified 7583 true negatives and three true positives.

Carbon Dioxide Artifacts In both the 2-3-5 and 5-15-60 experiments, both the front-labeling and end-labeling CO_2 artifact detection decision tree models had perfect results. The front-labeling models, shown in Figures 4-30 and 4-31, classified 7712 and 7585 true negatives, respectively, and two and one true positives, respectively. The end-labeling models, shown in Figures 4-32 and 4-33, also correctly classified 7712 and 7585 true negatives and two and one true positives, respectively.

Heart Rate Artifacts Using 2-3-5 feature attributes, both the front-labeling and end-labeling HR artifact detection decision tree models had perfect results. The front-labeling model, shown in Figure 4-34, classified 7623 true negatives and 79 true positives. The end-labeling model, shown in Figure 4-35, classified 7628 true negatives and 74 true positives.

Results for the 5-15-60 HR trees were similar: the front-labeling model, shown in Figure 4-36,

```
bp_low5 <= 2 : 1 (34.0)
bp_low5 > 2 : 0 (59643.0)
```

Figure 4-29: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with end-labeling.

```
co2_med3 > 0.4 : 0 (60593.0)
co2_med3 <= 0.4 :
|   co2_high2 <= 0.3 : 1 (19.0)
|   co2_high2 > 0.3 : 0 (15.0)
```

Figure 4-30: Carbon dioxide artifact detection decision tree model using 1-second data (2-3-5) with front-labeling.

```
co2_high5 > 0.4 : 0 (59661.0)
co2_high5 <= 0.4 :
|   co2_high5 <= 0.3 : 1 (15.0)
|   co2_high5 > 0.3 : 0 (15.0)
```

Figure 4-31: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with front-labeling.

```
co2_low3 > 0.3 : 0 (60596.0)
co2_low3 <= 0.3 :
|   co2_avg2 <= 0.3 : 1 (19.0)
|   co2_avg2 > 0.3 : 0 (12.0)
```

Figure 4-32: Carbon dioxide artifact detection decision tree model using 1-second data (2-3-5) with end-labeling.

```
co2_low5 > 0.3 : 0 (59665.0)
co2_low5 <= 0.3 :
|   co2_range5 <= 0 : 1 (18.0)
|   co2_range5 > 0 : 0 (8.0)
```

Figure 4-33: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with end-labeling.

```
hr_high2 <= 0 : 1 (633.0)
hr_high2 > 0 : 0 (59913.0)
```

Figure 4-34: Heart rate artifact detection decision tree model using 1-second data (2-3-5) with front-labeling.

```
hr_high2 <= 0 : 1 (631.0)
hr_high2 > 0 : 0 (59915.0)
```

Figure 4-35: Heart rate artifact detection decision tree model using 1-second data (2-3-5) with end-labeling.

```
hr_high5 <= 0 : 1 (521.0)
hr_high5 > 0 : 0 (58870.0)
```

Figure 4-36: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with front-labeling.

classified 7505 true negatives and 54 true positives, while the end-labeling model, shown in Figure 4-37, classified 7499 true negatives and 60 true positives.

Oxygen Artifacts The models developed for detection of oxygen artifacts were slightly larger than those developed for the other signal artifacts. Using 2-3-5 feature attributes, both the front-labeling and end-labeling oxygen artifact detection decision tree models again had perfect results. The front-labeling model, shown in Figure 4-38, classified 7693 true negatives and 20 true positives. The end-labeling model, shown in Figure 4-39, also classified 7693 true negatives and 20 true positives.

Results for the 5-15-60 oxygen trees were quite good but one of the models had a false negative. The front-labeling model, shown in Figure 4-40, classified 7565 true negatives, 20 true positives, and one false negative. The end-labeling model, shown in Figure 4-41, classified 7562 true negatives and 24 true positives.

```
hr_high5 <= 0 : 1 (528.0)
hr_high5 > 0 : 0 (58863.0)
```

Figure 4-37: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with end-labeling.

```
ox_med3 > 20 : 1 (139.0)
ox_med3 <= 20 :
|   ox_med3 > 2.1 : 0 (60461.0)
|   ox_med3 <= 2.1 :
|   |   ox_high2 <= 0 : 1 (11.0)
|   |   ox_high2 > 0 : 0 (15.0)
```

Figure 4-38: Oxygen artifact detection decision tree model using 1-second data (2-3-5) with front-labeling.

```
ox_med3 > 20 : 1 (143.0)
ox_med3 <= 20 :
|   ox_med3 > 2.1 : 0 (60456.0)
|   ox_med3 <= 2.1 :
|   |   ox_high2 <= 0 : 1 (13.0)
|   |   ox_high2 > 0 : 0 (14.0)
```

Figure 4-39: Oxygen artifact detection decision tree model using 1-second data (2-3-5) with end-labeling.

```
ox_med5 > 19.9 : 1 (132.0)
ox_med5 <= 19.9 :
|   co2_abs_slope15 <= 0.15 : 0 (59520.0)
|   co2_abs_slope15 > 0.15 :
|   |   ox_low5 <= 5 : 1 (8.0)
|   |   ox_low5 > 5 : 0 (17.0)
```

Figure 4-40: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with front-labeling.

```
ox_med5 > 19.9 : 1 (128.0)
ox_med5 <= 19.9 :
|   ox_avg5 > 2.1 : 0 (59524.0)
|   ox_avg5 <= 2.1 :
|   |   ox_high5 <= 0 : 1 (10.0)
|   |   ox_high5 > 0 : 0 (15.0)
```

Figure 4-41: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with end-labeling.

```
bp_low5 <= 2 : 1 (34.0)
bp_low5 > 2 : 0 (59643.0)
```

Figure 4-42: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 1.

```
bp_low5 <= 2 : 1 (36.0)
bp_low5 > 2 : 0 (59646.0)
```

Figure 4-43: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 2.

Strictness

For each signal artifact type, four models are presented, corresponding to strictness labeling Methods 1 through 4, as described in Section 4.2.3. Comparisons are made by looking at numbers of errors since most models were very small and had perfect performance on their test sets.

Blood Pressure Artifacts All four BP models were able to correctly classify negative and positive examples of artifacts. Figures 4-42 through 4-45 show the decision tree models for detecting BP artifacts for Methods 1 through 4, respectively. Table 4.10 shows the results of each model run on its own test set, as well as the most strict model run on increasingly less strict test sets.

Carbon Dioxide Artifacts Three of the four CO_2 models were able to correctly classify negative and positive examples of artifacts; the least strict method (Method 4) had one error. Figures 4-46 through 4-49 show the decision tree models for detecting CO_2 artifacts for Methods 1 through 4, respectively. Table 4.11 shows the results of each model run on its own test set, as well as the most

```
bp_med5 <= 3 : 1 (55.0)
bp_med5 > 3 : 0 (59694.0)
```

Figure 4-44: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 3.

```
bp_low5 <= 4 : 1 (119.0)
bp_low5 > 4 : 0 (59630.0)
```

Figure 4-45: Blood pressure artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 4.

Table 4.10: Performance of 1-second NICU data 5-15-60 blood pressure models using different class labeling strictness methods.

		Method 1 model	Method 2 model	Method 3 model	Method 4 model
Method 1 test set	True Neg	7583			
	True Pos	3			
	False Neg				
	False Pos				
Method 2 test set	True Neg	7582	7582		
	True Pos	4	4		
	False Neg				
	False Pos				
Method 3 test set	True Neg	7582		7591	
	True Pos	5		5	
	False Neg				
	False Pos	9			
Method 4 test set	True Neg	7582			7582
	True Pos	14			14
	False Neg				
	False Pos				

```
co2_high5 > 0.4 : 0 (59661.0)
co2_high5 <= 0.4 :
|   co2_high5 <= 0.3 : 1 (15.0)
|   co2_high5 > 0.3 : 0 (15.0)
```

Figure 4-46: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 1.

```
co2_low5 > 0.3 : 0 (59668.0)
co2_low5 <= 0.3 :
|   co2_med5 <= 0.3 : 1 (18.0)
|   co2_med5 > 0.3 : 0 (9.0)
```

Figure 4-47: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 2.

strict model run on increasingly less strict test sets.

Heart Rate Artifacts All four HR models were able to correctly classify negative and positive examples of artifacts in their own strictness category. Figures 4-50 through 4-53 show the decision tree models for detecting HR artifacts for Methods 1 through 4, respectively. Table 4.12 shows the results of each model run on its own test set, as well as the most strict model run on increasingly less strict test sets.

Oxygen Artifacts The oxygen models had mixed results. Figures 4-54 through 4-57 show the decision tree models for detecting oxygen artifacts for Methods 1 through 4, respectively. Table 4.13 shows the results of each model run on its own test set, as well as the most strict model run on increasingly less strict test sets.

```
co2_med5 <= 0.3 : 1 (26.0)
co2_med5 > 0.3 : 0 (59723.0)
```

Figure 4-48: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 3.

```
ox_low5 <= 0 : 1 (67.0)
ox_low5 > 0 :
|   co2_high5 > 0.4 : 0 (59654.0)
|   co2_high5 <= 0.4 :
|   |   ox_std_dev15 <= 1.12 : 0 (15.0)
|   |   ox_std_dev15 > 1.12 : 1 (13.0)
```

Figure 4-49: Carbon dioxide artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 4.

Table 4.11: Performance of 1-second NICU data 5-15-60 carbon dioxide models using different class labeling strictness methods.

		Method 1 model	Method 2 model	Method 3 model	Method 4 model
Method 1	True Neg	7585			
test set	True Pos	1			
	False Neg				
	False Pos				
Method 2	True Neg	7585	7585		
test set	True Pos		1		
	False Neg	1			
	False Pos				
Method 3	True Neg	7593		7593	
test set	True Pos	1		3	
	False Neg	2			
	False Pos				
Method 4	True Neg	7585			7584
test set	True Pos	1			11
	False Neg	10			
	False Pos				1

```
hr_high5 <= 0 : 1 (521.0)
hr_high5 > 0 : 0 (58870.0)
```

Figure 4-50: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 1.

```
hr_med5 <= 0 : 1 (590.0)
hr_med5 > 0 : 0 (58865.0)
```

Figure 4-51: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 2.

```
hr_med5 <= 0 : 1 (668.0)
hr_med5 > 0 : 0 (59081.0)
```

Figure 4-52: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 3.

```
hr_low5 <= 0 : 1 (885.0)
hr_low5 > 0 : 0 (58864.0)
```

Figure 4-53: Heart rate artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 4.

Table 4.12: Performance of 1-second NICU data 5-15-60 heart rate models using different class labeling strictness methods.

		Method 1	Method 2	Method 3	Method 4
		model	model	model	model
Method 1	True Neg	7505			
test set	True Pos	54			
	False Neg				
	False Pos				
Method 2	True Neg	7505	7505		
test set	True Pos	52	59		
	False Neg	7			
	False Pos				
Method 3	True Neg	7506		7506	
test set	True Pos	77		90	
	False Neg	13			
	False Pos				
Method 4	True Neg	7482			7482
test set	True Pos	77			114
	False Neg	37			
	False Pos				

```
ox_med5 > 19.9 : 1 (132.0)
ox_med5 <= 19.9 :
|   co2_abs_slope15 <= 0.15 : 0 (59520.0)
|   co2_abs_slope15 > 0.15 :
|   |   ox_low5 <= 5 : 1 (8.0)
|   |   ox_low5 > 5 : 0 (17.0)
```

Figure 4-54: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 1.

```
ox_med5 > 19.9 : 1 (138.0)
ox_med5 <= 19.9 :
|   co2_low5 > 0.4 : 0 (59517.0)
|   co2_low5 <= 0.4 :
|   |   ox_low60 <= 4 : 1 (9.0)
|   |   ox_low60 > 4 : 0 (18.0)
```

Figure 4-55: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 2.

```
ox_med5 > 20 : 1 (151.0)
ox_med5 <= 20 :
|   co2_med5 > 0.4 : 0 (59570.0)
|   co2_med5 <= 0.4 :
|   |   bp_low5 <= 24 : 1 (17.0)
|   |   bp_low5 > 24 : 0 (11.0)
```

Figure 4-56: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 3.

```
ox_med5 <= 19.7 :
|   ox_low5 <= 0 : 1 (67.0)
|   ox_low5 > 0 :
|   |   ox_avg5 <= 19 : 0 (59496.0)
|   |   ox_avg5 > 19 :
|   |   |   ox_slope15 <= -0.46 : 1 (2.0)
|   |   |   ox_slope15 > -0.46 :
|   |   |   |   bp_slope15 <= 0.01 : 0 (20.0)
|   |   |   |   bp_slope15 > 0.01 : 1 (3.0/1.0)
ox_med5 > 19.7 :
|   co2_med60 <= 1.2 : 1 (152.0)
|   co2_med60 > 1.2 :
|   |   bp_range5 <= 0 : 0 (2.0)
|   |   bp_range5 > 0 : 1 (7.0/1.0)
```

Figure 4-57: Oxygen artifact detection decision tree model using 1-second data (5-15-60) with class labeling strictness Method 4.

Table 4.13: Performance of 1-second NICU data 5-15-60 oxygen models using different class labeling strictness methods.

		Method 1	Method 2	Method 3	Method 4
		model	model	model	model
Method 1	True Neg	7565			
test set	True Pos	20			
	False Neg	1			
	False Pos				
Method 2	True Neg	7566	7566		
test set	True Pos	19	20		
	False Neg	1			
	False Pos				
Method 3	True Neg	7575		7577	
test set	True Pos	19		19	
	False Neg				
	False Pos	2			
Method 4	True Neg	7569			7567
test set	True Pos	21			27
	False Neg	6			
	False Pos				2

4.3.5 Multi-Phase Models

Neural Networks

The multi-phase neural network model developed to detect blood pressure artifacts had 24 input nodes, no hidden nodes, and 2 output nodes. The area under its ROC curve, shown in Figure 4-58, was 94.42%. For carbon dioxide artifacts, the multi-phase neural network model, consisting of 24 input nodes, no hidden nodes, and 2 output nodes, achieved an area under the ROC curve of 91.93%; its ROC curve is shown in Figure 4-59. The multi-phase neural network model developed to detect heart rate artifacts had 24 inputs, 20 hidden nodes in one layer, and 2 output nodes. It achieved an area under the ROC curve of 97.50%; Figure 4-60 shows its ROC curve. Finally, for the multi-phase oxygen artifact detection model, the neural network had 24 inputs, 20 hidden nodes in a single layer, and 2 outputs. Its ROC curve, shown in Figure 4-61, had an area of 99.84%. These values are summarized in Table 4.14.

Decision Trees

LNKnet decision trees were given the same training data with the same numbers of feature attributes (24) for each case as had been given to the neural networks. The multi-phase blood pressure artifact detection decision tree was allowed to grow fully (i.e., continue expanding until no misclassifications existed), then was pruned to use only 10 nodes. It achieved an area under the ROC curve of 94.30%, shown in Figure 4-62. For carbon dioxide, the tree was fully grown and pruned to use 20 nodes, with a resulting ROC curve area of 98.00%; Figure 4-63 shows its ROC curve. For heart rate, the tree was also fully grown, then was pruned to use only 30 nodes. Its ROC curve, shown in Figure 4-64, had an area of 97.82%. For multi-phase oxygen artifact detection, the decision tree was fully grown and then pruned to use 40 nodes. The resulting area under the ROC curve, shown in Figure 4-65 was 92.62%. These results are tabulated with the neural network results in Table 4.14. Figure 4-66 illustrates the performance of both multi-phase decision trees and multi-phase neural networks on detection of artifacts of each signal type.

The multi-phase blood pressure decision tree built with c4.5 that was smallest in size (seven nodes) while still performing well is shown in Figure 4-67. Similarly, the CO_2 model built with c4.5 is shown in Figure 4-68; it also had seven nodes. The HR decision tree had 51 nodes and is shown in Figure 4-69. Further attempts at tree size reduction for the HR model resulted in large numbers of errors. For oxygen, the c4.5 decision tree that still performed reasonably had 22 nodes; it is shown

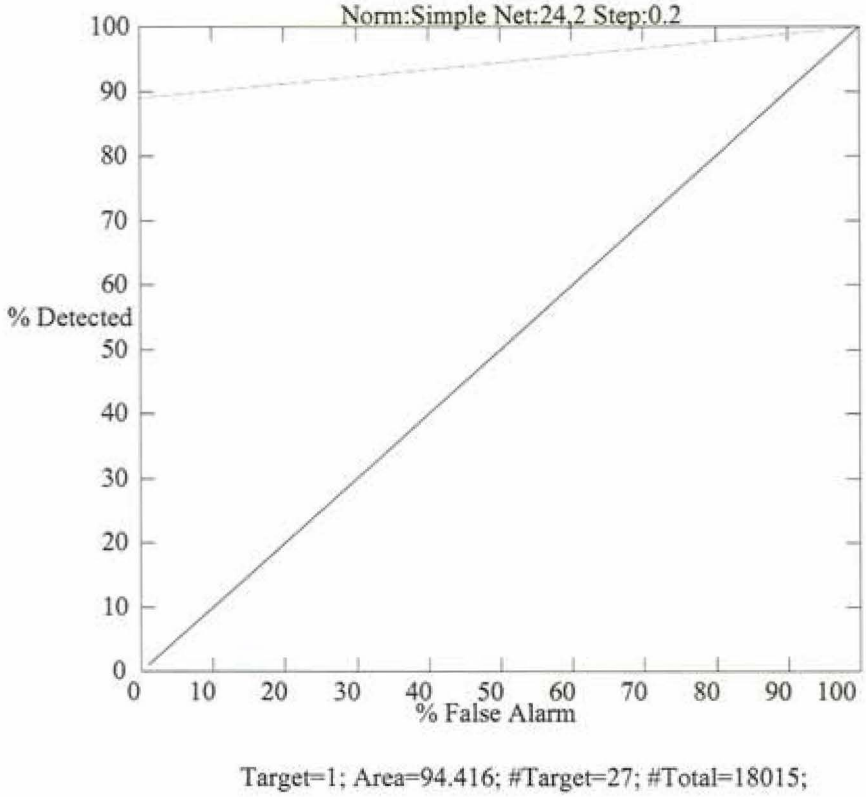
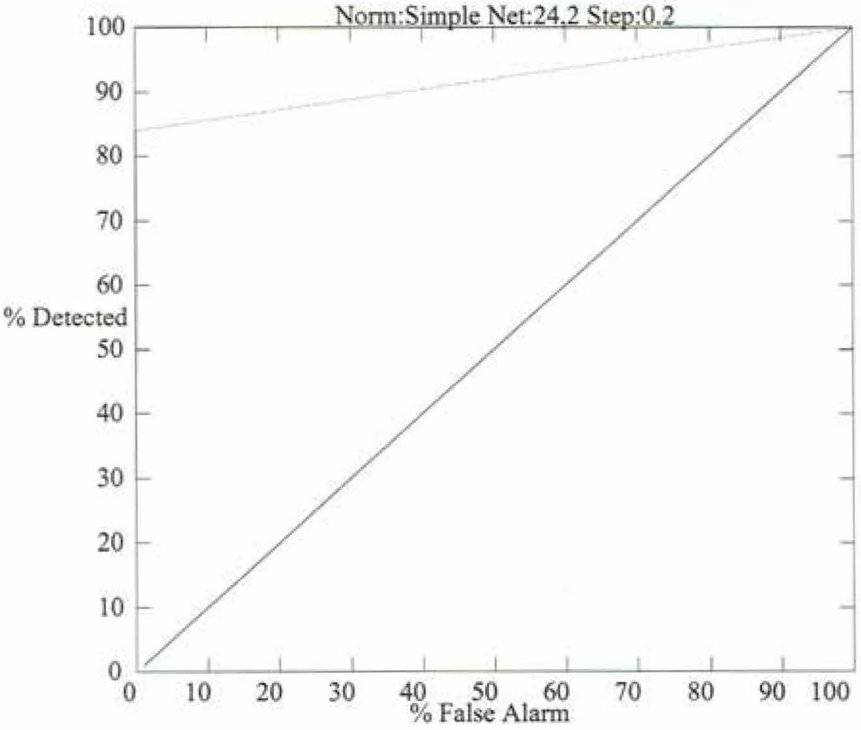


Figure 4-58: ROC curve for blood pressure artifact neural network multi-phase model.

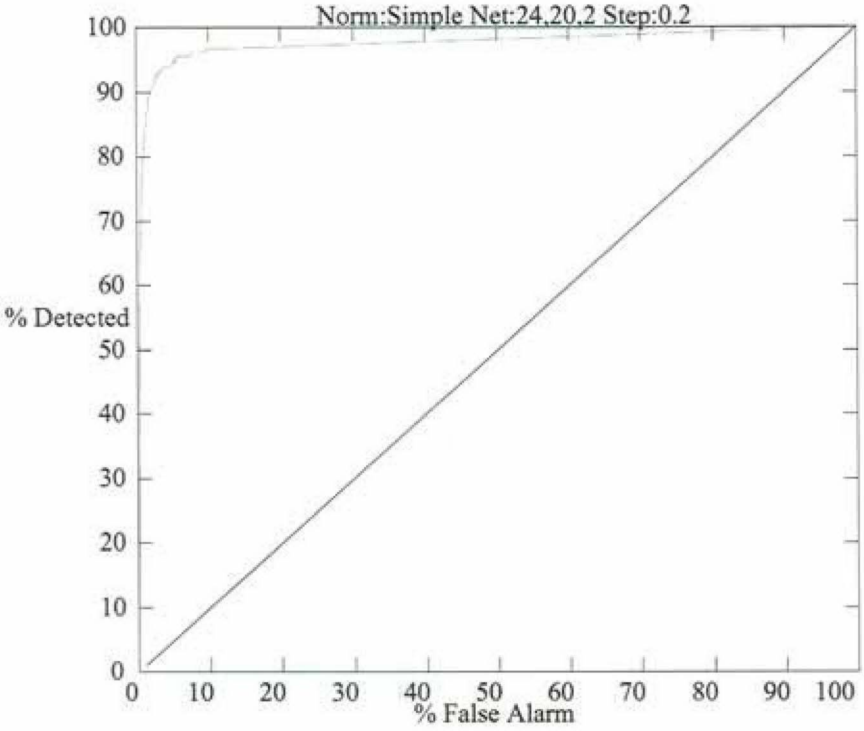
Table 4.14: ROC curve areas for multi-phase neural network and decision tree models.

Signal	neural network ROC area	decision tree ROC area
Blood pressure	94.42%	94.30%
Carbon dioxide	91.93%	98.00%
Heart rate	97.50%	97.82%
Oxygen	99.84%	92.62%



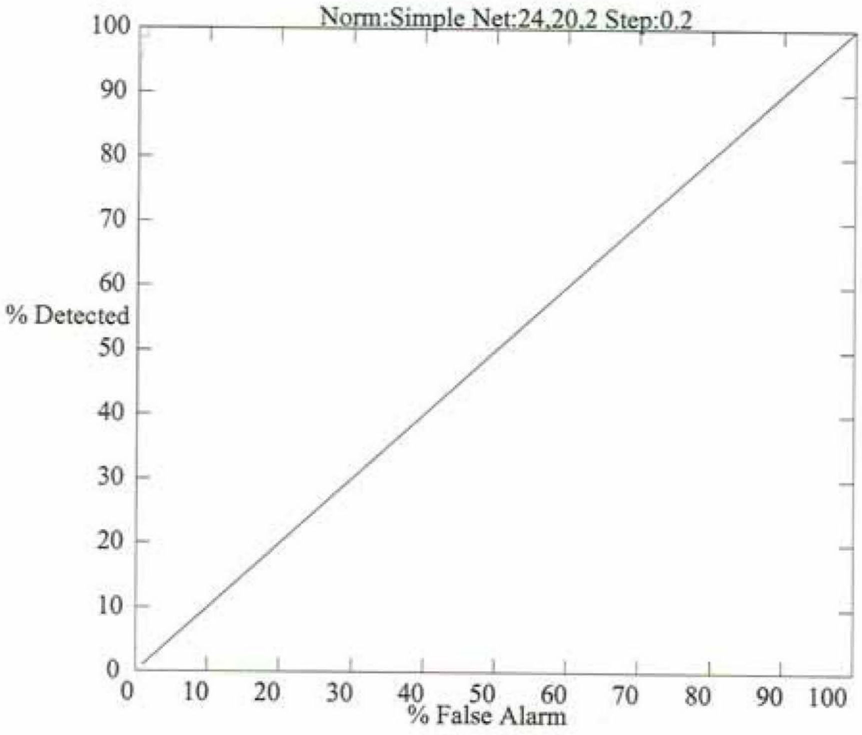
Target=1; Area=91.929; #Target=25; #Total=18015;

Figure 4-59: ROC curve for carbon dioxide artifact neural network multi-phase model.



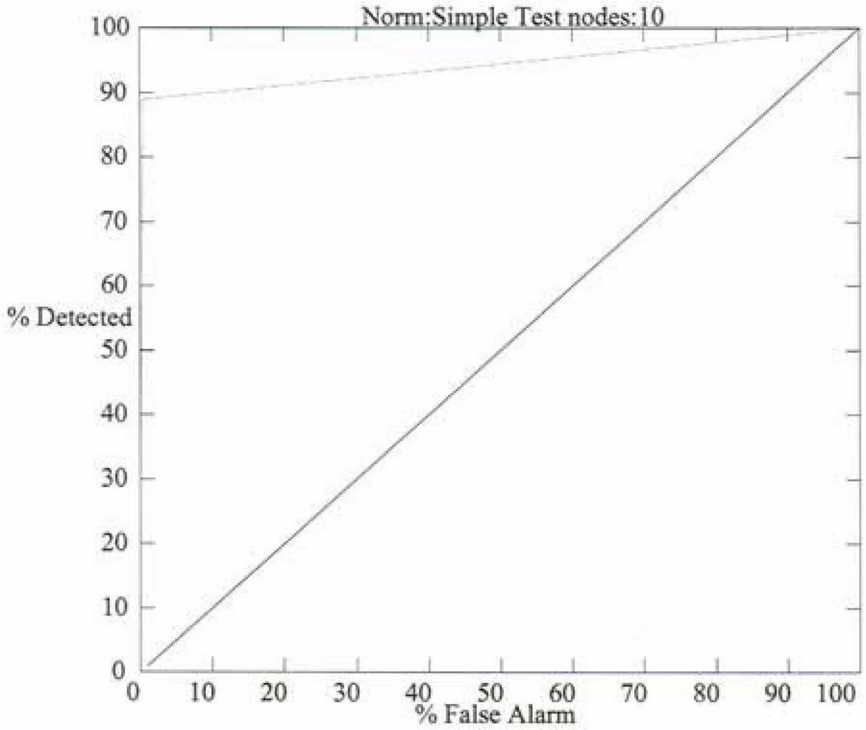
Target=1; Area=97.496; #Target=244; #Total=18015;

Figure 4-60: ROC curve for heart rate artifact neural network multi-phase model.



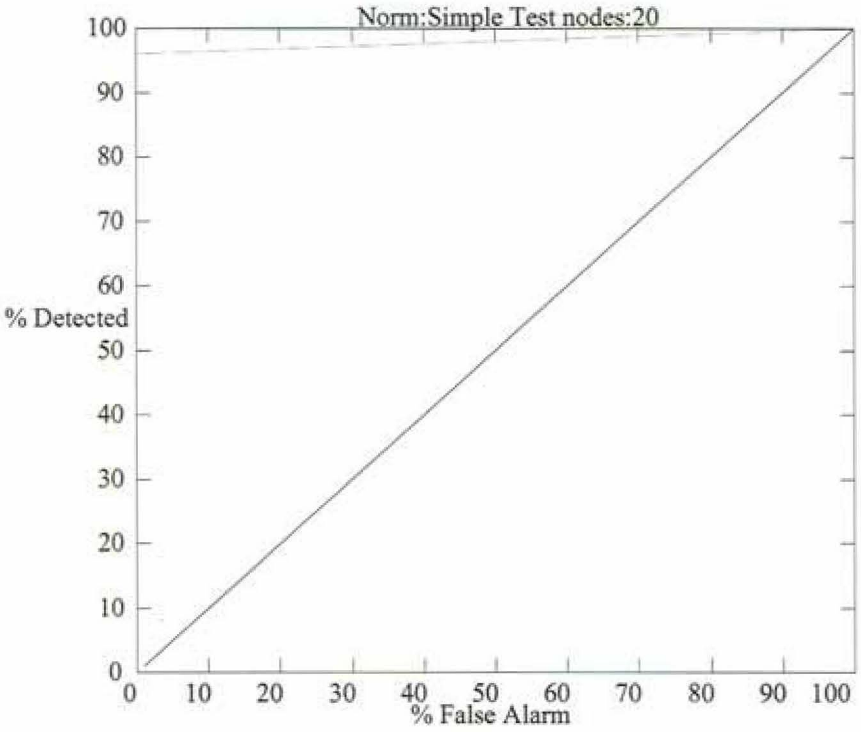
Target=1; Area=99.840; #Target=68; #Total=18015;

Figure 4-61: ROC curve for oxygen artifact neural network multi-phase model.



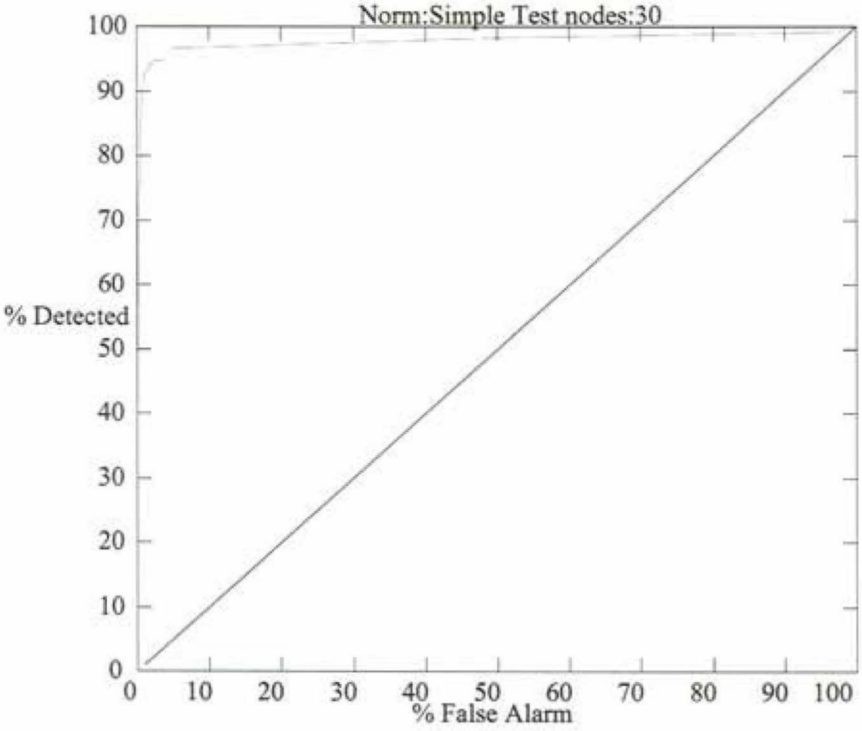
Target=1; Area=94.297; #Target=27; #Total=18015;

Figure 4-62: ROC curve for blood pressure artifact decision tree multi-phase model.



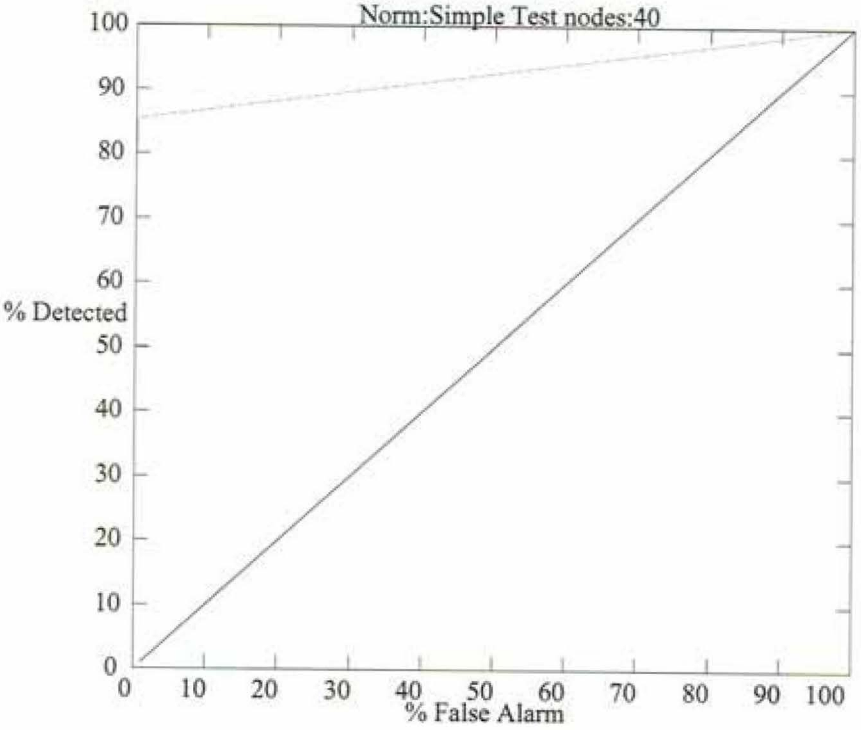
Target=1; Area=97.997; #Target=25; #Total=18015;

Figure 4-63: ROC curve for carbon dioxide artifact decision tree multi-phase model.



Target=1; Area=97.817; #Target=244; #Total=18015;

Figure 4-64: ROC curve for heart rate artifact decision tree multi-phase model.



Target=1; Area=92.617; #Target=68; #Total=18015;

Figure 4-65: ROC curve for oxygen artifact decision tree multi-phase model.

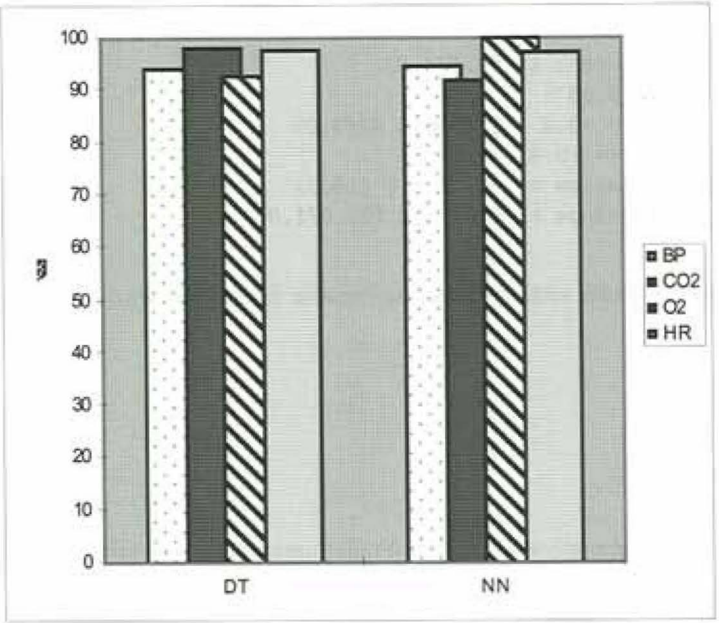


Figure 4-66: Performance of multi-phase decision trees and multi-phase neural networks on detection of artifacts for each signal type. BP: gray dots on white; CO2: solid dark gray; O2: black stripes on white; HR: solid light gray.

```
bp_5_phase2_slope > 3.3 : 1 (32.0/5.0)
bp_5_phase2_slope <= 3.3 :
|   bp_30_phase2_slope <= 0.95 : 0 (59649.0/13.0)
|   bp_30_phase2_slope > 0.95 :
|   |   bp_30_phase1_slope <= -0.35 : 1 (40.0)
|   |   bp_30_phase1_slope > -0.35 : 0 (28.0/8.0)
```

Figure 4-67: Blood pressure artifact detection decision tree model with multi-phase attributes.

```
co2_5_phase2_slope > 0.08 : 1 (25.0/4.0)
co2_5_phase2_slope <= 0.08 :
|   ox_3_phase2_slope > -0.4 : 0 (59694.0/12.0)
|   ox_3_phase2_slope <= -0.4 :
|   |   co2_30_phase1_slope <= -0.01 : 0 (10.0)
|   |   co2_30_phase1_slope > -0.01 : 1 (20.0/1.0)
```

Figure 4-68: Carbon dioxide artifact detection decision tree model with multi-phase attributes.

in Figure 4-70.

4.4 Discussion

The results have shown that applying the TrendFinder paradigm to the task of detecting artifacts on signals from a neonatal intensive care unit may be a very useful way of integrating multiple signals for the purpose of detecting artifacts in one of those signals. Furthermore, these results indicate that pre-calculation of a set of derived values from raw streams of signal data may be a valid method for better interpretation of these data. We conclude this chapter with a discussion of several issues: comparison of different classifiers, data visualization, data granularity, class labeling, multi-phase learning, and study limitations.

4.4.1 Comparison of Classifiers

We have evaluated the performance of five different classifiers–neural networks, decision trees, logistic regression, radial basis function networks, and support vector machines. These results are tabulated in Table 4.15 for ease in comparison. Figure 4-71 illustrates the performance of five different types

```
hr_3_phase2_slope <= -61.5 : 1 (83.0/3.8)
hr_3_phase2_slope > -61.5 :
|   hr_3_phase2_slope > 32.5 : 1 (98.0/16.0)
|   hr_3_phase2_slope <= 32.5 :
|   |   hr_5_phase1_slope <= -26.2 : 1 (120.0/26.7)
|   |   hr_5_phase1_slope > -26.2 :
|   |   |   hr_30_phase2_slope <= 1.06 :
|   |   |   |   hr_30_phase1_slope <= -4.05 :
|   |   |   |   |   hr_5_phase1_slope <= 0 : 1 (68.0/15.9)
|   |   |   |   |   hr_5_phase1_slope > 0 : 0 (48.0/8.3)
|   |   |   |   hr_30_phase1_slope > -4.05 :
|   |   |   |   |   bp_30_phase2_slope <= 0.14 :
|   |   |   |   |   |   hr_5_biphase_pattern = 1: 0 (9737.0/8.5)
|   |   |   |   |   |   hr_5_biphase_pattern = 2: 0 (2649.0/3.9)
|   |   |   |   |   |   hr_5_biphase_pattern = 3: 0 (13007.0/1.4)
|   |   |   |   |   |   hr_5_biphase_pattern = 4: 0 (2996.0/5.1)
|   |   |   |   |   |   hr_5_biphase_pattern = 6: 0 (2678.0/1.4)
|   |   |   |   |   |   hr_5_biphase_pattern = 7: 0 (12622.0/3.9)
|   |   |   |   |   |   hr_5_biphase_pattern = 8: 0 (3030.0/9.6)
|   |   |   |   |   |   hr_5_biphase_pattern = 9: 0 (9601.0/1.4)
|   |   |   |   |   |   hr_5_biphase_pattern = 5:
|   |   |   |   |   |   |   bp_30_phase2_slope <= -0.05 :
|   |   |   |   |   |   |   |   hr_30_phase2_slope <= -0.01 : 0 (101.0/15.0)
|   |   |   |   |   |   |   |   hr_30_phase2_slope > -0.01 :
|   |   |   |   |   |   |   |   |   hr_30_phase2_slope <= 0 : 1 (84.0/6.2)
|   |   |   |   |   |   |   |   |   hr_30_phase2_slope > 0 : 0 (65.0/12.7)
|   |   |   |   |   |   |   bp_30_phase2_slope > -0.05 :
|   |   |   |   |   |   |   |   bp_30_phase1_slope > -0.04 : 0 (739.0/7.4)
|   |   |   |   |   |   |   |   bp_30_phase1_slope <= -0.04 :
|   |   |   |   |   |   |   |   |   hr_30_phase1_slope <= -0.01 : 0 (107.0/1.4)
|   |   |   |   |   |   |   |   |   hr_30_phase1_slope > -0.01 :
|   |   |   |   |   |   |   |   |   |   hr_30_phase1_slope <= 0.01 : 1 (94.0/23.4)
|   |   |   |   |   |   |   |   |   |   hr_30_phase1_slope > 0.01 : 0 (50.0/1.4)
|   |   |   |   |   bp_30_phase2_slope > 0.14 :
|   |   |   |   |   |   bp_30_phase1_slope <= 0.1 : 0 (941.0/36.6)
|   |   |   |   |   |   bp_30_phase1_slope > 0.1 :
|   |   |   |   |   |   |   hr_5_phase2_slope <= -0.1 : 0 (46.0/1.4)
|   |   |   |   |   |   |   hr_5_phase2_slope > -0.1 :
|   |   |   |   |   |   |   |   hr_5_phase2_slope <= 0 : 1 (46.0/11.5)
|   |   |   |   |   |   |   |   hr_5_phase2_slope > 0 : 0 (44.0/6.1)
|   |   |   hr_30_phase2_slope > 1.06 :
|   |   |   |   hr_5_biphase_pattern = 1: 0 (66.0/8.4)
|   |   |   |   hr_5_biphase_pattern = 2: 1 (21.0/11.1)
|   |   |   |   hr_5_biphase_pattern = 3: 0 (102.0/1.4)
|   |   |   |   hr_5_biphase_pattern = 4: 1 (26.0/4.9)
|   |   |   |   hr_5_biphase_pattern = 5: 1 (133.0/2.6)
|   |   |   |   hr_5_biphase_pattern = 6: 0 (14.0/1.3)
|   |   |   |   hr_5_biphase_pattern = 7: 0 (138.0/6.2)
|   |   |   |   hr_5_biphase_pattern = 8: 1 (19.0/4.8)
|   |   |   |   hr_5_biphase_pattern = 9: 0 (176.0/2.6)
```

Figure 4-69: Heart rate artifact detection decision tree model with multi-phase attributes.

```
bp_5_phase2_slope > 3.3 : 1 (32.0/13.4)
bp_5_phase2_slope <= 3.3 :
|   co2_30_phase1_slope <= -0.02 :
|   |   ox_30_phase1_slope > 0.35 : 1 (39.0/1.4)
|   |   ox_30_phase1_slope <= 0.35 :
|   |   |   hr_30_phase2_slope <= -0.23 : 0 (46.0/1.4)
|   |   |   hr_30_phase2_slope > -0.23 :
|   |   |   |   bp_30_phase2_slope <= -0.06 : 1 (35.0/4.9)
|   |   |   |   bp_30_phase2_slope > -0.06 :
|   |   |   |   |   ox_30_biphase_pattern = 1: 0 (4.0/1.2)
|   |   |   |   |   ox_30_biphase_pattern = 2: 1 (12.0/4.7)
|   |   |   |   |   ox_30_biphase_pattern = 3: 0 (32.0/4.9)
|   |   |   |   |   ox_30_biphase_pattern = 4: 1 (1.0/0.8)
|   |   |   |   |   ox_30_biphase_pattern = 5: 0 (88.0/27.5)
|   |   |   |   |   ox_30_biphase_pattern = 6: 1 (21.0/5.9)
|   |   |   |   |   ox_30_biphase_pattern = 7: 1 (6.0/1.2)
|   |   |   |   |   ox_30_biphase_pattern = 8: 0 (18.0/1.3)
|   |   |   |   |   ox_30_biphase_pattern = 9: 0 (0.0)
|   co2_30_phase1_slope > -0.02 :
|   |   co2_3_phase2_slope <= -0.15 : 1 (32.0/17.5)
|   |   co2_3_phase2_slope > -0.15 : 0 (59383.0/38.8)
```

Figure 4-70: Oxygen artifact detection decision tree model with multi-phase attributes.

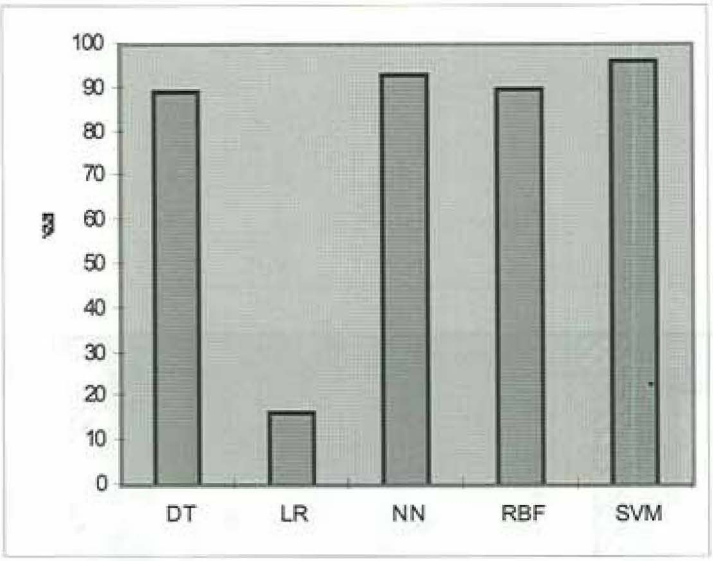


Figure 4-71: Performance of five different classification models on blood pressure artifact detection (shown as ROC curve areas). DT = decision tree, LR = logistic regression, NN = neural network, RBF = radial basis function network, SVM = support vector machine.

of classifiers on blood pressure artifact detection. Figure 4-72 illustrates the performance trends of decision trees, logistic regression, and neural networks for detection of each of the four types of signal artifact. What we have found is that the choice of linear or non-linear classification system can make a tremendous difference. Non-linear classifiers (i.e., neural networks, decision trees, RBF networks, and SVMs) perform quite well. While the differences between these non-linear methods are statistically significant (e.g., $p < 0.01$ for the 3.9% difference in ROC curve areas between the decision tree and neural network models for detecting BP artifacts), they pale in comparison to the differences between the LR models and the other models. For example, the difference in performance between the BP artifact decision tree model and LR model was significant beyond the $p = 0.0000001$ level.

For NICU monitor data such as ours, we found that the nature of the data space is such that linear techniques are grossly inadequate. Logistic regression, a linear classifier as used (without interaction terms), performed very poorly in this domain. This is likely due to the discontinuous nature of artifacts on the spectrums defined by the attributes used. This seems probable especially in light

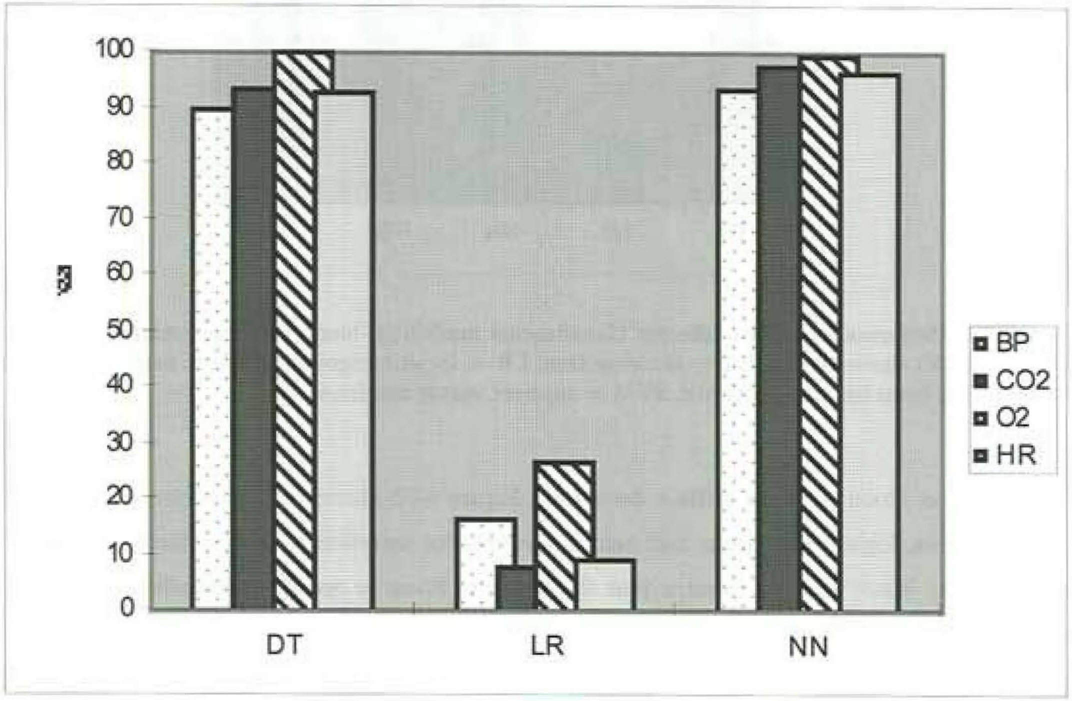


Figure 4-72: Performance of decision trees (DT), logistic regression (LR), and neural networks (NN) on detection of each of the four signal artifact types. BP: gray dots on white; CO2: solid dark gray; O2: black stripes on white; HR: solid light gray.

Table 4.15: Comparison of five classifiers for detection of NICU artifacts.

Signal	Classifier	ROC curve area
Blood pressure	MLP	93.27%
	Decision tree	89.4%
	LR	16.7%
	RBF	90.00%
	SVM	95.95%
Carbon dioxide	MLP	97.46%
	Decision tree	93.3%
	LR	7.8%
Heart rate	MLP	96.62%
	Decision tree	92.8%
	LR	8.9%
Oxygen	MLP	99.31%
	Decision tree	99.9%
	LR	26.8%
	LR with dichotomization	86.0%

of the large performance improvement (ROC curve area increase from 26.8% to 86.0%) observed by dichotomizing one attribute (ox_med3) that was known (by observation of the decision tree model) to have a discontinuous influence on O_2 artifact detection. While LR methods could likely perform well given appropriately hand-crafted new variables and interaction terms, it is impossible to know a priori how to craft these variables correctly. It makes better sense, therefore, to use methods such as neural networks or decision trees which can automatically determine appropriate ways for discriminating data such as ours.

Regarding the specific decision tree models developed for detection of artifact in each of the signal types, it is interesting to note that three of the models (decision tree models for BP, CO_2, and HR) indeed made use of data from more than one signal type in order to decide upon the artifact status for the one signal type in question. Two of these models (decision tree models for BP and HR) also included attributes from all three available time intervals (three-, five-, and ten-minute interval-derived values). The O_2 decision tree model was small enough that a clinician might be able to evaluate it mentally. The other models, because of size or inclusion of calculations such as standard deviation, would be less practical for a clinician to evaluate mentally, but would, for example, lend themselves easily to being part of a computerized monitoring system.

It is particularly interesting to note that the decision tree model with the best performance of the four was the decision tree model for detecting O_2 artifacts. The especially interesting finding here is that this model, unlike the others, consisted exclusively of attributes derived from O_2 monitor data. While the original concept had been that multiple signal integration would provide a key towards effective artifact detection, this particular model does well without multiple signal information. This finding begs the question of how to better understand interrelationships between monitored signals. It may just be that artifacts in the oxygen signal are somehow more predictable, as we see that the neural network model is also able to achieve the same ROC curve area. Chapter 5, Section 5.3.1, presents a more thorough comparison and analysis of single-signal versus multi-signal models.

While neural networks and SVMs had better performance than decision trees for finding artifacts on the blood pressure signal ($p < 0.01$), they also required more computation time. The RBF classifier performed almost equally with the decision tree but also required more computation time. This is because the nature of this data, as Quinlan puts it, is more 'S-type', where 'S' stands for sequential [164]. (The next section gives a better sense for how our data 'look.') Essentially, this means that not every single one of the 96 attributes is important for classification of a case as artifact or not. Quinlan argues that S-type data do better, in terms of performance and/or computational requirements, with decision trees, while 'P-type,' or parallel, data (i.e., data in which all attributes are important for classification) do better with neural network-type classifiers. This is because neural networks (and SVMs and RBFs) use all of the attributes for each calculation, even when some of the input feature attributes may not be relevant. Looking at all of the data slows computation, while looking at irrelevant data is not helpful with tuning network node weights appropriately. S-type data are data that have many irrelevant feature attributes, while P-type data are data with mostly all relevant feature attributes. Decision trees, he further argues, would not be ideal for P-type data because in order to look at all attributes (since all of them are relevant in P-type data), the tree will need to be enormous.

We see that although Quinlan was correct about the computational aspect of neural network-type classifiers, performance in this case was not compromised with our S-type neonatal ICU monitor data. In fact, performance was better than that achieved with the decision trees. This may be due to the greedy nature of decision tree development, in which the selection of feature attributes at each node is chosen to be the best 'for the moment,' rather than some sense of 'global best.' This may have resulted in a final decision tree that did not include one or several globally informative attributes. Such a scenario could account for the decision tree's lower performance than the neural network-type classifiers, which do use all of the attributes.

An optimal approach to model-building for these type of data, in terms of both model performance and computational requirements, might be to build support vector machine models using computationally more efficient algorithms [153]. This would mean that traditional computational disadvantages may no longer be a factor, while model performance for our domain would be better.

4.4.2 Data Visualization

In Figures 4-73 to 4-75, we give a sense for how the preprocessed neonatal ICU data look in two-dimensional space, where each of the two axes per graph is chosen from those attributes present in the blood pressure artifact detection decision tree (1-minute data). Note that the plots in Figures 4-73 and 4-74 also show decision region boundaries; the lower region on each of these is the 'no_alarm' region. Note that in Figures 4-76 and 4-77, a group of artifacts (poorly labeled as 'false_alarm' in the plots) clearly exist on the lower-valued end of the x-axis (values plotted are normalized bp_med3 values). This corresponds nicely with the first line of the decision tree that was shown in Figure 4-5.

In Figures 4-78 to 4-80, we similarly display scatter plots and decision regions for carbon dioxide artifact detection. Again, attributes chosen for the plotted axes are those in the corresponding decision tree model shown in Figure 4-6. Note in Figure 4-78 that oxygen values are more discrete than values of the other three monitored signals. Note in Figures 4-79 and 4-80 how narrow a region of values are assumed by most calculated carbon dioxide ranges and slopes.

In Figures 4-81 to 4-86, we similarly display scatter plots and decision regions for heart rate artifact detection. Again, attributes chosen for the plotted axes are those in the corresponding decision tree model shown in Figure 4-7. We can see from Figures 4-81, 4-83, and 4-85 that, clearly, as the calculated heart rate range value increases, so too does the number of artifacts (poorly labeled as 'false_alarm' in the figures). Figure 4-82 shows the intuitive result that heart rate low values and heart rate median values increase together, hence giving the diagonal nature of the data points. In Figures 4-83 and 4-86, we again see the discrete nature of oxygen data.

In Figures 4-87 to 4-89, we display decision regions and scatter plots for oxygen artifact detection. Attributes chosen for the plotted axes are those in the corresponding decision tree model shown in Figure 4-8. We can see in all three of the figures that again, median values and low values (or median values of three points and median values of five points, as is the case in Figure 4-88) of the same signal increase together.

We have not focused upon data visualization techniques in our studies, though others, for example Combi *et al.* [39], have. Such techniques might prove very helpful in furthering understanding of the nature of one's data, and therefore, which machine learning techniques might work better.

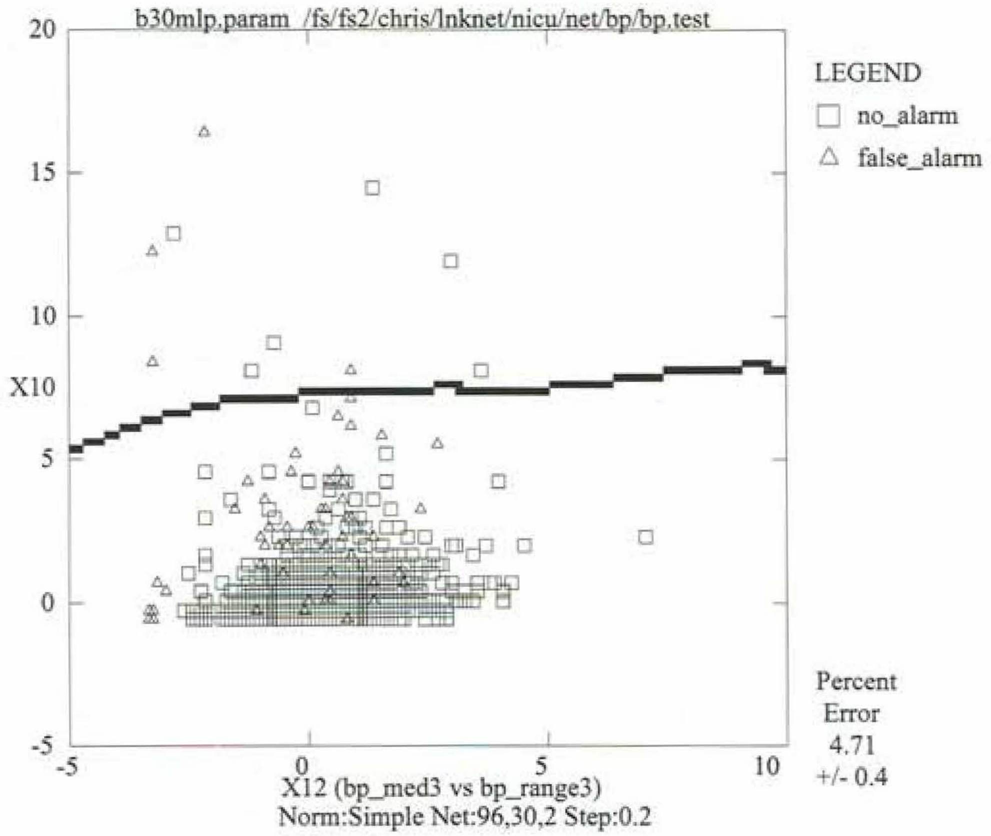


Figure 4-73: Decision region scatter plot for blood pressure artifact detection data cases for two attribute values, bp_med3 and bp_range3 (shown normalized). Lower region is the no_alarm decision region.

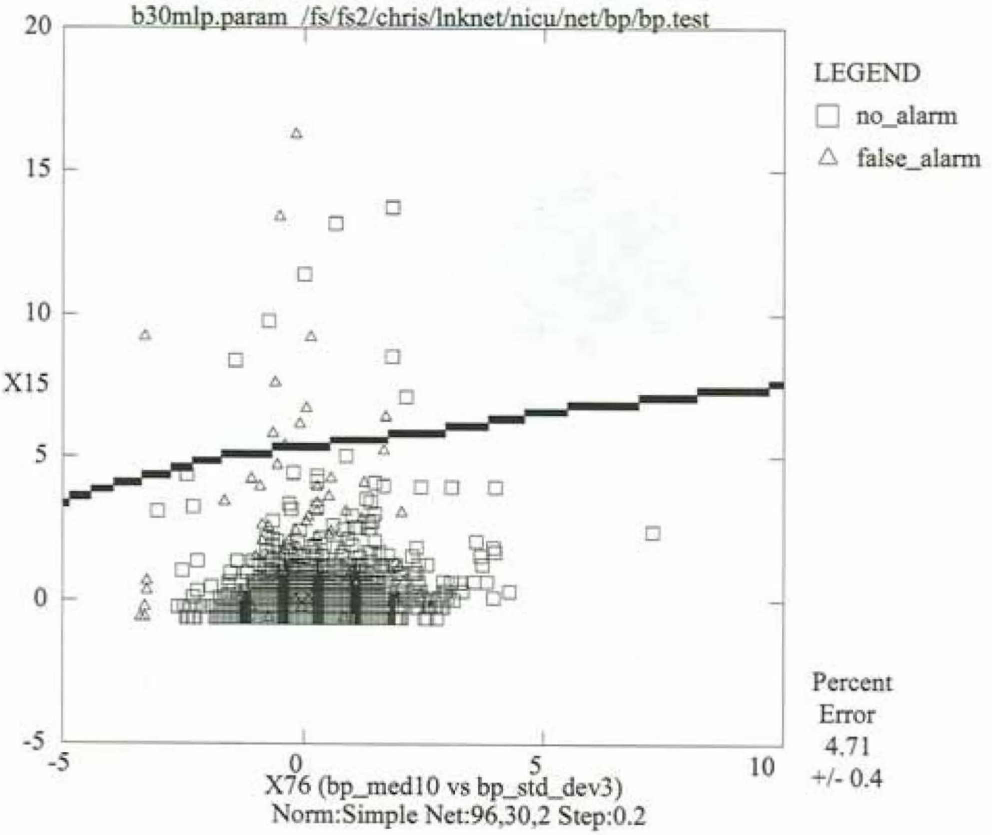


Figure 4-74: Decision region scatter plot for blood pressure artifact detection data cases for two attribute values, bp_med10 and bp_std_dev3 (shown normalized). Lower region is the no_alarm decision region.

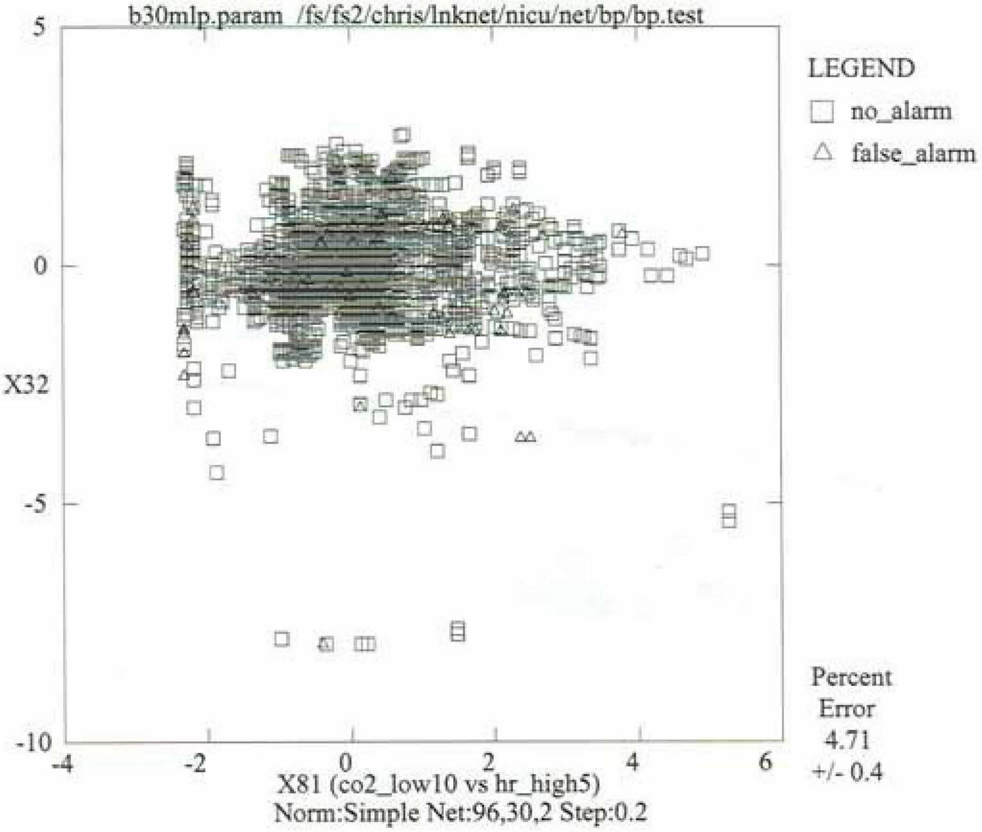


Figure 4-75: Scatter plot for blood pressure artifact detection data cases for two attribute values, co2_low10 and hr_high5 (shown normalized). No decision boundaries are present on the graphed axes.

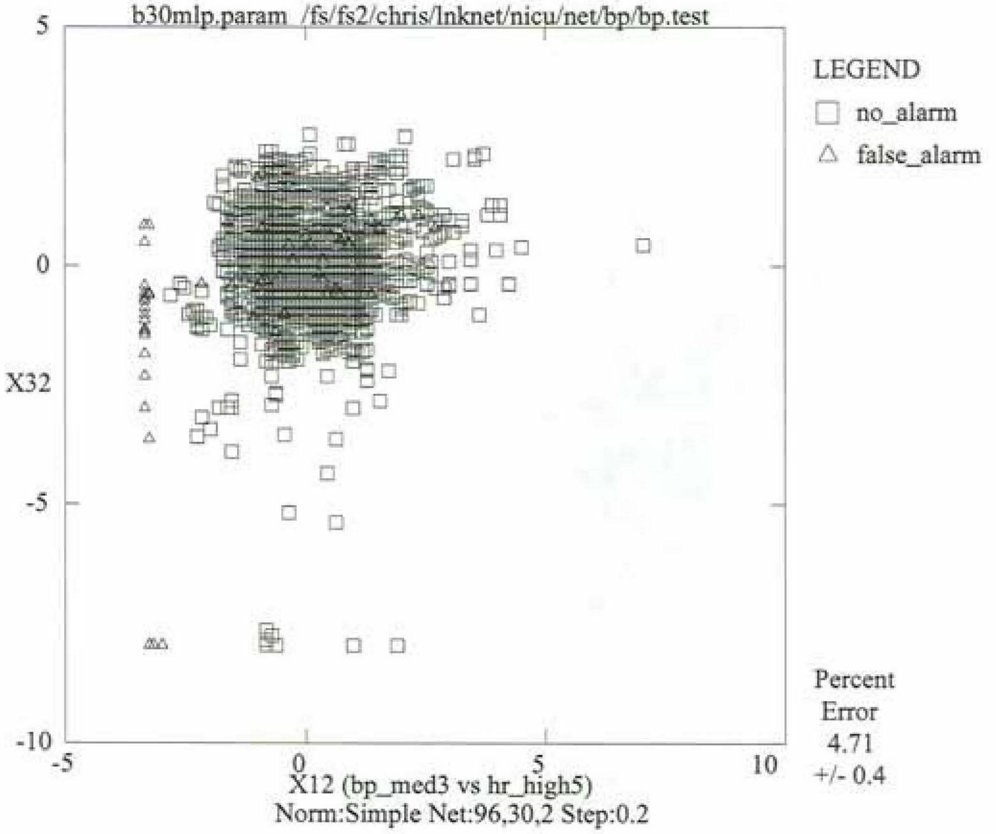


Figure 4-76: Scatter plot for blood pressure artifact detection data cases for two attribute values, bp_med3 and hr_high5 (shown normalized). No decision boundaries are present on the graphed axes.

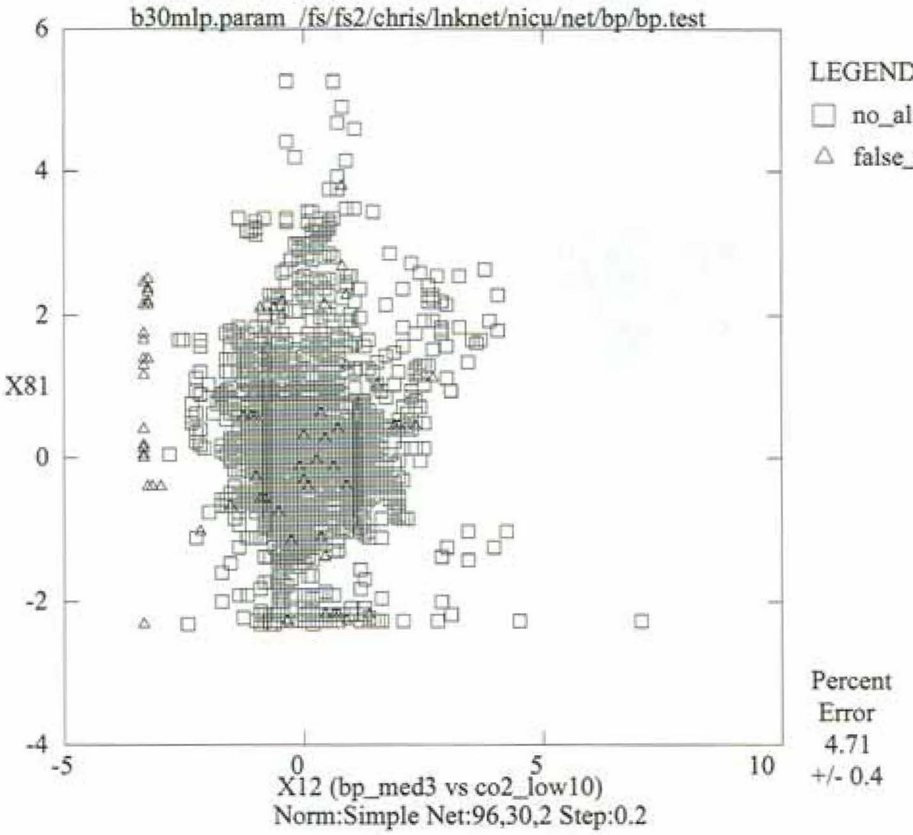


Figure 4-77: Scatter plot for blood pressure artifact detection data cases for two attribute values, bp_med3 and co2_low10 (shown normalized). No decision boundaries are present on the graphed axes.

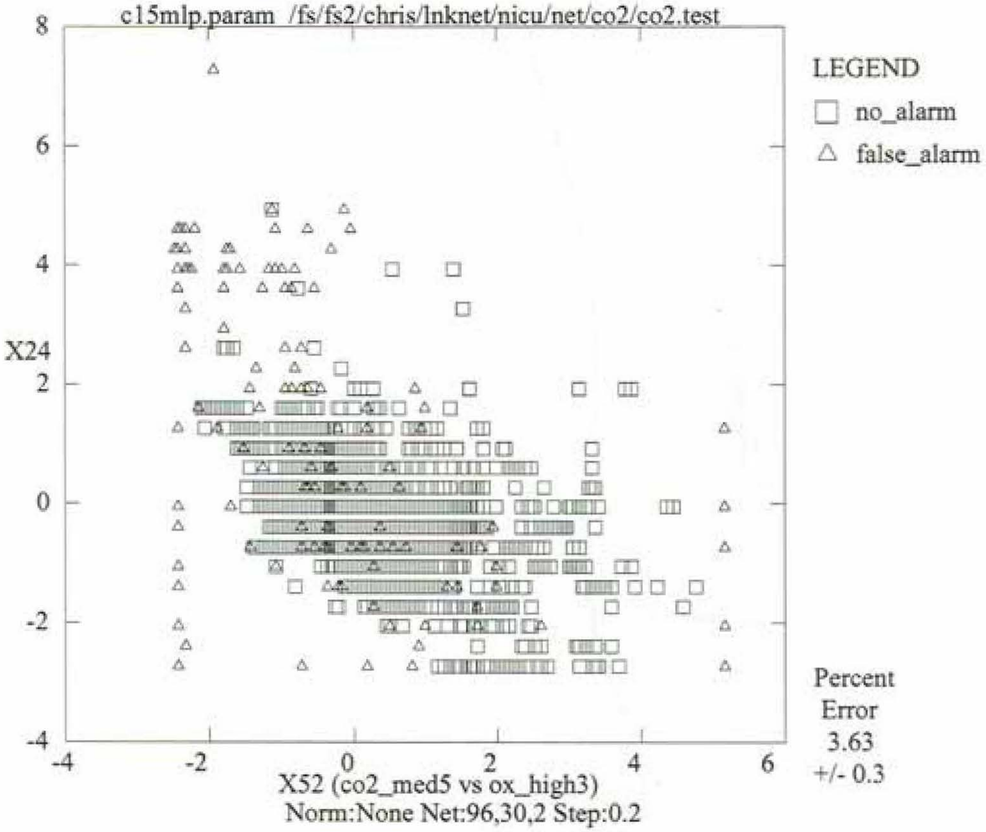


Figure 4-78: Scatter plot for carbon dioxide artifact detection data cases for two attribute values, co2_med5 and ox_high3 (shown normalized). No decision boundaries are present on the graphed axes.

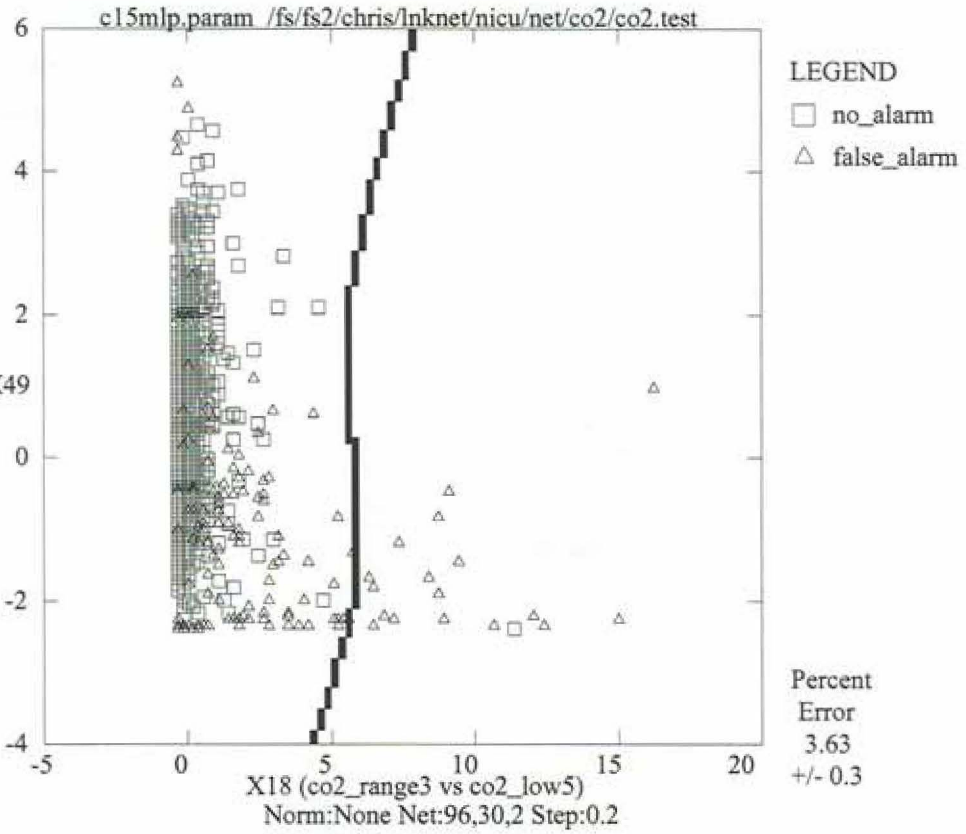


Figure 4-79: Decision region scatter plot for carbon dioxide artifact detection data cases for two attribute values, co2_range3 and co2_low5 (shown normalized). Left region is the no_alarm decision region.

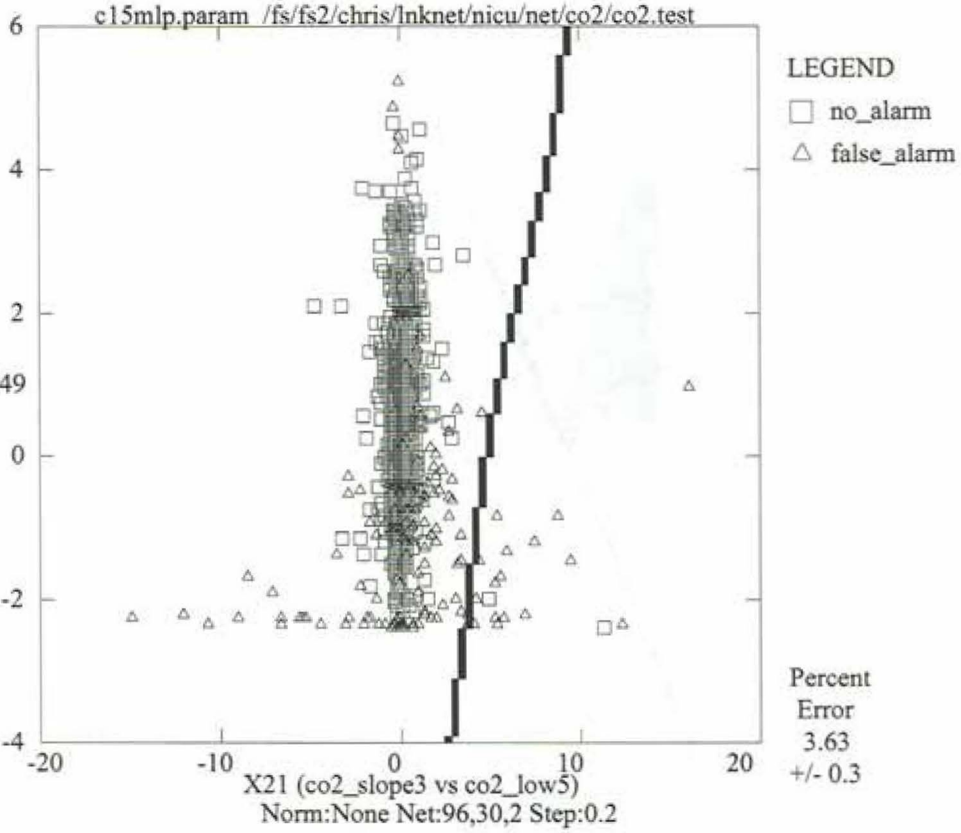


Figure 4-80: Decision region scatter plot for carbon dioxide artifact detection data cases for two attribute values, co2_slope3 and co2_low5 (shown normalized). Left region is the no_alarm decision region.

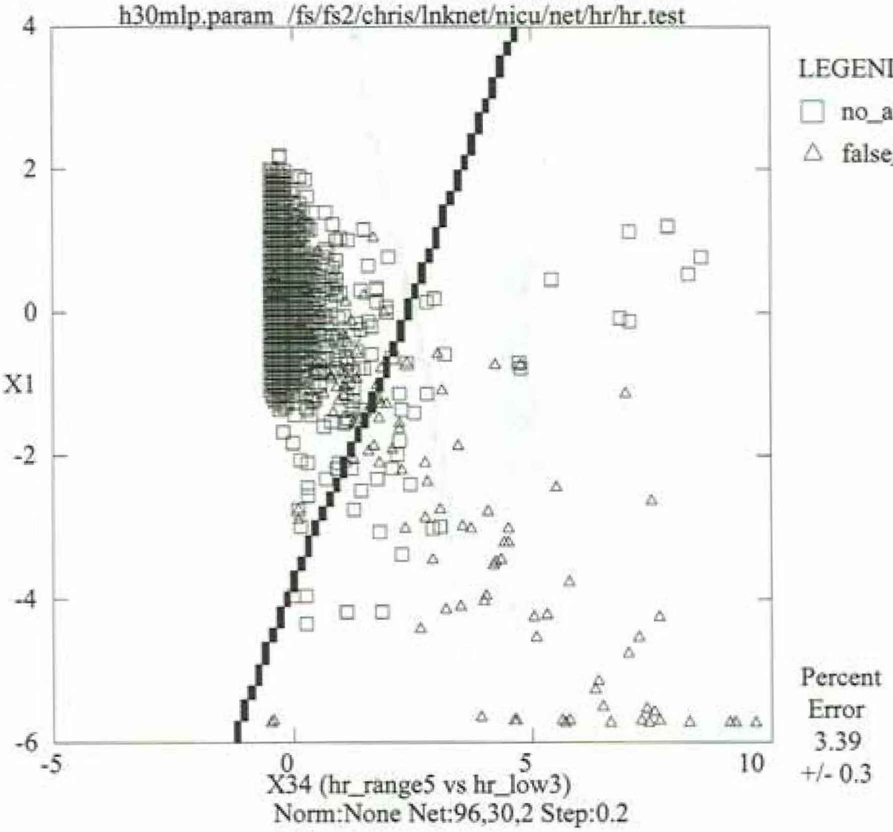


Figure 4-81: Decision region scatter plot for heart rate artifact detection data cases for two attribute values, hr_range5 and hr_low3 (shown normalized). Left region is the no_alarm decision region.

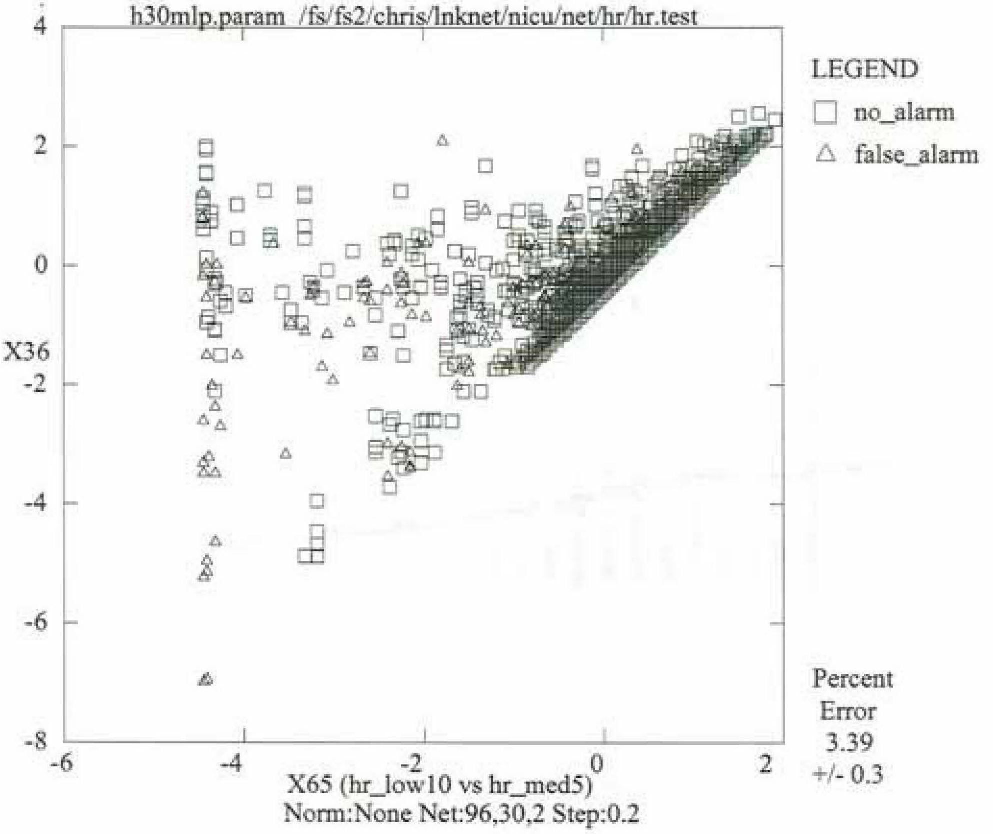


Figure 4-82: Scatter plot for heart rate artifact detection data cases for two attribute values, hr_low10 and hr_med5 (shown normalized). No decision boundaries are present on the graphed axes.

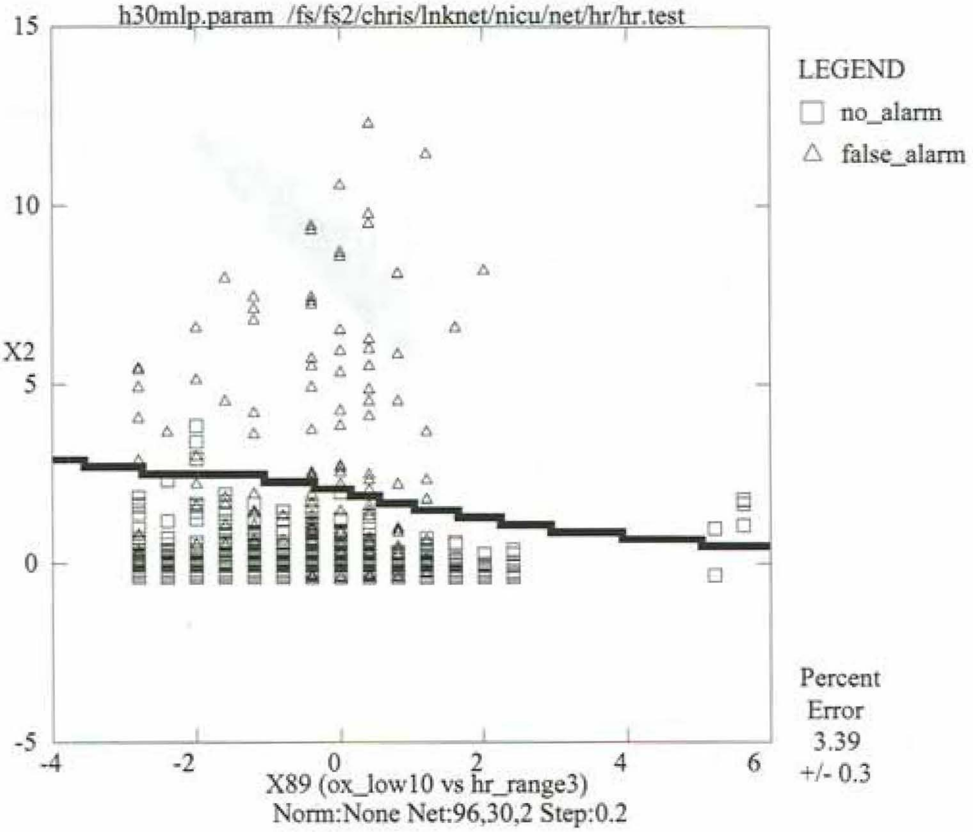


Figure 4-83: Decision region scatter plot for heart rate artifact detection data cases for two attribute values, ox_low10 and hr_range3 (shown normalized). Lower region is the no_alarm decision region.

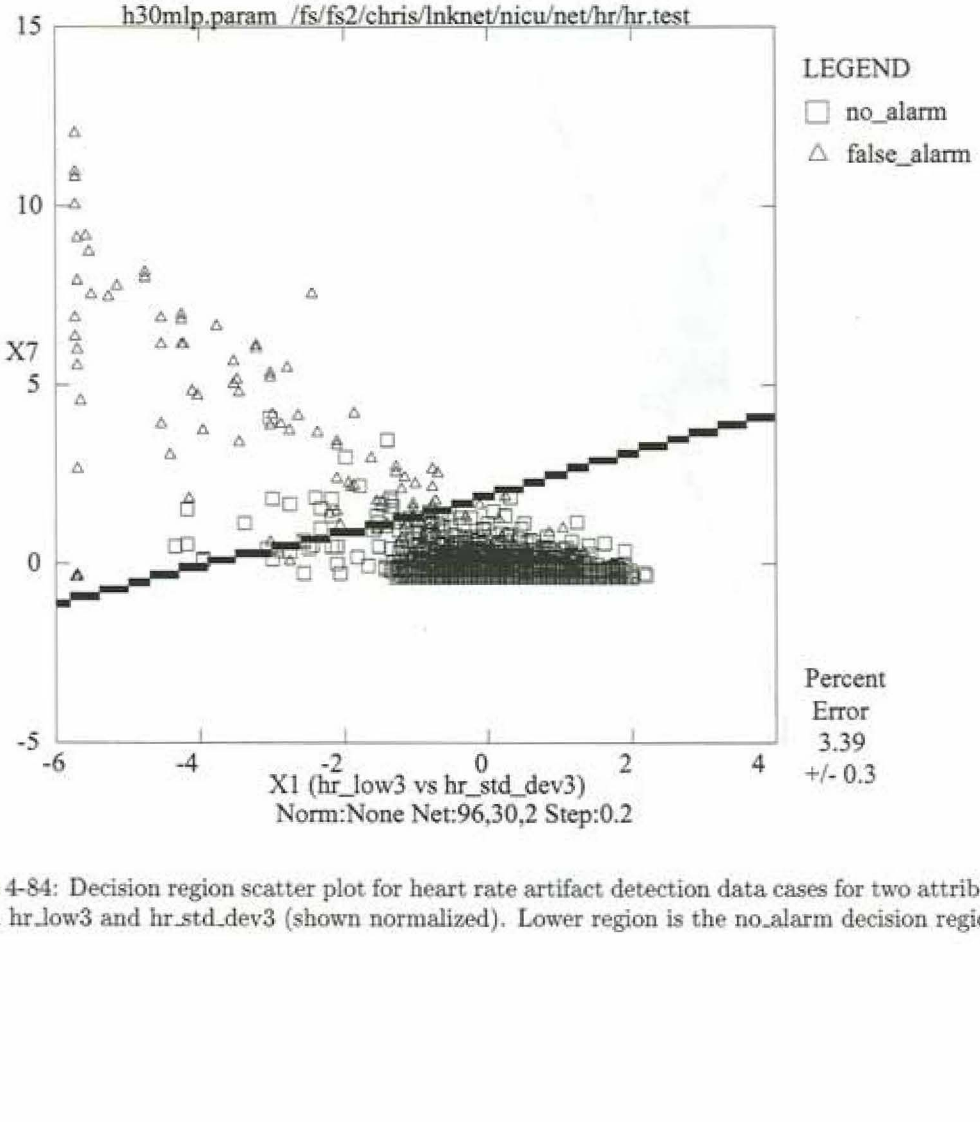


Figure 4-84: Decision region scatter plot for heart rate artifact detection data cases for two attribute values, hr_low3 and hr_std_dev3 (shown normalized). Lower region is the no_alarm decision region.

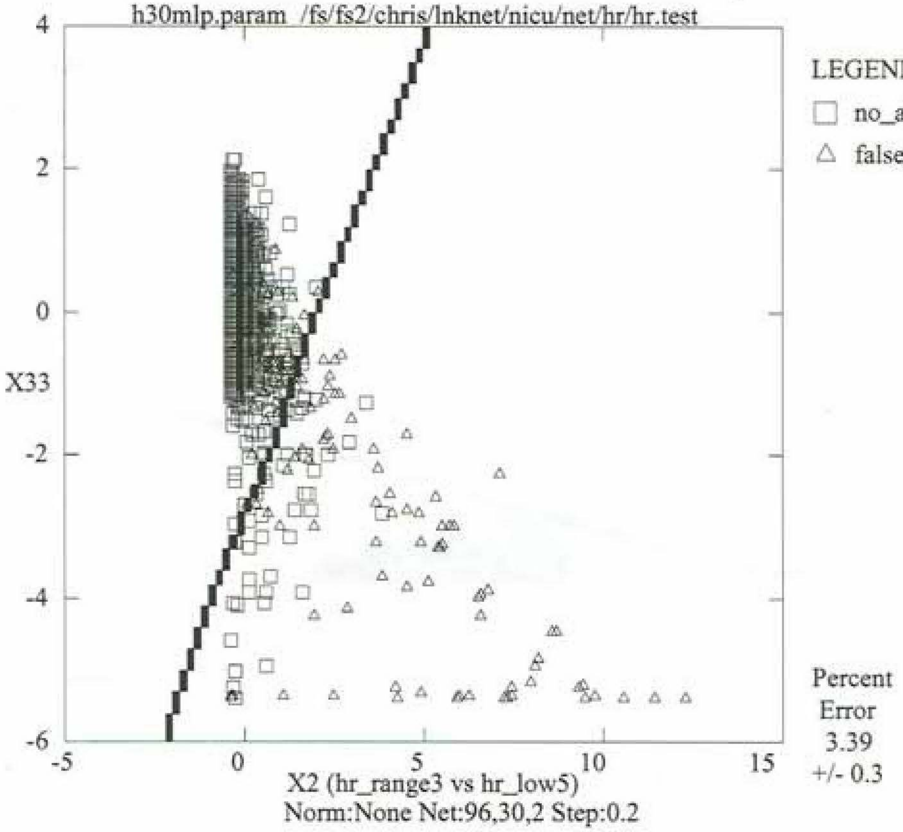


Figure 4-85: Decision region scatter plot for heart rate artifact detection data cases for two attribute values, hr_range3 and hr_low5 (shown normalized). Left region is the no_alarm decision region.

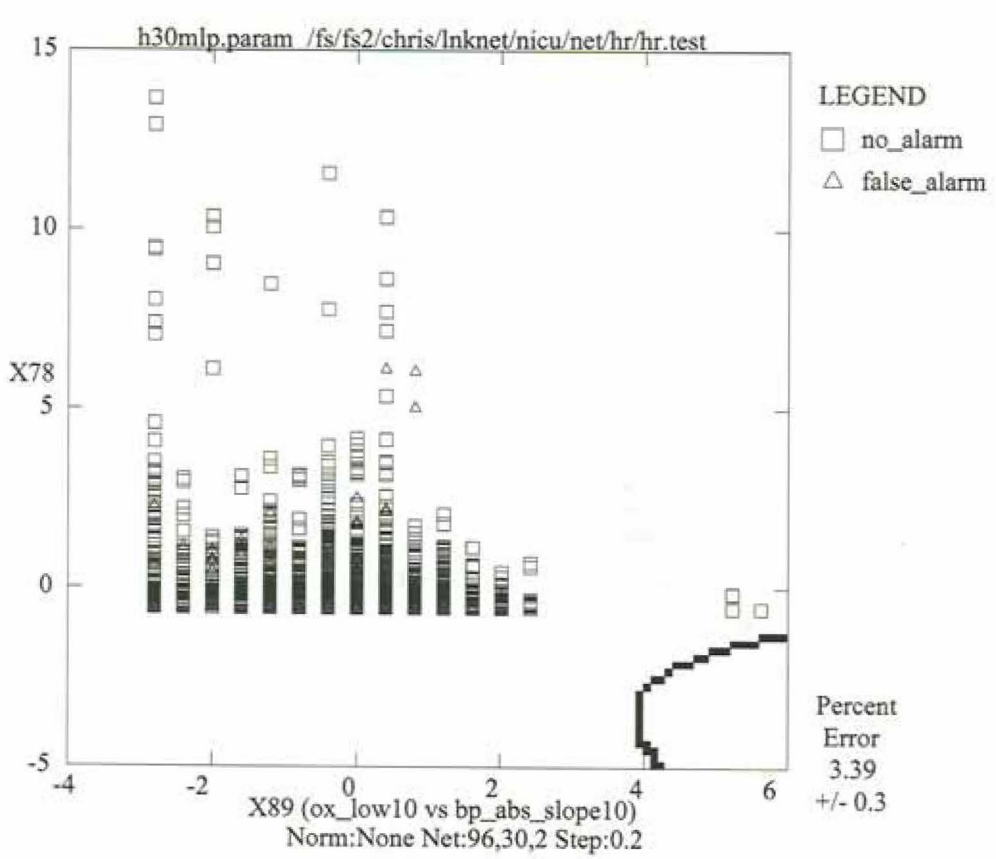


Figure 4-86: Decision region scatter plot for heart rate artifact detection data cases for two attribute values, ox_low10 and bp_abs_slope10 (shown normalized). Lower right corner region is the artifact (labeled 'false_alarm') decision region.

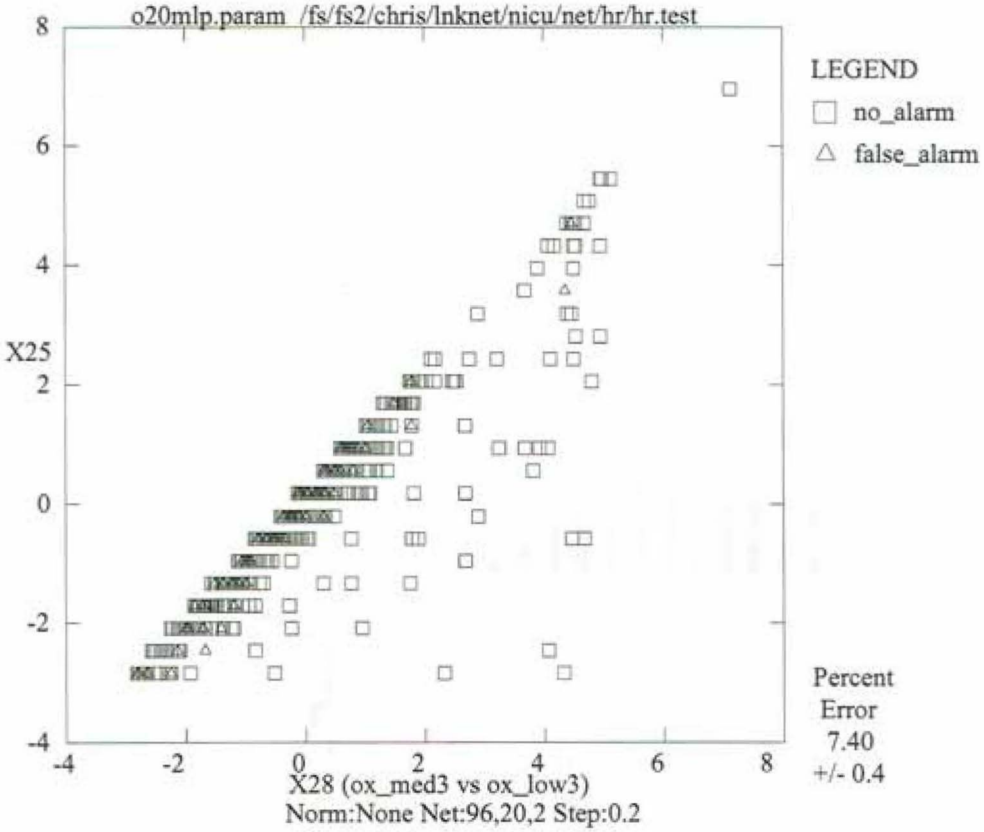


Figure 4-87: Scatter plot for oxygen artifact detection data cases for two attribute values, ox_med3 and ox_low3 (shown normalized). No decision boundaries are present on the graphed axes.

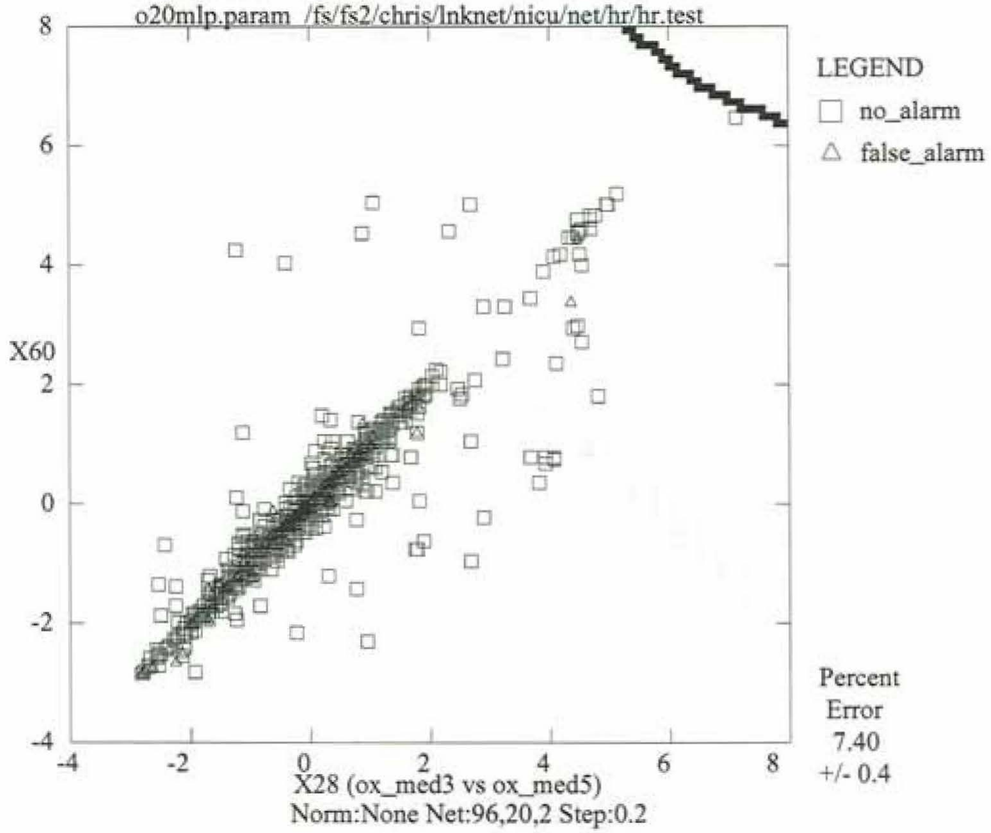


Figure 4-88: Decision region scatter plot for oxygen artifact detection data cases for two attribute values, ox_med3 and ox_med5 (shown normalized). Upper right corner region represents the artifact (labeled 'false_alarm') decision region.

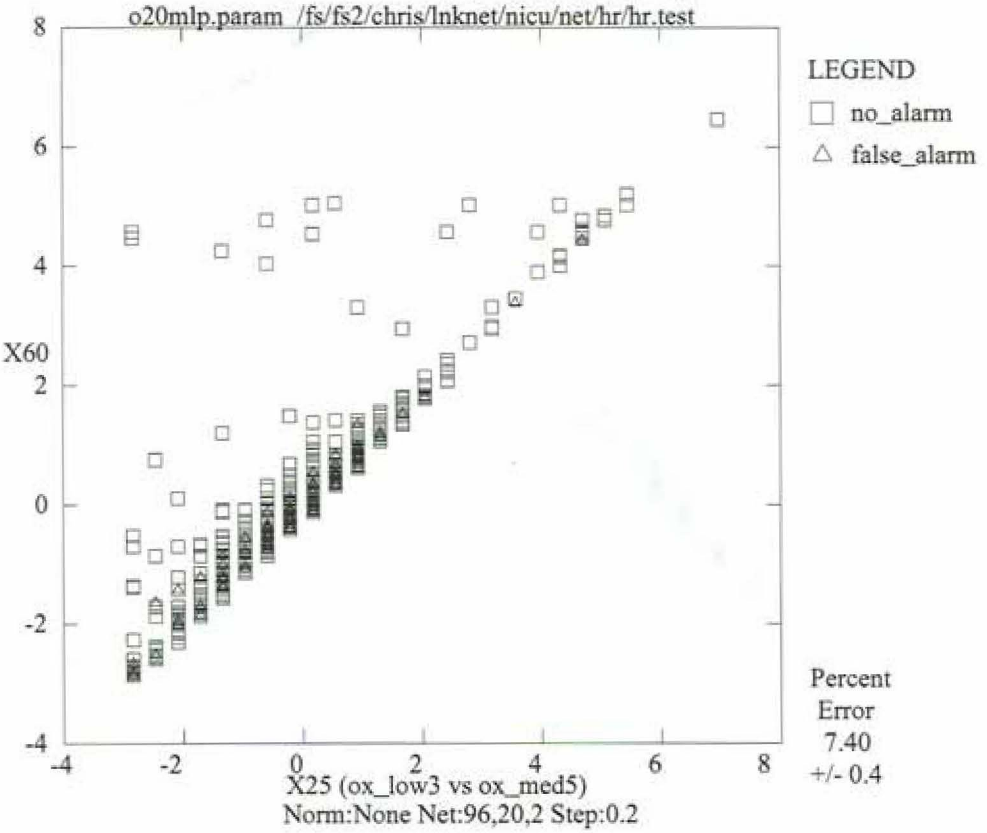


Figure 4-89: Scatter plot for oxygen artifact detection data cases for two attribute values, ox_low3 and ox_med5 (shown normalized). No decision boundaries are present on the graphed axes.

4.4.3 Data Granularity

These experiments have shown that multi-signal detection of ICU artifacts by decision trees built from data of 1-minute granularity could do fairly well when run on data of 1-minute granularity, with ROC curve areas ranging from 89.41% to 99.93%. We have also found that decision trees built from data of 1-second granularity could perform effectively on data of 1-second granularity, even more so effectively in fact, with ROC curve areas ranging from 99.40% to 100.00%. This was somewhat to be expected since 1-second data by nature contain more information than 1-minute data.

The most surprising finding, however, was that models built with 1-minute granularity data performed extremely well on the test sets derived from 1-second data, both 180-300-600 and 3-5-10 experiments. ROC curve areas for Experiment 1 (1-minute models run on the 180-300-600 1-second test sets) ranged from 96.30% to 100.00%, while ROC curve areas for Experiment 2 (1-minute models run on the 3-5-10 1-second test sets) ranged from 99.70% to 100.00%. For three of the four models (blood pressure, carbon dioxide, and heart rate), the 1-minute models run on 1-second data in both Experiments 1 and 2 had better results than the same 1-minute models run on 1-minute data. In the fourth model (oxygen), the results of running the 1-minute model on 1-second data in Experiment 1 (ROC curve area of 98.81%), although not quite as good as the results of the 1-minute model run on 1-minute data (ROC curve area of 99.93%), were still very good. The results of running the 1-minute model on 1-second data in Experiment 2 (ROC curve area of 99.92%) were equal ($p = 0.40$) to the results of the 1-minute model run on 1-minute data (99.93%).

These findings can be exploited in appropriate situations. In situations in which an event develops slowly over several minutes, for example, in some cases of pneumothorax, models for pneumothorax detection could be developed with 1-minute granularity data and then run on 1-second data that are processed using the same *time intervals*, as done in Experiment 1. On the other hand, in situations in which an event is fleeting, such as short-lived false alarm soundings lasting only a couple seconds each, models for false alarm detection could be developed with 1-minute granularity data and then run on 1-second data that are processed using the same *numbers of values*, as done in Experiment 2. We found our a priori assumption–that 1-minute models would perform poorly on 1-second data–to be incorrect.

A comparison of the decision tree models themselves is also interesting. For blood pressure artifact detection, both models found the median of three minutes of blood pressure raw values ('bp_med3' in the 1-minute model and 'bp_med180 in the 1-second model) to be a useful first predictor of artifact status. For detection of artifacts in the carbon dioxide signal, both models also found the median value to be a useful first predictor of artifact status, though the 1-minute model calculated this median over a five-minute time interval ('co2_med5') while the 1-second model calculated it over a three-minute time interval ('co2_med180'). In the oxygen artifact detection models, not only was the median over three minutes present in both models, but moreover, both models used two identical threshold values for labeling a feature vector as artifact ("ox_med3 > 20" and "ox_med3 <= 0.5" in the 1-minute model, "ox_med180 > 20" and "ox_med180 <= 0.5" in the 1-second model). The presence of identical attributes and thresholds is further reassurance that development of artifact detection models is similar, and valid, regardless of the granularity of data used in the development process when data are compressed using arithmetic mean. The usefulness of the median value in this domain is also consistent with the findings by Makivirta [123].

The heart rate models did not contain identical attributes. The 1-second heart rate model consisted of only one attribute, the maximum value over three minutes ('hr_high180'), which was not present in the 1-minute heart rate model. The 1-minute heart rate model's first predictor of artifact status was the minimum value over three minutes ('hr_low3') instead. The heart rate models were developed using a different class labeling scheme than that used for the other three signals; this would not, however, be expected to account for the difference observed since identical class labeling techniques were used for both the 1-minute and the 1-second heart rate models.

The data granularity results indicate that while artifact detection models developed from 1-second data are effective when tested on 1-second data, so too are models developed from 1-minute data effective when tested on 1-second data. This is a very important finding since developing models with 1-minute data has a tremendous advantage: during model development, more hours of ICU monitor data can be processed in less time. This is useful not only in general, but especially for data-intensive domains such as the ICU in particular. Because of the relative scarcity of artifacts, which are scattered sparsely amongst all the 'normal' values, voluminous amounts of physiological data streams need to be examined to ensure development of more robust models. The 1-minute models required processing of approximately 48,000 raw data values (200 hours multiplied by 60 minutes per hour multiplied by four data signals), while the 1-second models required processing of approximately 1,065,600 raw data values (74 hours multiplied by 3600 seconds per hour multiplied by four data signals). Thus, developing models from 1-minute data required roughly two orders of magnitude fewer calculations to process more than 2.5 times the number of monitor-hours. Moreover, these 1-minute models still performed well 'in the clinical setting' scenario, i.e., on 1-second monitor data. Data compression of temporal data by arithmetic mean, therefore, can be an effective method for decreasing knowledge discovery processing time without compromising learning. Future studies should focus on validating these techniques in other domains.

4.4.4 Class Labeling

Results from class labeling experiments indicate that any of several different labeling techniques can be used effectively in building an ICU artifact detection model.

In the front-labeling versus end-labeing experiments, we observed nearly identical results with both methods. The derived decision trees were sometimes even identical using these two labeling methods. Only in one comparison out of eight (two different time-interval models for each of four signal artifacts) did end-labeling out-perform front-labeling. In an area such as ICU monitoring, therefore, end-labeling, which is more useful from a clinical standpoint, should be used to be able to detect an event as quickly as possible when it occurs.

In comparisons of strictness of class labeling, we found that very good models can be derived from each of the strictness methods. Models derived from the strictest method of labeling, however, are inconsistent in performance when run on test cases created from less strict labeling methods. We conclude, therefore, that when possible, all available data and the least strict labeling method should be used to derive models. In this way, more robust models are likely to be developed. If models with inadequate performance result, due to inaccurate annotations, for example, use of more strict labeling methods may then be useful.

4.4.5 Multi-Phase Techniques

Multi-phase models, given only slope information, were still able to achieve fairly good performance. Compared to single-phase models made from 1-second granularity data, however, which had perfect accuracy, multi-phase models performed significantly poorer ($p < 0.0001$).

In one case, for carbon dioxide artifacts, the multi-phase decision tree model given the same inputs as the multi-phase neural network did much better than that neural network model. In another case, for oxygen artifacts, the neural network model did much better than the corresponding decision tree model. For heart rate and blood pressure, performance was equal for the two types of classifiers.

Inspection of the c4.5 multi-phase decision tree models can in some cases help us to better understand the way in which artifacts manifest in the measured values. This is especially true for BP and CO_2 artifacts in this case because both models are quite small. The BP model shown in

Figure 4-67, for example, describes that BP artifacts occur when the BP phase 2 slope over five seconds is greater than 3.3; or, if the BP phase 2 slope over five seconds is less than or equal to 3.3, and the BP phase 2 slope over 30 seconds is greater than 0.95, and the BP phase 1 slope over 30 seconds is less than or equal to -0.35. Once the models become much larger, however, as is the case for oxygen and HR artifacts, it is difficult to glean a simplistic pattern or set of rules from the model.

The pre-specified 'biphase' patterns appear to be of minimal use since very good models were developed without these inputs. This is likely because we did not provide an exhaustive list of the possible pattern combinations given our calculated slopes; as a result, the best fitting patterns were not options. Clearly, using only piecewise phase characteristics is much more flexible and simplistic a method than trying to enumerate all possible combinations as inputs. Future work should explore the additional characterization of individual phases with functions other than linear ones, as well as the use of more than two phases for learning.

4.4.6 Limitations

Clearly this case study on artifact detection in the NICU has its limitations. Most notably is the methodology employed for data annotation. Ideally, such annotations would be created prospectively with adequate details for understanding any surrounding clinical conditions occurring during alarms. Although it is likely that artifacts in the data signals would correspond to soundings of false alarms in the ICU, this cannot be verified retrospectively. Prospective annotation would furthermore ideally record not only false alarms but also true alarms. Only with knowledge of true alarm occurrences can one thoroughly examine how well a model performs to decrease false alarms–it must do so without compromising the detection of true alarms. In the current study, annotations of data artifacts were made retrospectively, hence suffer from potential inclusion bias. Additionally, knowledge of the actual false or true alarms or clinical happenings is unavailable.

The current work may also be limited by the arbitrary choosing of certain values, such as the time intervals with which derived values were calculated. Future work should aim to systematically analyze the effect on model performance of using different time intervals. (This is done for the medical ICU experiments in Chapter 5.) Correlation of clinical knowledge regarding time in this domain with choosing of time intervals for model development may also be useful when data of the appropriate granularity are available. For the BP, CO_2, and O_2 signals, the class label, 'artifact,' was assigned for those time intervals containing at least half of the raw data values annotated with an asterisk indicating artifact. The threshold of half was set arbitrarily and thus might be another area for further experimentation.

Despite these limitations, this study serves a useful purpose; it uses real ICU bedside data in the form in which they are available and explores how such data may be integrated to, currently, detect artifacts, and ultimately, provide more effective bedside monitoring.

Regarding class labels, all sets of derived values in the HR signal labeled 'transition' were disregarded in the current experiments. Study of how to predict transition periods into and out of artifact occurrences may prove equally useful for rapid detection of artifacts (and thus possibly false alarms) in bedside monitor data.

Also discarded in the current experiments were all incomplete data, for example, data from a given patient for whom there were fewer than the four monitored signals available. This made analysis and model development more tractable for the current study; realistically, however, it is inconceivable that all input data used by a multivariate monitoring system would always be available. It is imperative, therefore, that such monitoring systems be able to work effectively in the presence of incomplete data. Future work should identify possible reasons for missing data in this domain and strive to handle each type of scenario appropriately. This may require study of classification with missing data in other domains [116, 158, 165] and/or use or development of novel techniques [207, 222] for handling missing values.

The results presented in this chapter indicate that integration of multiple signals by means of applying a classification system to sets of derived values from physiologic data streams may be a viable approach to detecting artifacts in the neonatal ICU. Application of the TrendFinder paradigm has enabled exploration of this approach.

Chapter 5

TrendFinder to Detect True Alarms in the Medical ICU

We now demonstrate how the described event discovery process can be used to decrease false alarms in the medical intensive care unit (MICU). Previous studies have shown that as many as 86% of alarm soundings in the ICU are actually false [118, 206]. Current systems for monitoring vital signs typically sound an alarm any time the monitored signal surpasses a high threshold limit or falls below a low threshold limit. This simplistic rule, however, usually results in a large number of spurious readings that cause false alarms. This can lead to several problems (discussed in Chapter 3), the most important end result being compromised patient care.

Our approach is to develop multi-signal, machine-learned models, using the TrendFinder paradigm, which are able to detect 'true alarm' events from bedside time-series data.

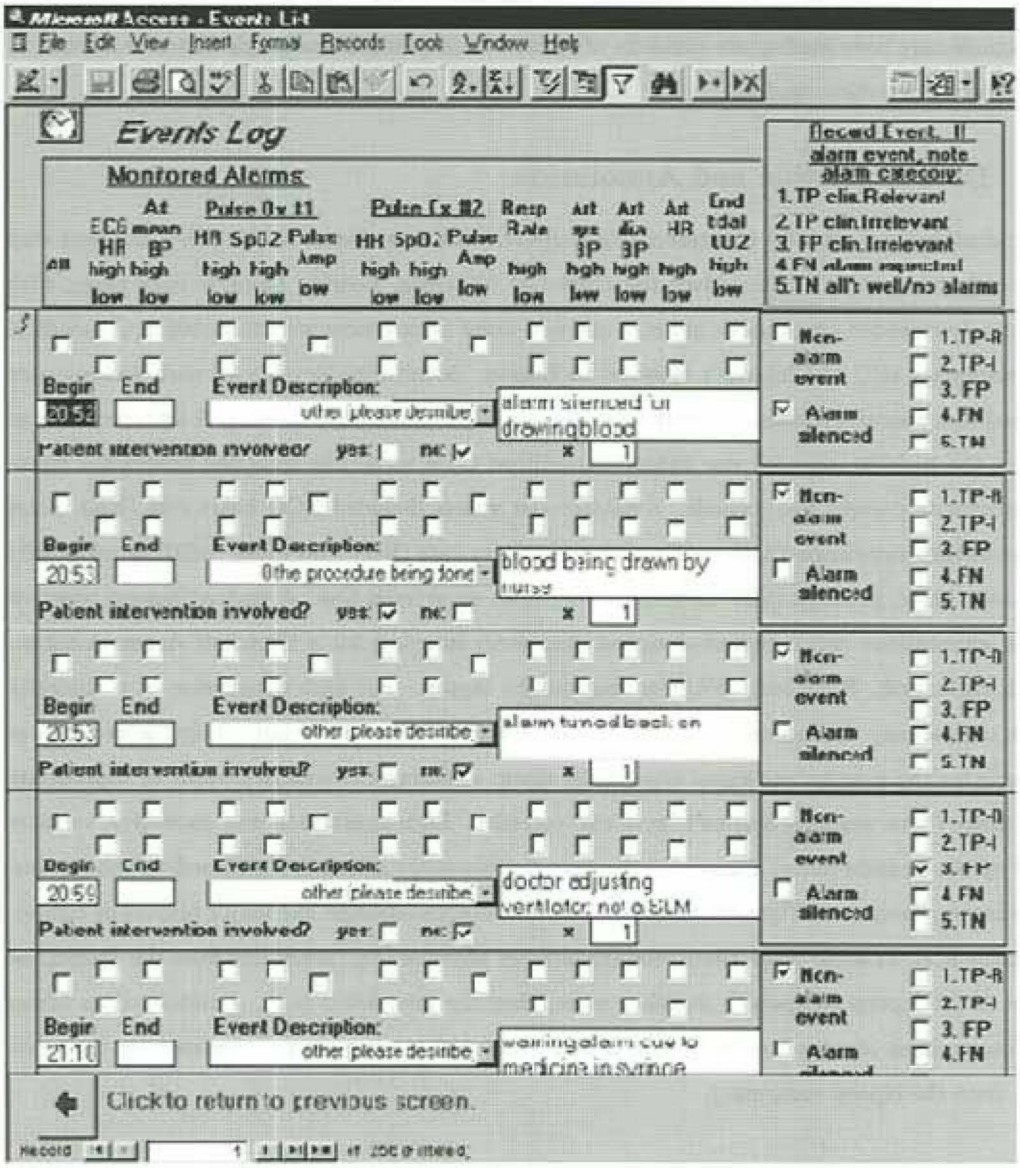
5.1 Methods

5.1.1 Event Identification

As an example, we choose our primary event of interest to be true alarms that are clinically relevant (of any cause) occurring in the ICU on the arterial line systolic blood pressure signal. We choose this signal because, as was seen in Table 3.3 in Chapter 3, the arterial line blood pressure signal had the largest number of clinically-relevant true alarms and thus is initially more suitable for machine learning purposes. We also demonstrate the same methods for detecting clinically-relevant true alarms on four other ICU monitored signals–arterial oxygen saturation from a pulse oximeter, ECG heart rate, arterial line mean blood pressure, and respiratory rate. We expect that some of these latter signals may have inadequate numbers of positive examples (clinically-relevant true alarms) to create effective detection models.

5.1.2 Data Collection and Annotation

Having identified our event(s) of interest (clinically-relevant true alarms in the ICU), the next step was to collect annotated data. Over the course of 12 weeks, bedside monitor data along with prospectively recorded annotations of event and non-event occurrences were recorded in the medical multidisciplinary ICU of Children's Hospital in Boston. Monitoring devices for each patient were connected to a SpaceLabs bedside monitor (SpaceLabs Medical, Redmond, WA). A laptop computer placed at the bedside recorded raw values transmitted via a serial line from the SpaceLabs monitor approximately every five seconds. Available raw values included ECG heart rate (HR), pulse oximeter oxygen saturation (S_pO_2 or O_2 sat), respiratory rate (RR), and mean (MBP or mean BP) and systolic blood pressure (SBP or systolic BP) from an arterial line. A trained human observer recorded annotations into a custom-designed data entry interface for a Microsoft Access database program (Microsoft, Redmond, WA) running on the laptop. For each occurrence of a clinically relevant true alarm, the trained observer created a time-stamped note indicating the true alarm occurrence for the appropriate signal type. False alarm soundings, as well as periods of appropriate alarm silence ('true negative alarms'), were also recorded. Each annotation was, moreover, verbally verified by the bedside nurse. Figure 5-1 displays the computer interface for recording annotations into the laptop computer. Figure 5-2 depicts the relationships amongst the major players or components in the ICU. In some cases, these relationships are physical (e.g., serial line from the SpaceLabs monitor to the laptop computer); in other cases, they are interactive (e.g., sounding of an alarm causes the trained observer to record the alarm along with a time-stamp, depicted by the clock symbol, onto the laptop computer).



Record Event. If alarm event, note alarm category:
1. TP clin.Relevant
2. TP clin.Irrelevant
3. FP clin.Irrelevant
4. FN alarm expected
5. TN all's well/no alarms

Begin: 20:53 End:
Event Description: other please describe – alarm silenced for drawingblood
Patient intervention involved? yes: □ no: ☑ x 1
□ Non-alarm event ☑ Alarm silenced
□ 1.TP-R □ 2.TP-I □ 3. FP □ 4.FN □ 5.TN

Begin: 20:53 End:
Event Description: Other procedure being done – blood being drawn by nurse
Patient intervention involved? yes: ☑ no: □ x 1
☑ Non-alarm event □ Alarm silenced
□ 1.TP-R □ 2.TP-I □ 3. FP □ 4.FN □ 5.TN

Begin: 20:53 End:
Event Description: other please describe – alarm turned back on
Patient intervention involved? yes: □ no: ☑ x 1
☑ Non-alarm event □ Alarm silenced
□ 1.TP-R □ 2.TP-I □ 3. FP □ 4.FN □ 5.TN

Begin: 20:59 End:
Event Description: other please describe – doctor adjusting ventilator, not a SLM
Patient intervention involved? yes: □ no: ☑ x 1
□ Non-alarm event □ Alarm silenced
□ 1.TP-R □ 2.TP-I ☑ 3. FP □ 4.FN □ 5.TN

Begin: 21:10 End:
Event Description: other please describe – warningalarm due to medicine in syringe
☑ Non-alarm event □ Alarm silenced
□ 1.TP-R □ 2.TP-I □ 3. FP □ 4.FN

Figure 5-1: Computer interface for recording annotations in the MICU.

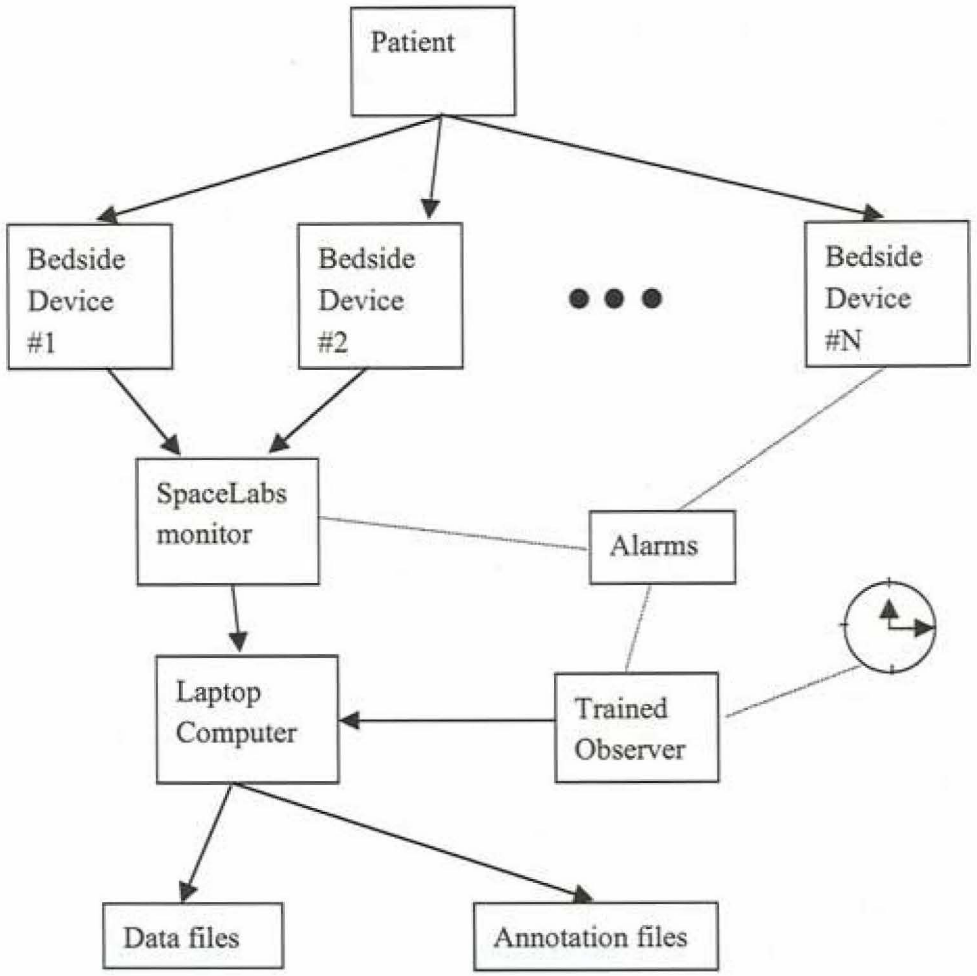


Figure 5-2: Annotated data collection setup.

5.1.3 Data Preprocessing

Feature attribute derivation

Preprocessing of the annotated data first involved calculation of eight different mathematical quantities for each successively overlapping group of raw data values. These calculated quantities include moving mean ('avg'), median ('med'), maximum value ('high'), minimum value ('low'), range, linear regression slope ('slope'), absolute value of linear regression slope ('abs_slope'), and standard deviation ('std_dev'). As mentioned in the previous chapter, moving mean and median were chosen because of their potential usefulness described in earlier work [123, 205]; overall, these were chosen because of their usefulness in artifact detection, as described in Chapter 4.

The eight derived values were calculated for each of three different time intervals. The time intervals initially chosen were 10 seconds, 20 seconds, and 45 seconds, corresponding to feature derivation over two raw values, four raw values, and nine raw values, respectively ('2-4-9'). A second set of time intervals chosen for experimentation were six raw values, 12 values, and 24 values ('6-12-24'), corresponding to approximately 30, 60, and 120 seconds. Both of these sets of time intervals were chosen with the general knowledge that false alarms tend to occur fleetingly, while true alarms tend to develop more slowly; the exact numbers themselves were otherwise chosen arbitrarily. We discuss a more principled method for choosing time intervals later in this chapter.

Signal Experiments The 24 described values (eight different quantities for each of three different time intervals) were calculated for each of the five recorded data signals, resulting in sets containing 120 feature attributes (120-dimensional feature vectors) for multi-signal learning. We additionally explored single-signal learning by using only the 24 values derived from a particular signal type to find true alarms of that signal type. This was done for each of the five signals.

Time Interval Selection To better understand the effect of time interval selection on model performance, we derived multi-signal sets of feature attributes of only one time interval at a time, ranging from two values (10 seconds) to 36 values (180 seconds). This was done for both systolic blood pressure alarms and oxygen saturation alarms. By looking for the lowest numbers of errors, we selected the two thus-defined 'best' single time intervals for use in the regular multi-signal system (with attributes derived from three time intervals). The third time interval selected was six values (30 seconds) to correspond with the '6' in the 6-12-24 experiments; this enables us to compare performance of these two sets of models on the same test cases, as will be explained in the next section on class labeling.

Class Labeling

Each feature vector was next labeled according to the annotations that had been prospectively recorded by the trained observer. Feature vectors whose attributes were derived from raw values labeled 'clinically-relevant true alarm' were given the true alarm class label (see Chapter 3 for details about the alarm categorization method during annotation). Feature vectors whose attributes were derived from raw values occurring during false alarm or true negative (no alarm) periods were labeled 'no alarm' (meaning that the desired result was to have no alarm sound at those times). Feature vectors whose attributes were derived from raw values spanning more than one label type were not used in model derivation for this set of experiments. Class labels for the 2-4-9 experiments were derived from the longest time interval, nine values, or approximately 45 seconds. Class labels for the 6-12-24 experiments were derived from the smallest time interval, six values, which was temporally located within the latter-most part of the longer time intervals (i.e., we used the end-labeling technique, as described in Chapter 4). We thus chose six values to be the smallest interval for our detailed time interval selection experiments in order to use cases similarly labeled on contiguous sets of six values.

Data Partitioning

Data (in the form of labeled feature vectors) were divided into a training data set, consisting of 70% of the data; a test set, consisting of 70% of the remaining 30% of the data (21%); and an evaluation set, consisting of the rest of the data (9%). Use of each data set has been described elsewhere in Chapters 2 and 4.

5.1.4 Model Derivation

The training data were then given to both a decision tree induction system (c4.5) [166] and a neural network classifier system (LNKnet) (Lincoln Laboratory, Lexington, MA). As described elsewhere, the decision tree system allows for model experimentation in various ways, such as changing the 'selectivity' ('c') of pruning a tree, or the number of cases ('m') necessary in a branch to grow the tree. Decision tree models were preferred if they had fewer errors when run on the evaluation set, and/or smaller size with little to no increase in the number of errors when run on the evaluation set. The neural network system allows for model variation also, for example, by changing the number of layers of hidden nodes to be included in the network structure, or by changing the number of hidden nodes per layer. Networks with simpler structure and fewer hidden nodes, having similar performance on the evaluation set compared to more complicated networks, were preferred.

Signal Experiments

Both decision trees and neural networks were developed for each of the five true alarm signal types using multi-signal 2-4-9 feature attributes. For the multi-signal 6-12-24 feature attributes, decision trees and neural networks were developed for the systolic blood pressure true alarms and the oxygen saturation true alarms.

Neural networks were also developed for each of the single-signal 2-4-9 feature attribute sets. Decision trees were attempted on the single-signal data sets; details are given in the Results section.

Time Interval Experiments

Time interval experiments were carried out first with the use of only decision trees for each of the single-time feature value sets. For each systolic BP alarm single-time feature value set, three decision trees were created, using the following option settings: (1) c = 25%, m = 2; (2) c = 25%, m = 10; (3) c = 10%, m = 2. For each oxygen saturation alarm single-time feature vector, two decision trees were created: (1) c = 25%, m = 2; (2) c = 25%, m = 10. Time intervals that had collectively fewest errors amongst the trees for a given time interval and a given signal alarm type, and which were representative of different time spans (e.g., not within 30 seconds of each other), were chosen for that signal (signal alarm type).

The resulting triple-time feature value sets, created by using the two best performing single time intervals (along with the time interval of six values, for labeling purposes), were then given to both decision tree and neural network classification systems.

5.1.5 Performance Evaluation

Final models were run on their respective test sets at varying thresholds for classifying cases as true alarm or no alarm. This resulted in sensitivity-specificity pairs used to plot corresponding ROC curves, from which the areas under the curves could then be calculated.

For comparison, we also developed ROC curves for upper and lower limit thresholding. Each signal needed two ROC curves: one to describe a high thresholding alarm, and the other to describe

```
            ECG        RESP    ART mmHg     ICP   SP02
  TIME     HR LEAD    RATE  SYS/DIA MEAN   mmHg    %
********  ********    ****  ************   ****   ****
14:24:54  124    I      31  133/ 58   76    10     100
14:25:00  121    I      17  139/ 89   91    24     100
14:25:06  151    I      31  146/ 56   92    20     100
14:25:11  151    I      53  136/ 60   90    14     100

            ECG        RESP    ART mmHg     ICP
  TIME     HR LEAD    RATE  SYS/DIA MEAN   mmHg
********  ********    ****  ************   ****
14:25:17  146    I      54  141/ 67   91    14
14:25:22  137    I      33  162/ 80   97    20
```

Figure 5-3: Sample monitored data from SpaceLabs monitor in the MICU.

a low thresholding alarm. For varying threshold values (approximately 10–15), for the entire range of values seen, we calculated the theoretical sensitivity and specificity values given a limit alarm set at each threshold. From this we could determine the area under the equivalent ROC curve. Cases used for thresholding were labeled based on the label of each raw value's neighborhood nine values; this results in labeling that is equivalent to that used for the 2-4-9 experiments. Separate ROC curves were also determined for thresholding on cases using six values for labeling; this gives labeling that is equivalent to that used for 6-12-24 and time interval experiments.

Where possible, developed true alarm detection models were additionally run on completely different data sets that had been collected by different trained observers under different hospital conditions, with different nurses and different patients.

5.2 Results

5.2.1 Annotated Data Collection

Over the twelve-week data collection period in 1996, approximately 585 hours of bedside signal values were recorded along with annotations of alarm and no-alarm periods. Only monitored data containing all five signals of interest (heart rate, oxygen saturation, respiratory rate, and mean and systolic blood pressure) were further used in this study. Figure 5-3 shows an example of actual monitor data from the SpaceLabs unit.

Table 5.1: Breakdown of MICU data sets by each signal type.

Signal	Training set	Evaluation set	Test set
Systolic BP	86,062	10,906	25,952
Oxygen Sat	86,165	10,918	25,981
Heart Rate	86,191	10,919	25,986
Mean BP	86,165	10,918	25,981
Resp. Rate	86,177	10,918	25,985

Table 5.2: Breakdown of MICU data by class label for each signal type.

Signal	True Alarm	Not Alarm
Systolic BP	1,550	121,350
Oxygen Sat	122	122,942
Heart Rate	63	123,033
Mean BP	73	122,991
Resp. Rate	28	123,052

5.2.2 Annotated Data Preprocessing

Data were preprocessed by the described methodology. For the systolic blood pressure signal alarms, there were 86,062 training cases, 25,952 test cases, and 10,906 evaluation cases, collectively consisting of 1550 true-alarm cases and 121,350 no-alarm cases. (The no-alarm cases included 2109 false-alarm cases.) Tables 5.1 and 5.2 list the analogous counts for each of the five data signals. The training and evaluation sets were then given to each of c4.5 and LNKnet.

5.2.3 Model Derivation

Signal Experiments: Multi-Signal versus Single-Signal

Multi-Signal Models We first present the true alarm detection models created with use of feature attributes derived from multiple monitored signals.

Systolic Blood Pressure Alarms The final 2-4-9 decision tree model chosen for detection of true alarms on the systolic blood pressure signal is shown in Figure 5-4. Class labels are represented by '1' for the true-alarm class and '0' for the no-alarm class. Parentheses after a class label indicate the number of training data cases which arrived at that node, followed by the number of training cases which were incorrectly classified at that node. Attribute names are a concatenation of the

```
sbp_avg9 <= 136.9 : 0 (71141.0/137.8)
sbp_avg9 > 136.9 :
|   mbp_avg9 <= 82.9 :
|   |   hr_med9 <= 104 : 1 (642.0/3.9)
|   |   hr_med9 > 104 : 0 (390.0/3.9)
|   mbp_avg9 > 82.9 :
|   |   mbp_low9 > 97 : 0 (11230.0/106.7)
|   |   mbp_low9 <= 97 :
|   |   |   hr_med9 <= 127 : 0 (1579.0/8.9)
|   |   |   hr_med9 > 127 :
|   |   |   |   sbp_high9 <= 149 : 0 (704.0/34.9)
|   |   |   |   sbp_high9 > 149 :
|   |   |   |   |   hr_avg9 <= 132.6 : 0 (91.0/18.9)
|   |   |   |   |   hr_avg9 > 132.6 :
|   |   |   |   |   |   rr_high9 <= 20 : 1 (226.0/66.3)
|   |   |   |   |   |   rr_high9 > 20 : 0 (59.0/22.6)
```

Figure 5-4: Systolic blood pressure true alarm detection (2-4-9) decision tree model.

abbreviation of the signal name, an abbreviation of the derived feature name, and the number of values over which the derived feature was calculated. For example, the first line in the decision tree model, "sbp_avg9 <= 136.9 : 0 (71141.0/137.8)," means: "if the average value over nine raw values of systolic blood pressure is less than or equal to 136.9, the case will be labeled no-alarm. During training, 71141.0 training cases arrived at this node and were labeled no-alarm; 137.8 of those cases were incorrectly labeled." (Fractional numbers of cases can arise due to pruning of the tree.) The final model was created with options c = 2% (meaning high levels of pruning), and m = 45 (meaning that a test node on the tree was only added if at least 45 cases were classified by one outcome branch of that node). The decision tree model achieved an area under the ROC curve of 94.35% when run on its test set; Figure 5-5 shows its ROC curve.

The final neural network model for 2-4-9 systolic blood pressure alarm detection contained 120 input nodes, one hidden layer with 30 nodes, and two output nodes. During network training, a step size of 0.2 was chosen as the amount by which network weights were to be updated during error propagation. The training process updated network weights during each of 20 cycles. (All LNKnet neural network models described in this chapter use a step size of 0.2, with 20 training cycles.) The model achieved an ROC curve area, shown in Figure 5-6, of 98.98% when run on its test set.

The final 6-12-24 decision tree model chosen (c = 5%, m = 30) for detection of true alarms on

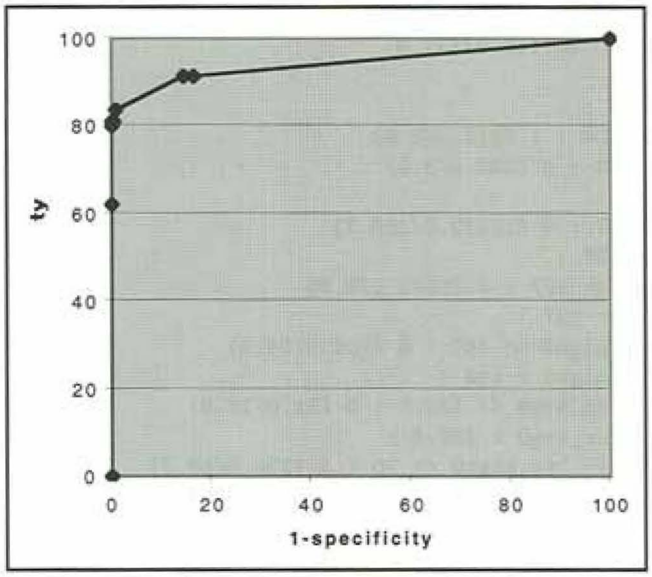


Figure 5-5: Systolic blood pressure true alarm detection (2-4-9) decision tree ROC curve. Area = 94.35%.

the systolic blood pressure signal is shown in Figure 5-7. It achieved an ROC curve area of 95.79%, shown in Figure 5-8.

The final 6-12-24 neural network model for systolic BP alarms contained 120 inputs, 30 hidden nodes in one layer, and 2 outputs. Its ROC curve, shown in Figure 5-9, had an area of 99.80%.

Oxygen Saturation Alarms The final 2-4-9 decision tree model chosen for detection of true alarms on the oxygen saturation signal is shown in Figure 5-10. The final model used c = 2% and m = 2. The decision tree model achieved an area under the ROC curve of 89.16% when run on its test set. Figure 5-11 shows the ROC curve.

The final neural network model for 2-4-9 oxygen saturation alarm detection contained 120 input nodes, one hidden layer with 30 nodes, and two output nodes. The model achieved an ROC curve area of 97.23% when run on its test set, as shown in Figure 5-12.

The final 6-12-24 decision tree for oxygen saturation alarms, shown in Figure 5-13, was developed with c = 15% and m = 20. Its ROC curve area was 80.00%.

The 6-12-24 neural network for detecting oxygen saturation alarms had 120 inputs, 30 hidden

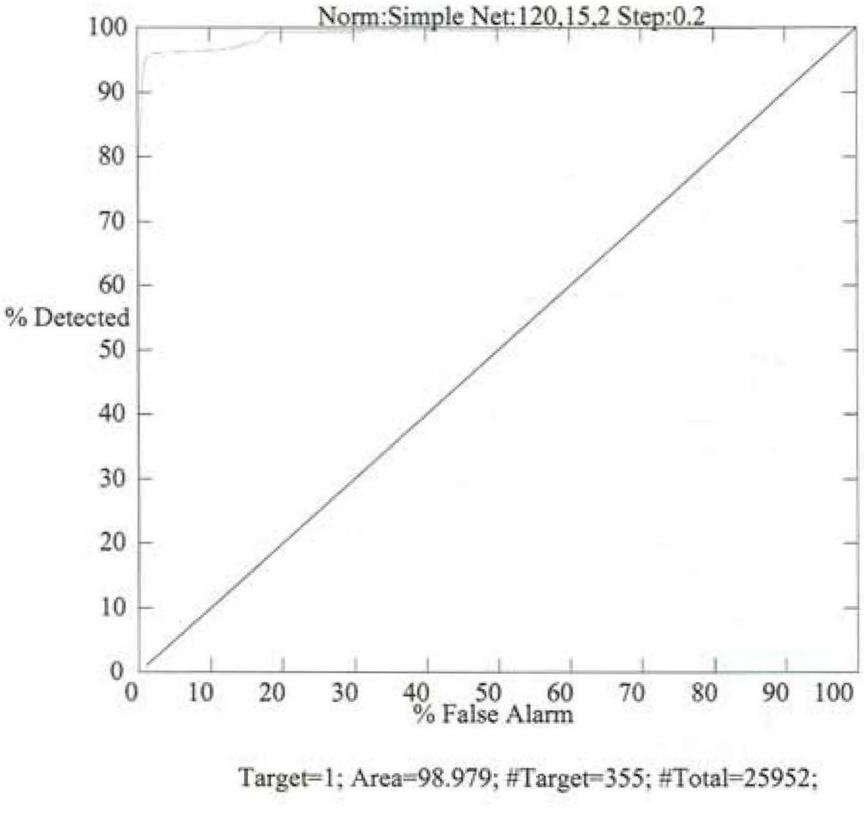


Figure 5-6: Systolic blood pressure true alarm detection (2-4-9) neural network ROC curve. Area = 98.98%.

```
sbp_avg24 <= 138.6 : 0 (71252.0/97.8)
sbp_avg24 > 138.6 :
|   mbp_high24 <= 84 :
|   |   rr_high12 <= 16 : 0 (280.0/3.0)
|   |   rr_high12 > 16 : 1 (632.0/3.0)
|   mbp_high24 > 84 :
|   |   rr_range24 <= 1 :
|   |   |   mbp_low24 <= 90 : 0 (801.0/14.8)
|   |   |   mbp_low24 > 90 :
|   |   |   |   hr_low24 <= 130 : 0 (169.0/3.0)
|   |   |   |   hr_low24 > 130 :
|   |   |   |   |   sbp_high12 <= 149 : 0 (83.0/10.5)
|   |   |   |   |   sbp_high12 > 149 :
|   |   |   |   |   |   mbp_avg24 > 96 : 1 (129.0/3.0)
|   |   |   |   |   |   mbp_avg24 <= 96 :
|   |   |   |   |   |   |   hr_slope24 <= -0.06 : 1 (31.0/2.9)
|   |   |   |   |   |   |   hr_slope24 > -0.06 : 0 (30.0/15.0)
|   |   rr_range24 > 1 :
|   |   |   sbp_range24 <= 48 : 0 (10597.0/28.3)
|   |   |   sbp_range24 > 48 :
|   |   |   |   hr_range24 <= 23 : 0 (179.0/3.0)
|   |   |   |   hr_range24 > 23 :
|   |   |   |   |   rr_high24 <= 44 : 0 (41.0/4.9)
|   |   |   |   |   rr_high24 > 44 : 1 (37.0/12.2)
```

Figure 5-7: Systolic blood pressure true alarm detection (6-12-24) decision tree model.

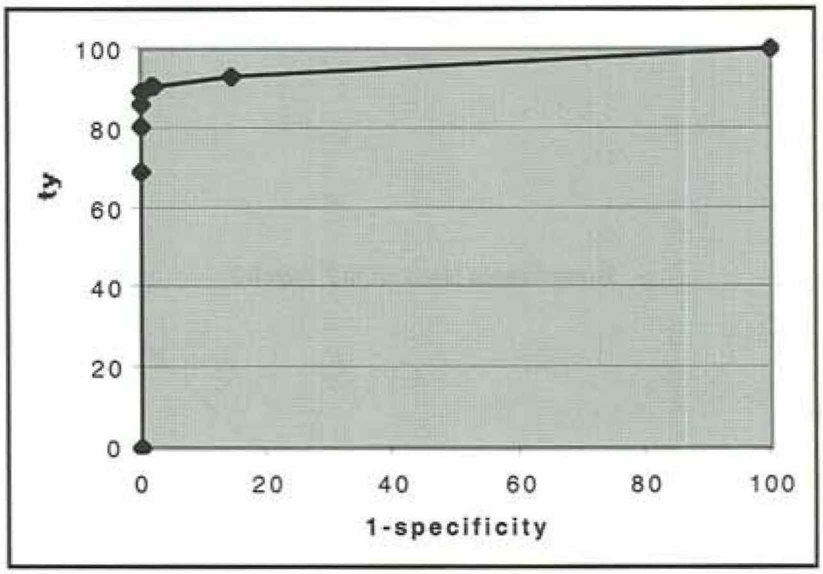


Figure 5-8: Systolic blood pressure true alarm detection (6-12-24) decision tree ROC curve. Area = 95.79%.

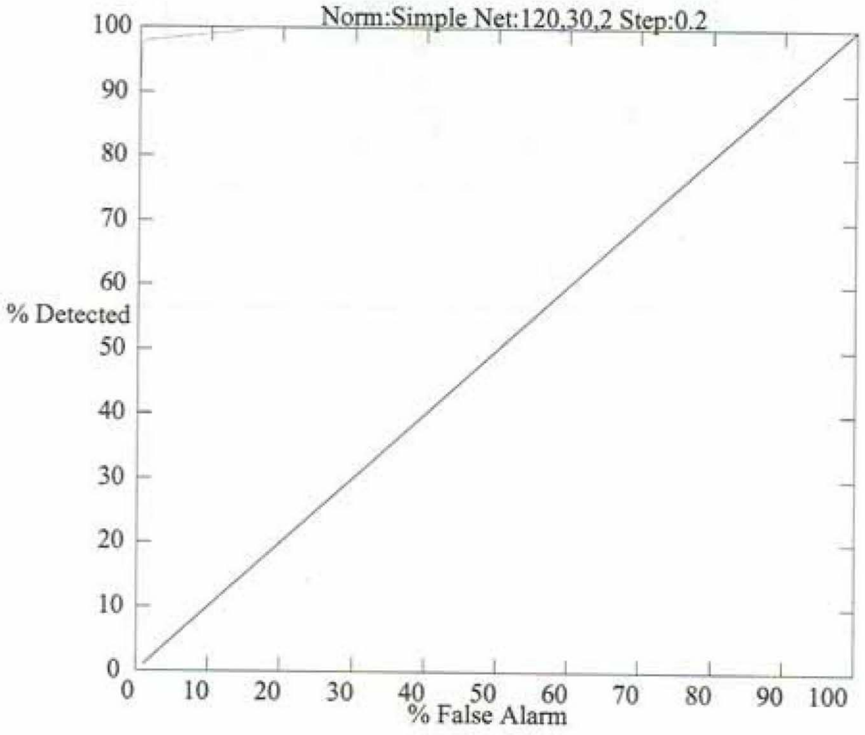
nodes in one layer, and 2 outputs. It achieved an area under the ROC curve of 90.49%, as shown in Figure 5-14.

Heart Rate Alarms The final 2-4-9 decision tree model chosen for detection of true alarms on the heart rate signal is shown in Figure 5-15. The final model used c = 15% and m = 20. The decision tree model achieved an area under the ROC curve of 99.72% when run on its test set. Figure 5-16 shows the ROC curve.

The final neural network model for 2-4-9 heart rate alarm detection contained 120 input nodes, one hidden layer with 30 nodes, and two output nodes. The model achieved an ROC curve area of 99.98% when run on its test set; this is shown in Figure 5-17.

Mean Blood Pressure Alarms The final 2-4-9 decision tree model chosen for detection of true alarms on the mean blood pressure signal is shown in Figure 5-18. The final model used c = 25% and m = 15. The decision tree model achieved an area under the ROC curve of 88.78%, as shown in Figure 5-19, when run on its test set.

The final neural network model for 2-4-9 mean blood pressure alarm detection contained 120



Target=1; Area=99.797; #Target=258; #Total=25434;

Figure 5-9: Systolic blood pressure true alarm detection (6-12-24) neural network ROC curve. Area = 99.80%.

```
ox_high9 <= 88 :
|   ox_low9 <= 86 :
|   |   hr_med9 <= 159 : 0 (699.0/11.9)
|   |   hr_med9 > 159 :
|   |   |   hr_high9 <= 165 : 0 (14.0/3.4)
|   |   |   hr_high9 > 165 : 1 (12.0/3.3)
|   ox_low9 > 86 :
|   |   hr_med9 <= 144 : 0 (66.0/3.8)
|   |   hr_med9 > 144 :
|   |   |   sbp_low9 <= 105 : 0 (5.0/2.7)
|   |   |   sbp_low9 > 105 : 1 (35.0/3.7)
ox_high9 > 88 :
|   ox_avg9 > 11.2 : 0 (85306.0/41.4)
|   ox_avg9 <= 11.2 :
|   |   hr_high9 <= 165 : 0 (21.0/3.6)
|   |   hr_high9 > 165 : 1 (7.0/3.0)
```

Figure 5-10: Oxygen saturation true alarm detection (2-4-9) decision tree model.

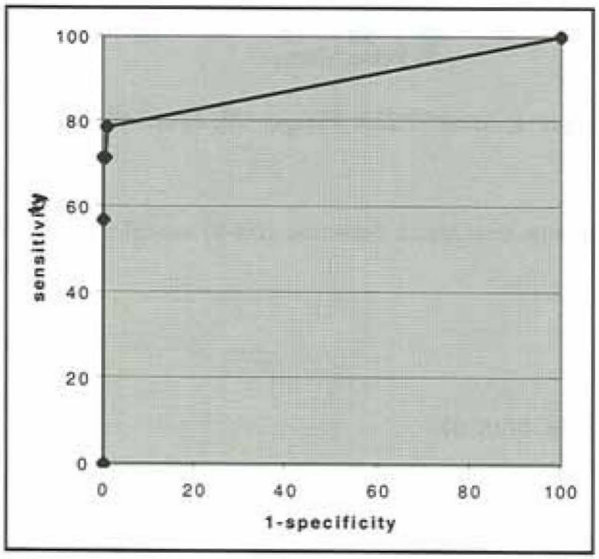


Figure 5-11: Oxygen saturation true alarm detection (2-4-9) decision tree ROC curve. Area = 89.16%.

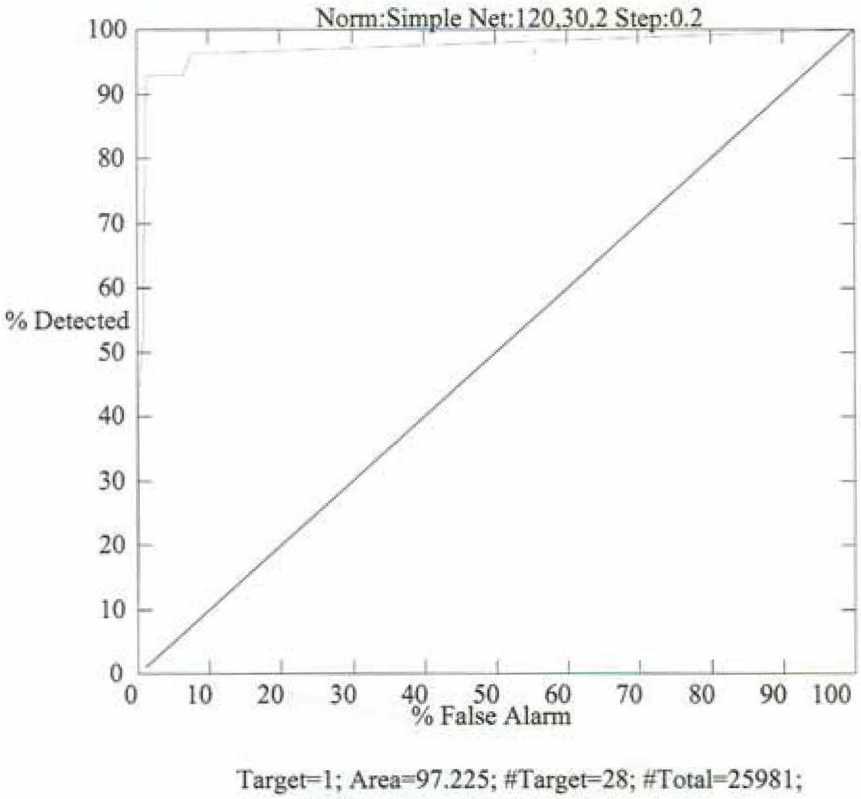
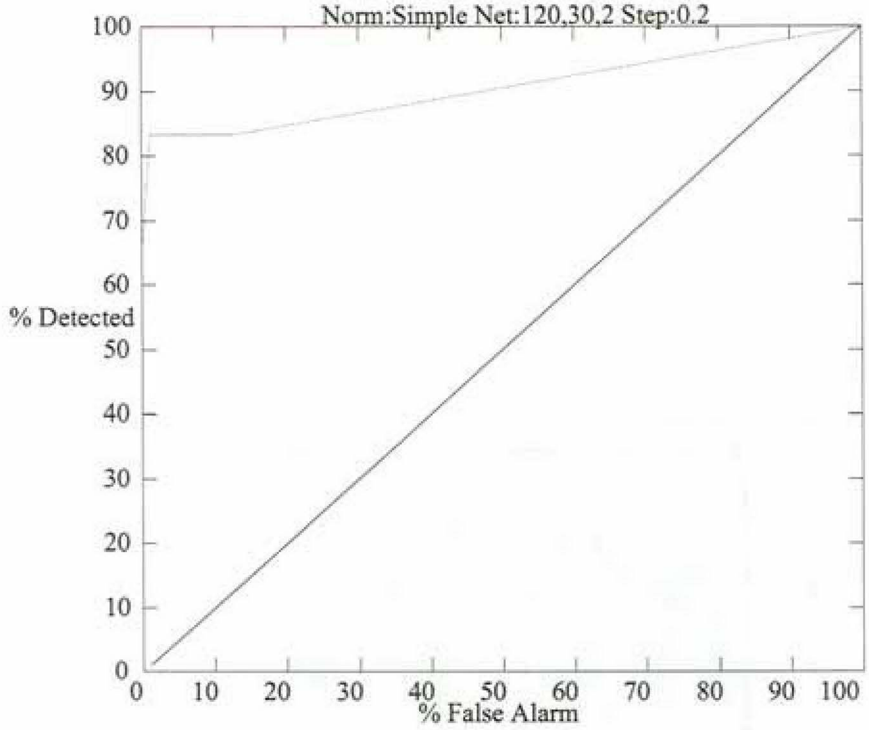


Figure 5-12: Oxygen saturation true alarm detection (2-4-9) neural network ROC curve. Area = 97.23%.

```
ox_high24 > 88 : 0 (83954.0/15.0)
ox_high24 <= 88 :
|   ox_low6 <= 83 : 0 (474.0)
|   ox_low6 > 83 :
|   |   hr_med24 <= 144.5 : 0 (56.0)
|   |   hr_med24 > 144.5 : 1 (30.0)
```

Figure 5-13: Oxygen saturation true alarm detection (6-12-24) decision tree model.



Target=1; Area=90.494; #Target=12; #Total=25499;

Figure 5-14: Oxygen saturation true alarm detection (6-12-24) neural network ROC curve. Area = 90.49%.

```
hr_low9 <= 168 :
|   sbp_med9 > 44 : 0 (83099.0)
|   sbp_med9 <= 44 :
|   |   mbp_low9 <= 30 : 0 (2369.0/5.0)
|   |   mbp_low9 > 30 :
|   |   |   rr_high9 <= 27 : 0 (52.0/1.0)
|   |   |   rr_high9 > 27 : 1 (33.0/11.0)
hr_low9 > 168 :
|   sbp_avg9 <= 116.1 : 1 (25.0/5.0)
|   sbp_avg9 > 116.1 : 0 (613.0)
```

Figure 5-15: Heart rate true alarm detection (2-4-9) decision tree model.

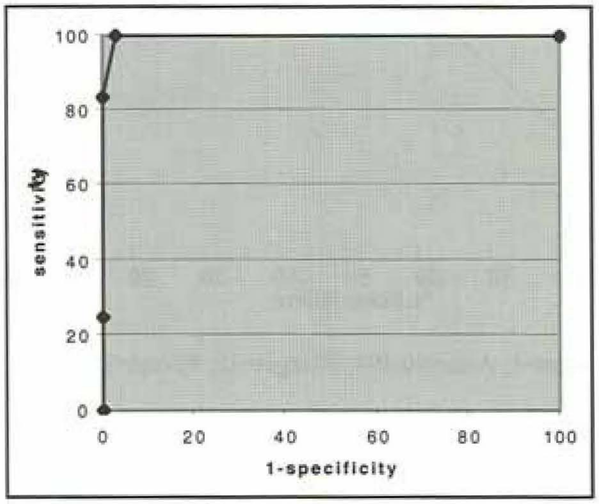


Figure 5-16: Heart rate true alarm detection (2-4-9) decision tree ROC curve. Area = 99.72%.

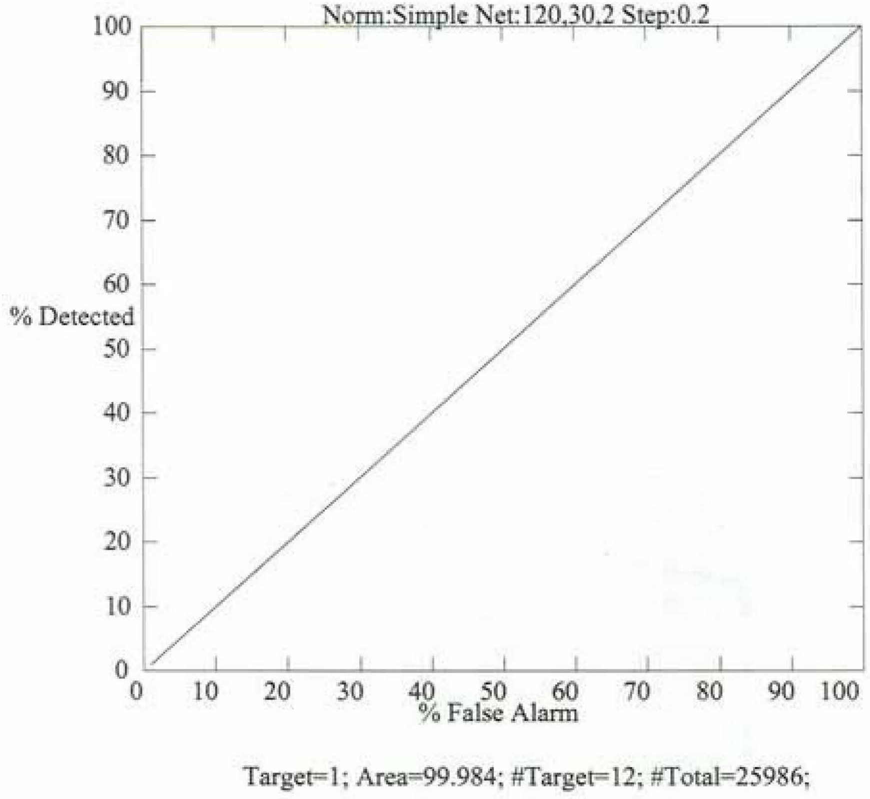


Figure 5-17: Heart rate true alarm detection (2-4-9) neural network ROC curve. Area = 99.98%.

```
mbp_low9 <= 158 :
|   rr_std_dev9 <= 21.66 :
|   |   sbp_low9 <= 151 : 0 (82505.0/20.6)
|   |   sbp_low9 > 151 :
|   |   |   sbp_high9 > 155 : 0 (2368.0/1.4)
|   |   |   sbp_high9 <= 155 :
|   |   |   |   rr_low2 <= 7 : 1 (15.0/3.7)
|   |   |   |   rr_low2 > 7 : 0 (15.0/1.3)
|   rr_std_dev9 > 21.66 :
|   |   mbp_low9 <= 41 : 1 (25.0/13.2)
|   |   mbp_low9 > 41 : 0 (1169.0/1.4)
mbp_low9 > 158 :
|   ox_high2 <= 99 : 1 (15.0/6.8)
|   ox_high2 > 99 : 0 (53.0/1.4)
```

Figure 5-18: Mean blood pressure true alarm detection (2-4-9) decision tree model.

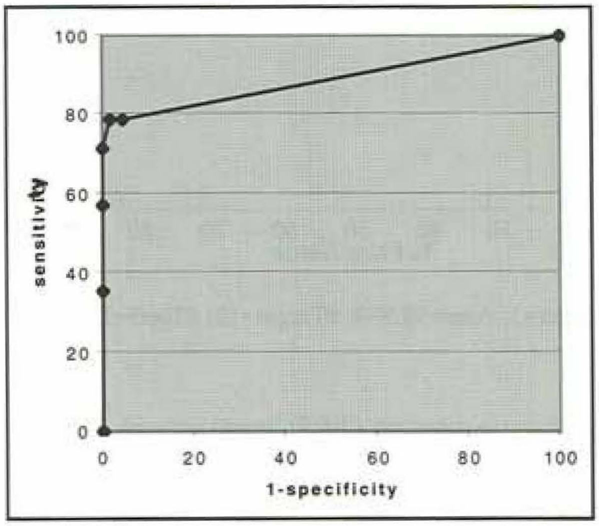
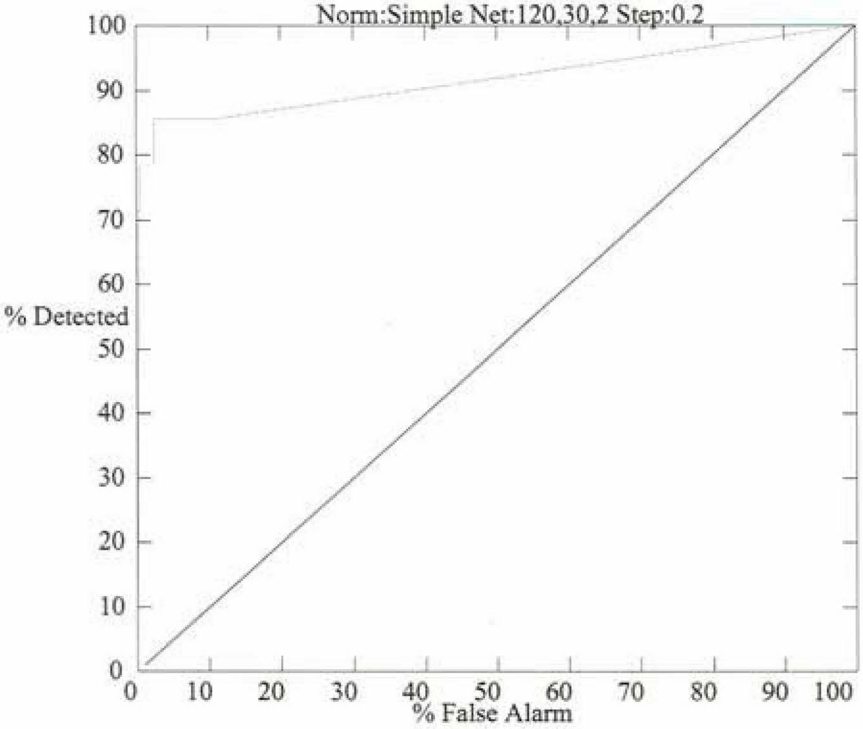


Figure 5-19: Mean blood pressure true alarm detection (2-4-9) decision tree ROC curve. Area = 88.78%.



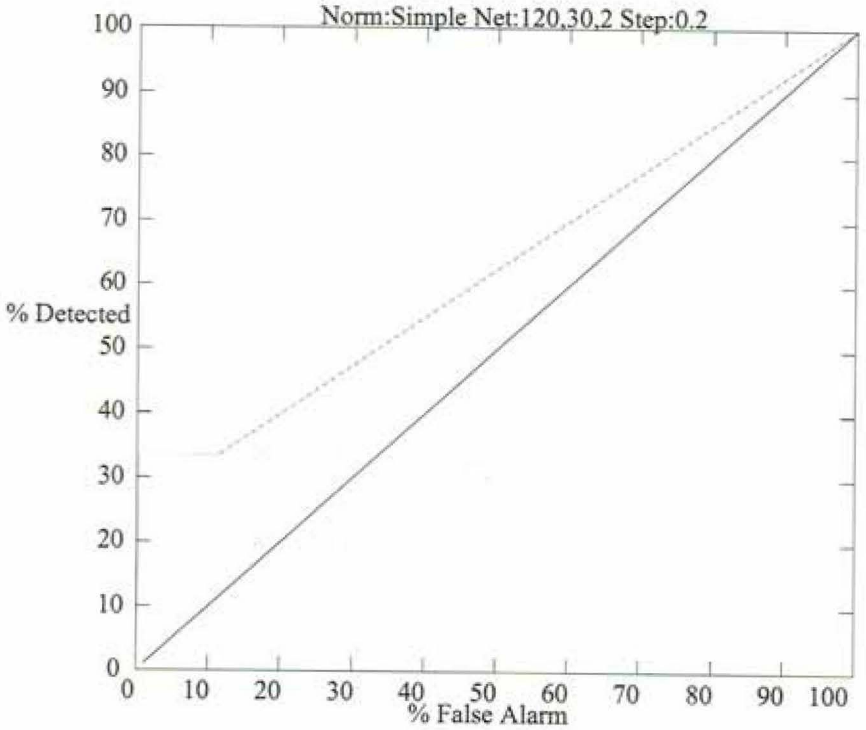
Target=1; Area=91.732; #Target=14; #Total=25981;

Figure 5-20: Mean blood pressure true alarm detection (2-4-9) neural network ROC curve. Area = 91.73%.

input nodes, one hidden layer with 30 nodes, and two output nodes. The model achieved an ROC curve area of 91.73% when run on its test set; this is depicted in Figure 5-20.

Respiratory Rate Alarms There were too few respiratory rate true alarms for c4.5 to be able to induce an appropriate tree. The derived decision tree model was simply one node: "not alarm."

The final neural network model for 2-4-9 respiratory rate alarm detection contained 120 input nodes, one hidden layer with 30 nodes, and two output nodes. The model achieved an ROC curve area of 62.86% when run on its test set. This ROC curve is shown in Figure 5-21.



Target=1; Area=62.858; #Target=6; #Total=25985;

Figure 5-21: Respiratory rate true alarm detection (2-4-9) neural network ROC curve. Area = 62.86%.

Single-Signal Models We then developed models that each only used feature attributes derived from one signal. Five final neural network models were selected, one each for detecting true alarms on a particular signal. The 2-4-9 single-signal systolic BP neural network had 24 input nodes, 20 hidden nodes in a single layer, and two output nodes. Figure 5-22 shows its ROC curve; it achieved an area under the ROC curve of 90.41%. The single-signal 2-4-9 oxygen saturation neural network had 24 inputs, a single hidden layer with 10 nodes, and two outputs. Figure 5-23 shows its ROC curve; it achieved an ROC area of 95.35%. The single-signal 2-4-9 heart rate network had 24 inputs, no hidden nodes, and two outputs. Figure 5-24 shows its ROC curve, which had an area of 87.76%. For mean BP alarm detection, the single-signal neural network had 24 inputs, 15 hidden nodes in a single layer, and two outputs; it achieved an area under the ROC curve of 70.06%, which is shown in Figure 5-25. Finally, for respiratory rate true alarms, the single-signal neural network had 24 inputs, a single hidden layer with 20 nodes, and two outputs. Figure 5-26 shows its ROC curve, which had an area of 31.21%.

Decision trees were also experimented with for the single-signal models. Resulting trees, however, were quite large and complicated; thus neural networks were preferred for this single-signal study.

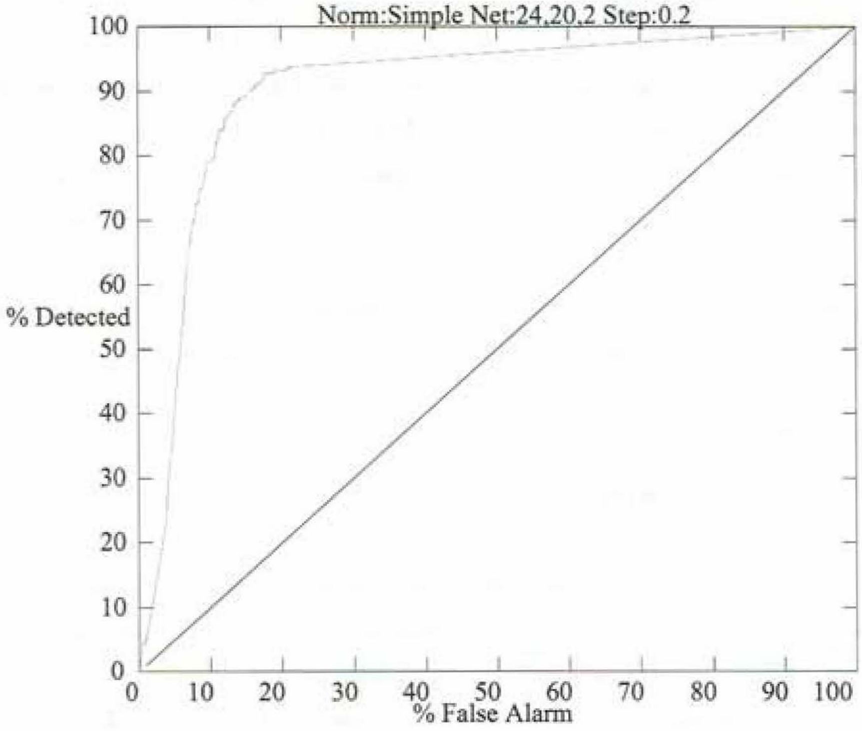
Time Interval Selection

Systolic Blood Pressure True Alarm Detection Tables 5.3 through 5.5 list the single-time systolic BP alarm detection decision trees; for each variation, tree size, number of errors, and percentage of errors, are reported. The two time intervals selected as best were 17 and 35.

The neural network developed from 6-17-35 feature attributes had 120 input nodes, 30 hidden nodes in a single layer, and two output nodes. It achieved an area under the ROC curve of 99.94%. The corresponding decision tree model (c = 5%, m = 65) is shown in Figure 5-27. The decision tree's ROC curve area was 91.97%.

Oxygen Saturation True Alarm Detection Tables 5.6 and 5.7 list the single-time oxygen saturation alarm decision trees; for each variation, the number of errors and percentage of errors are reported. The two time intervals selected as best were 16 and 23.

The neural network developed from 6-16-23 feature attributes had 120 input nodes, 15 hidden nodes in a single layer, and two output nodes. It achieved an area under the ROC curve of 99.96%. The corresponding decision tree model (c = 25%, m = 10) is shown in Figure 5-28. The decision tree's ROC curve area was 92.42%.



Target=1; Area=90.407; #Target=355; #Total=25952;

Figure 5-22: ROC curve for single-signal systolic blood pressure true alarm neural network model.

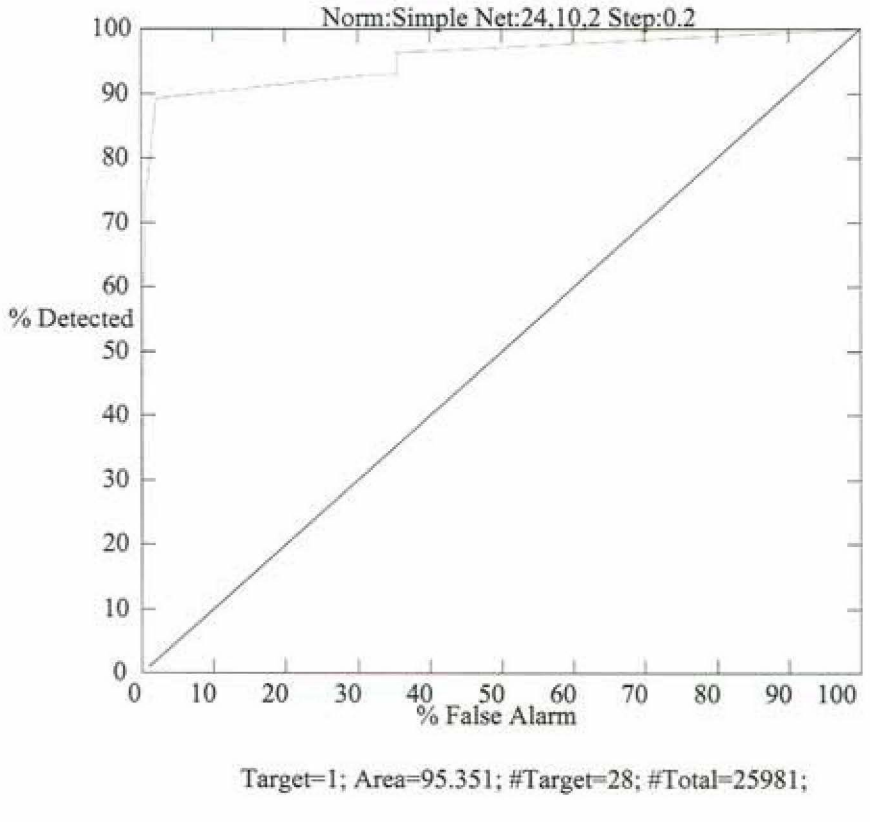
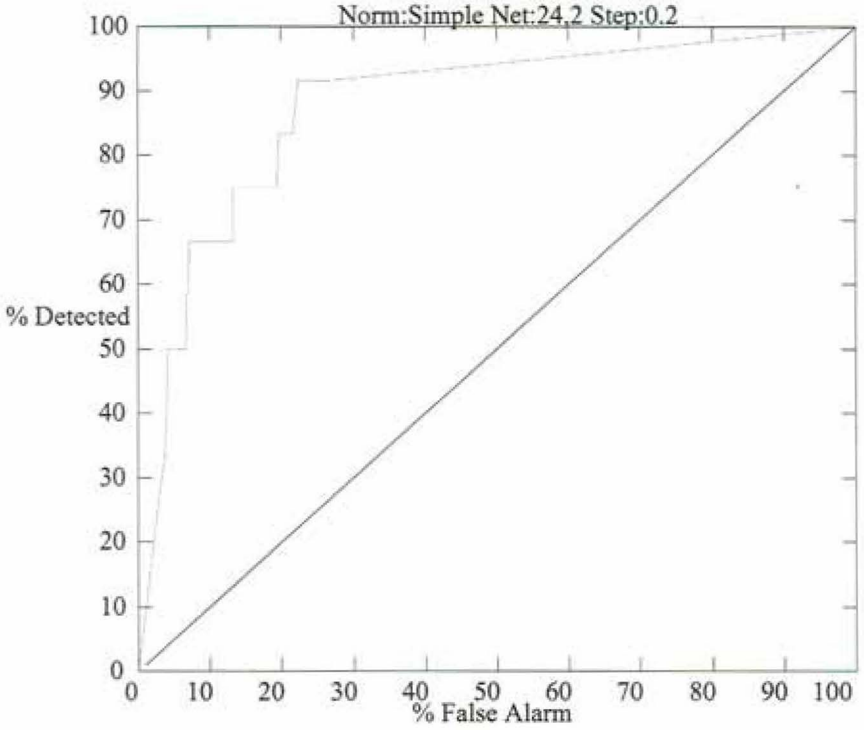
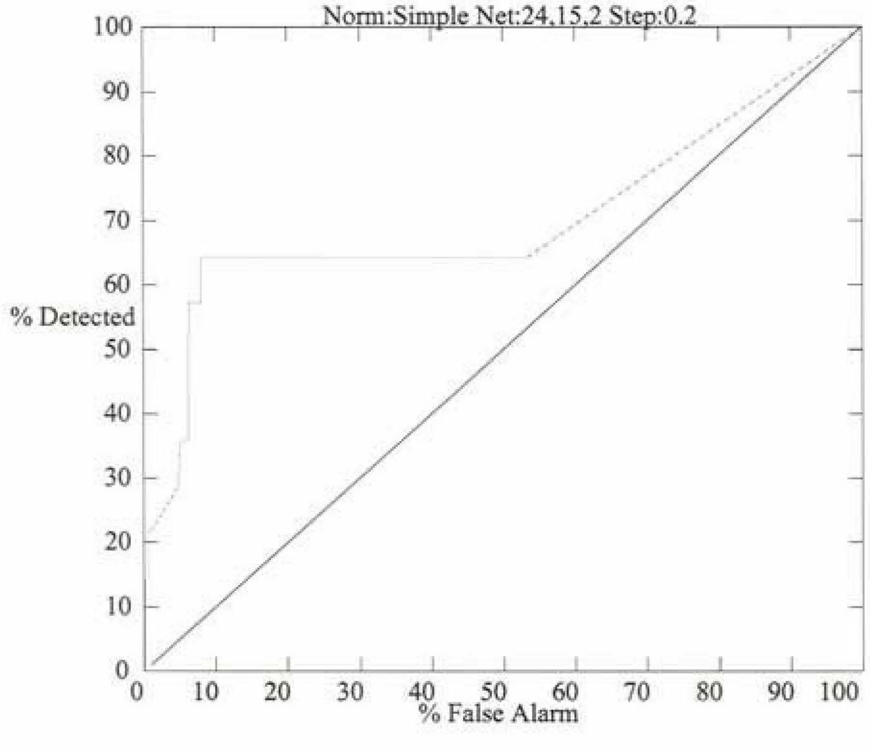


Figure 5-23: ROC curve for single-signal oxygen saturation true alarm neural network model.



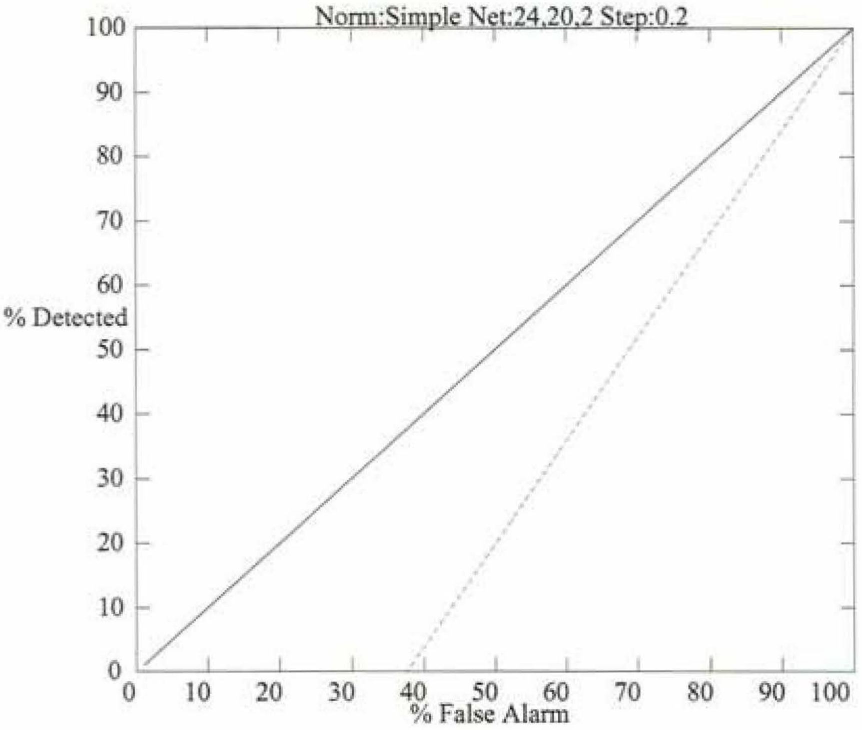
Target=1; Area=87.764; #Target=12; #Total=25986;

Figure 5-24: ROC curve for single-signal heart rate true alarm neural network model.



Target=1; Area=70.061; #Target=14; #Total=25981;

Figure 5-25: ROC curve for single-signal mean blood pressure true alarm neural network model.



Target=1; Area=31.210; #Target=6; #Total=25985;

Figure 5-26: ROC curve for single-signal respiratory rate true alarm neural network model.

Table 5.3: Single-time interval systolic BP alarm decision trees: c = 25%, m = 2.

Number of Values in Time Interval	Tree Size	Number of Errors	Percentage of Errors (%)
2	269	41	0.4
3	269	43	0.4
4	211	36	0.3
5	219	34	0.3
6	197	24	0.2
7	209	22	0.2
8	201	21	0.2
9	173	21	0.2
10	173	14	0.1
11	145	10	0.1
12	127	7	0.1
13	131	14	0.1
14	113	10	0.1
15	111	2	0.0
16	99	4	0.0
17	83	3	0.0
18	91	7	0.1
19	91	8	0.1
20	77	6	0.1
21	91	6	0.1
22	87	2	0.0
23	85	6	0.1
24	91	5	0.0
25	85	6	0.1
26	69	1	0.0
27	73	2	0.0
28	73	2	0.0
29	77	2	0.0
30	65	1	0.0
31	63	2	0.0
32	51	3	0.0
33	59	2	0.0
34	57	3	0.0
35	59	0	0.0
36	69	4	0.0

Table 5.4: Single-time interval systolic BP alarm decision trees: c = 25%, m = 10.

Number of Values in Time Interval	Tree Size	Number of Errors	Percentage of Errors (%)
2	105	42	0.4
3	101	45	0.4
4	113	44	0.3
5	99	36	0.3
6	125	33	0.3
7	97	30	0.3
8	93	29	0.3
9	85	24	0.2
10	67	26	0.2
11	71	13	0.1
12	69	17	0.2
13	91	12	0.1
14	65	21	0.2
15	73	14	0.1
16	85	9	0.1
17	73	5	0.0
18	69	12	0.1
19	73	8	0.1
20	67	5	0.0
21	57	9	0.1
22	69	6	0.1
23	67	11	0.1
24	71	7	0.1
25	61	7	0.1
26	57	4	0.0
27	51	5	0.0
28	49	7	0.1
29	59	4	0.0
30	43	5	0.0
31	51	7	0.1
32	43	2	0.0
33	43	3	0.0
34	43	3	0.0
35	39	3	0.0
36	45	7	0.1

Table 5.5: Single-time interval systolic BP alarm decision trees: c = 10%, m = 2.

Number of Values in Time Interval	Tree Size	Number of Errors	Percentage of Errors (%)
2	181	37	0.3
3	201	45	0.4
4	183	38	0.3
5	133	35	0.3
6	181	24	0.3
7	133	24	0.2
8	135	26	0.2
9	139	17	0.2
10	123	18	0.2
11	103	9	0.1
12	113	7	0.1
13	111	13	0.1
14	101	10	0.1
15	95	6	0.1
16	85	3	0.0
17	83	3	0.0
18	81	7	0.1
19	77	7	0.1
20	77	6	0.1
21	73	6	0.1
22	83	2	0.0
23	85	6	0.1
24	87	5	0.0
25	79	5	0.0
26	69	1	0.0
27	63	2	0.0
28	73	2	0.0
29	71	2	0.0
30	59	2	0.0
31	57	2	0.0
32	47	3	0.0
33	49	2	0.0
34	53	3	0.0
35	49	0	0.0
36	53	4	0.0

Table 5.6: Single-time interval oxygen saturation alarm decision trees: c = 25%, m = 2.

Number of Values in Time Interval	Number of Errors	Percentage of Errors (%)
2	6	0.1
3	9	0.1
4	10	0.1
5	5	0.0
6	4	0.0
7	3	0.0
8	3	0.0
9	5	0.0
10	3	0.0
11	4	0.0
12	2	0.0
13	0	0.0
14	3	0.0
15	1	0.0
16	0	0.0
17	0	0.0
18	0	0.0
19	0	0.0
20	0	0.0
21	1	0.0
22	0	0.0
23	0	0.0
24	1	0.0
25	1	0.0
26	1	0.0
27	0	0.0
28	0	0.0
29	2	0.0
30	0	0.0
31	1	0.0
32	0	0.0
33	1	0.0
34	0	0.0
35	0	0.0
36	2	0.0

Table 5.7: Single-time interval oxygen saturation alarm decision trees: c = 25%, m = 10.

Number of Values in Time Interval	Number of Errors	Percentage of Errors (%)
2	9	0.1
3	6	0.1
4	11	0.1
5	13	0.1
6	7	0.1
7	7	0.1
8	8	0.1
9	5	0.0
10	8	0.1
11	3	0.0
12	4	0.0
13	1	0.0
14	4	0.0
15	2	0.0
16	0	0.0
17	0	0.0
18	1	0.0
19	1	0.0
20	1	0.0
21	1	0.0
22	1	0.0
23	0	0.0
24	1	0.0
25	2	0.0
26	2	0.0
27	1	0.0
28	1	0.0
29	2	0.0
30	0	0.0
31	1	0.0
32	0	0.0
33	1	0.0
34	0	0.0
35	0	0.0
36	2	0.0

```
hr_high17 <= 98 :
|   hr_low35 <= 92 : 0 (6265.0/37.6)
|   hr_low35 > 92 :
|   |   sbp_high35 <= 136 : 0 (680.0/3.0)
|   |   sbp_high35 > 136 : 1 (640.0/3.0)
hr_high17 > 98 :
|   sbp_high17 <= 150 : 0 (69301.0/180.6)
|   sbp_high17 > 150 :
|   |   rr_range35 <= 1 :
|   |   |   hr_low35 <= 131 : 0 (397.0/3.0)
|   |   |   hr_low35 > 131 :
|   |   |   |   mbp_low35 <= 89 : 0 (81.0/40.9)
|   |   |   |   mbp_low35 > 89 : 1 (180.0/17.0)
|   |   rr_range35 > 1 :
|   |   |   mbp_low35 > 83 : 0 (5536.0/23.5)
|   |   |   mbp_low35 <= 83 :
|   |   |   |   rr_avg35 <= 31.3 : 0 (332.0/3.0)
|   |   |   |   rr_avg35 > 31.3 : 1 (65.0/30.1)
```

Figure 5-27: Systolic blood pressure true alarm detection (6-17-35) decision tree model.

5.2.4 ROC Curves for Threshold Limit Alarms

Label on Nine Values

For systolic BP high limit alarm detection, we used 14 threshold values to find 326 high alarms; the corresponding ROC curve area was 79.36%. For systolic BP low limit alarm detection, there were 19 low alarms with a thresholding ROC curve area of 54.50%.

For oxygen saturation alarms, thresholding achieved an ROC curve area of 91.79% on low limit alarms. There were no oxygen saturation high limit alarms.

There were 16 heart rate high limit alarms. The ROC curve for thresholding was 78.30%. There were no low limit heart rate alarms.

There were 12 high limit mean BP alarms and nine low limit mean BP alarms. ROC curve areas were 71.82% and 90.70% for high and low thresholding, respectively.

There were eight respiratory rate high limit alarms and four respiratory rate low limit alarms. Thresholding ROC curve areas were 54.06% and 52.85% for high and low thresholding, respectively.

```
ox_high23 <= 88 :
|   ox_low6 <= 83 : 0 (495.0/1.4)
|   ox_low6 > 83 :
|   |   ox_low16 <= 87 : 1 (44.0/1.4)
|   |   ox_low16 > 87 : 0 (57.0/1.4)
ox_high23 > 88 :
|   ox_high6 <= 87 :
|   |   mbp_abs_slope23 > 1.33 : 1 (16.0/7.9)
|   |   mbp_abs_slope23 <= 1.33 :
|   |   |   rr_avg23 <= 30.5 : 0 (276.0/1.4)
|   |   |   rr_avg23 > 30.5 :
|   |   |   |   ox_low6 <= 68 : 0 (15.0/1.3)
|   |   |   |   ox_low6 > 68 : 1 (10.0/3.5)
|   ox_high6 > 87 :
|   |   ox_slope16 <= -0.7 :
|   |   |   hr_slope23 <= 0.64 : 0 (555.0/1.4)
|   |   |   hr_slope23 > 0.64 :
|   |   |   |   sbp_high16 <= 157 : 1 (14.0/3.6)
|   |   |   |   sbp_high16 > 157 : 0 (12.0/1.3)
|   |   ox_slope16 > -0.7 :
|   |   |   hr_med23 > 86 : 0 (78976.0/5.1)
|   |   |   hr_med23 <= 86 :
|   |   |   |   ox_low16 <= 91 :
|   |   |   |   |   mbp_range23 <= 13 : 0 (46.0/1.4)
|   |   |   |   |   mbp_range23 > 13 : 1 (10.0/4.6)
|   |   |   |   ox_low16 > 91 :
|   |   |   |   |   sbp_slope16 <= 0.28 : 0 (3792.0/1.4)
|   |   |   |   |   sbp_slope16 > 0.28 :
|   |   |   |   |   |   rr_std_dev16 > 1.38 : 0 (318.0/2.6)
|   |   |   |   |   |   rr_std_dev16 <= 1.38 :
|   |   |   |   |   |   |   hr_std_dev6 <= 5.53 : 0 (42.0/1.4)
|   |   |   |   |   |   |   hr_std_dev6 > 5.53 : 1 (10.0/3.5)
```

Figure 5-28: Oxygen saturation true alarm detection (6-16-23) decision tree model.

Label on Six Values

Thresholding ROC curve areas with labeling on six values were only determined for systolic BP and oxygen saturation alarms since those are the useful values for comparison with time interval experiments.

Systolic BP high limit thresholding achieved an ROC curve area of 80.93%, while low limit thresholding achieved an ROC curve area of 54.30%.

There were no high limit oxygen saturation alarms. Oxygen saturation low limit thresholding achieved an ROC curve area of 85.43%.

5.2.5 Performance on Different Data Sets

To evaluate model performance on completely different data, we looked to data collected in 1995 and 1997. (Recall that our MICU experiments use 1996 data.) The trained observer was a different person from year to year, the patients were different, and some of the medical staff had changed. Our focus is mainly on testing the systolic BP, oxygen saturation, and heart rate alarm detection models because the other signals had too few true alarms for effective learning.

Data collected during the ten weeks of 1995, during the prospective alarm study described in Chapter 3, were only partially useful. Although the five required signals were present for at least part of the recorded data, there were no annotations of systolic blood pressure alarms. Oxygen saturation alarms were present in the annotations; however, there was a problem with the pulse oximeter connection in 1995 such that "bad data format/bad connection" caused 1,136 false alarms, or 39% of all alarms recorded (true and false alarms). This problem did not exist in 1996, which is the data we used for development of the oxygen saturation true alarm detection model. Nevertheless, we ran the oxygen saturation and heart rate alarm detection models on these data.

There were no recorded clinically-relevant true heart rate alarms. The 15,007 non-alarm cases were correctly classified by the 2-4-9 heart rate alarm decision tree model at both 50% and 80% thresholds (chosen arbitrarily). The heart rate model therefore had 100% accuracy.

In the preprocessed data, there were 14,922 non-alarm and 53 true alarm oxygen saturation cases. The 6-16-23 oxygen saturation alarm decision tree model's area under the ROC curve was 62.56%. The 6-16-23 oxygen saturation alarm neural network model achieved an ROC curve area of 67.13%.

Data collected during 1997 occurred over six weeks. Alarms sounding on any of the five signals of interest were annotated. Upon retrospective analysis, however, it became evident that none

Table 5.8: ROC curve areas for single-signal neural networks, multi-signal neural networks, and upper and lower limit thresholding.

True Alarm Signal Type	Single-Signal Neural Network	Multi-Signal Neural Network	Upper Limit Thresholding	Lower Limit Thresholding
Systolic BP	90.41%	98.98%	79.36%	54.50%
O_2 Sat	95.35%	97.23%	(none)	91.79%
Heart Rate	87.76%	99.98%	78.30%	(none)
Mean BP	70.06%	91.73%	71.82%	90.70%
Resp. Rate	31.21%	62.86%	54.06%	52.85%

of the patients during 1997 had any pulse oximetry signal data recorded into the laptop. It is not clear whether none of the patients monitored during our data collection had pulse oximetry in use; whether there had been a monitor hardware or software problem; or whether an oversight in connection of the pulse oximeter to the SpaceLabs monitor had occurred. Because there were no oxygen saturation data, none of the neural network models could be run on these data. Note, however, that the systolic blood pressure alarm decision tree model, shown in Figure 5-27, contains no oxygen saturation attribute nodes. We therefore ran the 6-17-35 systolic BP tree model on the 1997 data. This model achieved an area under the ROC curve of 78.12%.

5.3 Discussion

This chapter has explored several aspects of applying the TrendFinder event discovery paradigm to the problem of detecting true alarms in the ICU. We now discuss the following issues: single-signal versus multi-signal detection models; time interval selection; post-model refinement for new populations; models for detecting true alarms versus false alarms; and limitations of the study.

5.3.1 Single-Signal Versus Multi-Signal Methods

Table 5.8 shows the ROC curve areas for 1-2-9 single-signal neural network true alarm detection models, 1-2-9 multi-signal neural network alarm models, and upper and lower limit thresholding methods (labeled also on nine values). Figure 5-29 illustrates the performance of multi-signal neural networks, upper limit thresholding, and lower limit thresholding.

In five out of five true alarm detection models, multi-signal models out-performed single-signal models. The mean BP lower limit thresholding ROC curve area was quite good (90.70%), only

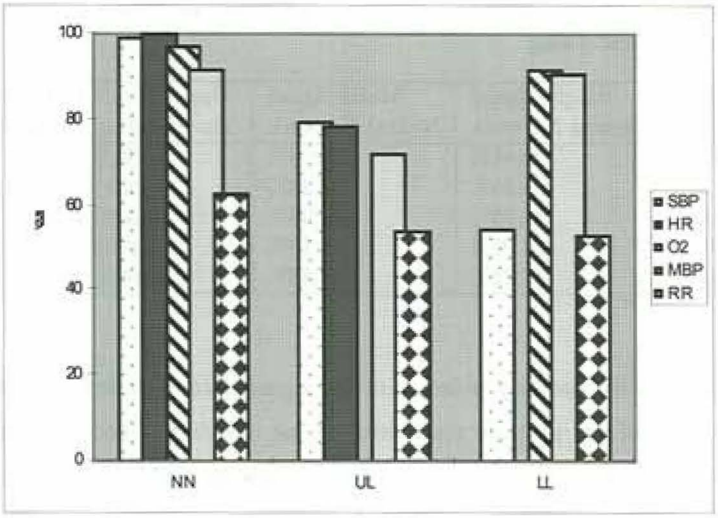


Figure 5-29: Performance of multi-signal neural networks (NN), upper limit thresholding (UL), and lower limit thresholding (LL) on detection of true alarms for each of the five signal types. SBP: gray dots on white; HR: solid dark gray; O2: black stripes on white; MBP: solid light gray; RR: black diamonds on white.

slightly worse than the mean BP multi-signal neural network model (91.73%). In all other cases, multi-signal neural network models clearly out-performed limit thresholding methods.

5.3.2 Time Interval Selection

We presented in Section 5.1.3 a method for choosing time intervals over which to derive feature attributes. Table 5.9 summarizes the results.

We see that in both cases, the models developed based on a principled manner of time interval selection did extremely well. In the case of systolic BP alarm detection, the model created from

Table 5.9: Comparison of ROC curve areas of models built from arbitrary time intervals vs. best times determined by time interval experiments.

True Alarm	6-12-24	6-17-35	6-16-23
Signal Type	Neural Network	Neural Network	Neural Network
Systolic BP	99.80%	99.94%	-
O_2 Sat	90.49%	-	99.96%

arbitrarily chosen time intervals did equally well. In the case of oxygen saturation alarm detection, the model created from arbitrarily chosen time intervals did significantly poorer.

The systolic BP 6-17-35 decision tree model (ROC area 91.97%) did not do as well as the 6-12-24 decision tree model (ROC area 95.79%), but neither of these decision tree models performed as well as their counterpart neural networks. The oxygen saturation 6-16-23 decision tree model (ROC area 92.42%) far out-performed this 6-12-24 decision tree model (ROC area 80.00%). Again, though, since neither of these decision tree models performed as well as their counterpart neural networks, we do not attempt to draw conclusions from these decision tree observations.

5.3.3 Post-Model Threshold Refinement for New Populations

Both the decision tree and neural network models developed for systolic BP and oxygen saturation true alarm detection performed well on their test sets. However, the two models evaluated on 1995 and 1997 data (oxygen saturation alarm neural network and systolic BP decision tree, respectively) both performed significantly worse.

For several reasons, the 1995 data used to test the oxygen saturation true alarm model may have been significantly different than the 1996 data used for development. First, as already discussed in Section 5.2.5, there had been a problem with the pulse oximeter connection in 1995 such that "bad data format/bad connection" caused 1,136 false alarms, equal to 39% of all recorded true and false alarms that year. This problem had been fixed by the time data were collected in 1996, which were the data subsequently used for model development. Second, differences in data annotation due to different trained observers, verbally-verifying nurses, or inconsistent recordings may have been great enough to warrant the poor performance of our neural network model on 1995 data. And finally, differences in patient population monitored for our study may have been great enough to account for the lower performance. We discuss one approach, 'post-model refinement' of threshold values to take into account new populations, in order to explore the third possibility. This will be illustrated with the systolic BP alarm detection decision tree model.

One reason that the systolic BP model may not have performed as well is that 1997 data contained no pulse oximetry (arterial oxygen saturation) data. It may have been that this signal was quite important for alarm detection despite its absence from the decision tree model. In fact, the counterpart neural network model had out-performed the decision tree by almost 8% area under the ROC curve (99.94% versus 91.97%). Others have suggested methods for handling missing attribute values [116, 158, 207]; this is clearly an important research area.

Differences in data annotation between 1996 and 1997 could also have accounted for the decreased performance seen when the systolic BP true alarm decision tree model was run on 1997 data. Likewise, differences in patient population, as discussed for oxygen saturation alarm detection, could also have caused the decreased performance.

Patient population in the multidisciplinary medical ICU at Children's Hospital is quite varied. The type of patients can range from babies to teenagers; clearly, these groups have physiological differences. As described in Chapter 4, when we ran our neonatal ICU artifact detection models on completely unseen data, they performed extremely well, with no decrease in performance compared to that of the original test sets. That may have been because the patient populations were neonates in both cases.

Just as traditional threshold limit alarms in the ICU have threshold levels that can be adjusted by the caregiver for a particular patient, we propose that machine-learned models could also have the flexibility of 'threshold refinement,' or adjustment, for a particular patient, i.e., 'post-model threshold refinement.'

To explore this possibility, we used 9% of the 1997 data to determine which threshold levels (for signal tests at each node in the systolic BP decision tree) might be more suited for the 'new patient.' The 'adapted' tree used six adjusted threshold levels and three original threshold levels. We otherwise kept the identical decision tree structure of the original 6-17-35 systolic BP true alarm detection model. The original tree was shown in Figure 5-27. The adapted tree is shown in Figure 5-30.

The adapted decision tree model achieved an area under the ROC curve of 97.36% on 1997 data. Recall that pre-adaptation, the original decision tree model's performance on 1997 data was 78.12%. We then took the adapted decision tree and ran it on the 1996 test set; it achieved an 85.71% ROC curve area. Figure 5-31 summarizes these results. Post-model refinement of thresholds may be one method for increasing the robustness of models originally developed from a more limited amount of annotated data when additional annotated data become available.

A different approach to tackling different patient populations or for using models developed originally for one patient population on a different patient population may be to generate models that from the start take into account both information about a patient and sensor data; Rosset *et al.* suggested a similar technique of looking at both customer information and credit card transaction history for fraud analysis [174]. In any case, after one or more alarm algorithms have been developed and show promise in retrospective analysis, prospective trials need to be performed to better

```
hr_high17 <= 98 :
|   hr_low35 <= 92 : 0
|   hr_low35 > 92 :
|   |   sbp_high35 <= 136 : 0
|   |   sbp_high35 > 136 : 1
hr_high17 > 98 :
|   sbp_high17 <= 117 : 0
|   sbp_high17 > 117 :
|   |   rr_range35 <= 24 :
|   |   |   hr_low35 <= 149 : 0
|   |   |   hr_low35 > 149 :
|   |   |   |   mbp_low35 <= 125 : 0
|   |   |   |   mbp_low35 > 125 : 1
|   |   rr_range35 > 24 :
|   |   |   mbp_low35 > 101 : 0
|   |   |   mbp_low35 <= 101 :
|   |   |   |   rr_avg35 <= 18.2 : 0
|   |   |   |   rr_avg35 > 18.2 : 1
```

Figure 5-30: Systolic blood pressure true alarm detection (6-17-35): decision tree model with six adapted thresholds using 'post-model threshold refinement.'

determine clinical utility. The data collection and alarm annotation system from the earlier phase of this project is in the process of being extended to perform, in addition to data collection, real-time processing of that data with candidate alarm algorithms [230].

5.3.4 Detecting True Alarms Versus False Alarms

Traditional committee classifiers use more than one model, each developed to classify the data of interest, for example, as Class A or Class B [191, 202]. We propose an alternative 'committee' classifier method for our particular target application in the ICU. Recall from the details of Chapter 3 that not only were clinically-relevant true alarms recorded, but also false alarms were recorded. Up to this point, all models in this chapter have been developed to detect clinically-relevant true alarms.

An idea is to also develop models specifically aimed at detecting ICU false alarms. That is, in the TrendFinder paradigm, our event of interest becomes the false alarms on a particular data signal. Observe in Figure 5-32 that true alarms and false alarms are non-overlapping subsets of all cases. 'No alarm' denotes the rest of the space in the Venn diagram. When searching specifically for

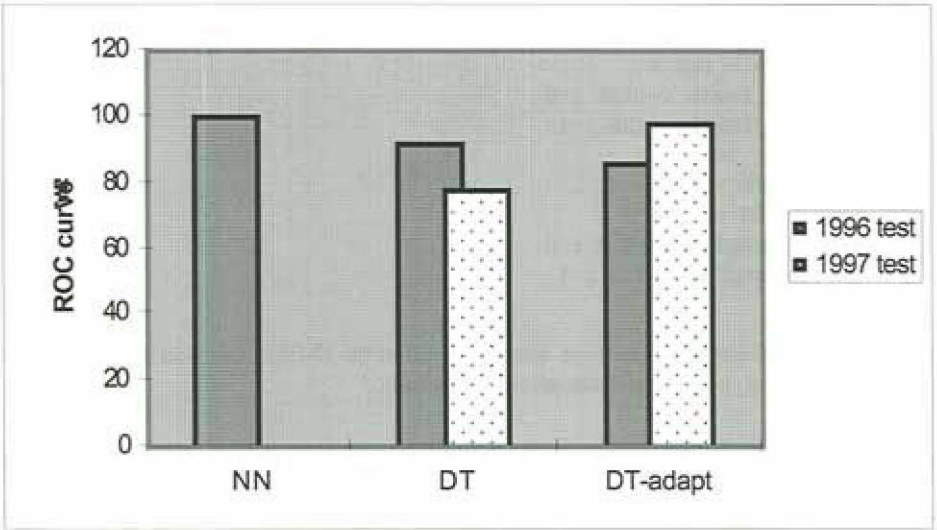


Figure 5-31: Performance of neural network model on 1996 data and performance of original and adapted decision tree models on 1996 and 1997 data. NN = neural network, DT = original decision tree, DT-adapt = adapted decision tree. 1996 test data: solid gray; 1997 test data: gray dots on white.

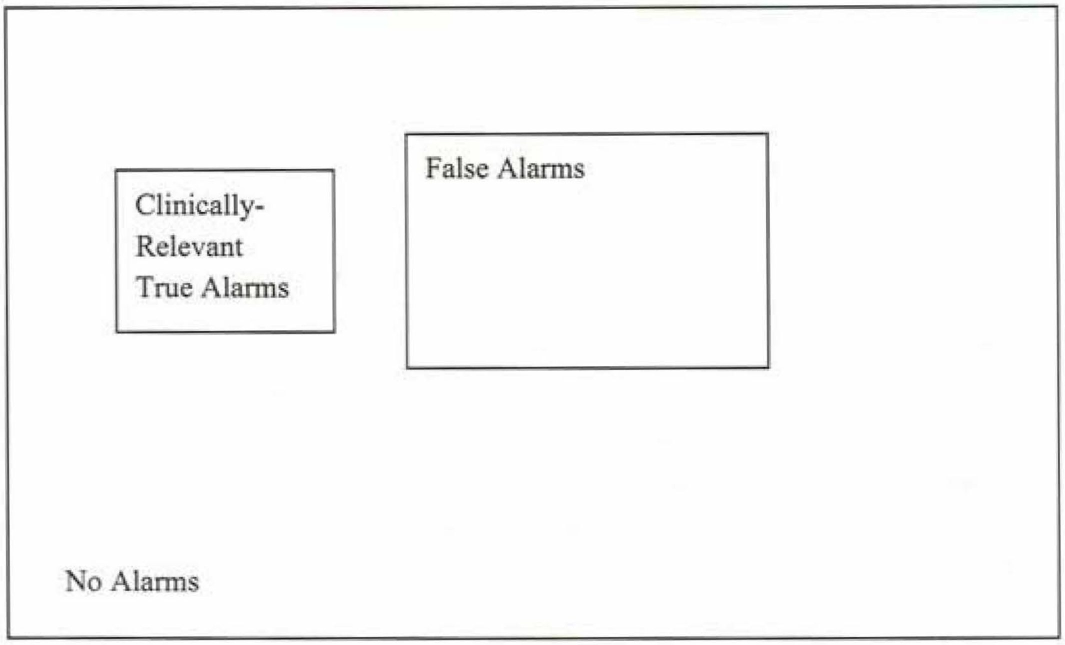


Figure 5-32: Venn diagram of ICU alarm categories.

true alarms, the 'rest of the world' consists of exactly false alarms plus non-alarms. When searching specifically for false alarms, the 'rest of the world' consists of exactly true alarms plus non-alarms. We can then have a system composed of one or more models that detect true alarms and one or more models that detect false alarms. The goal (other than having a perfect classifier of true alarms that has high sensitivity and high specificity) is to have a classifier that detects false alarms with high specificity and a classifier that detects true alarms with high sensitivity. The fundamental idea is then to first apply a true alarm detection model that finds all true alarms but also might label a few non-alarms as true alarms (i.e., false alarms), and then 'subtract' from the labeled true alarms those cases which the false alarm classifier labels with high specificity as false alarm.

Experimentation can help to refine the committee interaction amongst these classifiers. For example, we might have two true alarm classifiers and two false alarm classifiers. Depending on how well the true alarm classifiers perform, we might make a committee classifier that labels cases as true alarms when both true alarm classifiers agree; when they disagree, however, then the case in question is only called true alarm if both false alarm classifiers do not say 'false alarm.'

Table 5.10: ROC curve areas for three true alarm detection models and three false alarm detection models for systolic blood pressure.

Alarm Type	Neural Network Model	Decision Tree Model	Radial Basis Function Model
True Alarms	98.98%	99.32%	98.82%
False Alarms	96.56%	97.40%	92.12%

We explored implementation of such a hybrid system classifier by developing three true alarm detection models and three false alarm detection models for systolic blood pressure alarms. A decision tree, a neural network, and a radial basis function network were developed using the LNKnet system for each of true and false alarm detection on this signal. The performance of each classifier working alone is summarized in Table 5.10.

Each of these six models was then converted into computer-executable code in the C language using LNKnet code generation functionality. These models could then be used by other C language programs, which we wrote. We first examined the sensitivity and specificity achieved by the true alarm model having the highest ROC curve area–namely, the decision tree model. When run on the test set, the true alarm detection decision tree model had a sensitivity of 92.5% and a specificity of 99.1%. We then tried several potential hybrid true alarm-false alarm committee classification systems and noted each system's resultant sensitivity and specificity. An example of a hybrid system is shown in pseudocode in Figure 5-33. This hybrid system achieved an almost equal specificity of 99.0% and a higher sensitivity of 95.5%.

Future work might first focus on developing more accurate true and false alarm classifiers. One approach would be to look for false alarms due to a specific cause, instead of looking for false alarms due to all causes. Recall that Table 3.9 listed many different causes of false alarms observed in the ICU. Several false alarm detection models could then be developed, each one for finding just one type of false alarm. An approach for developing more accurate true alarm classifiers might be to develop one model to only detect true alarms that are high-valued alarms, and another model to only detect true alarms that are low-valued alarms. Recall that very different results were achieved by limit thresholding for high and low alarms; this could indicate that the two groups of alarms are sufficiently different to warrant individual models. In addition, as for false alarms, individual models could be developed to detect true alarms due to a particular cause. Each model could moreover be associated with a different sense of urgency for its alarm.

```
if   NN_TPR = TRUE_ALARM
     or RBF_TPR = TRUE_ALARM
then if   NN_TPR = TRUE_ALARM
          and RBF_TPR = TRUE_ALARM
     then Class = TRUE_ALARM
     else if   NN_FP = FALSE_ALARM
               and TREE_FP = FALSE_ALARM
               and RBF_FP = FALSE_ALARM
       and TREE_TPR = NO_ALARM
          then Class = NO_ALARM
          else if   NN_TPR = TRUE_ALARM
                    and TREE_TPR = TRUE_ALARM
               then Class = TRUE_ALARM
               else Class = NO_ALARM
else Class = NO_ALARM
```

```
where:

NN = neural network
TREE = decision tree
RBF = radial basis function
TPR = (model to detect) true positive alarms, clinically relevant
FP = (model to detect) false positive alarms
```

Figure 5-33: Pseudocode for hybrid true alarm-false alarm committee classifier.

5.3.5 Limitations

This ICU case study is not without limitations. First, the data collected were only available at a frequency of once per five seconds. This limits our choice in selecting appropriate time intervals for feature derivation; for example, a feature attribute derived from a time interval that is not a multiple of five seconds may in fact be the most accurate predictor for an event. The infrequency of data values from which to learn alarm patterns also may pose a problem if higher frequencies of data are later available when these models are tested prospectively, although this was surprisingly not the case in the neonatal ICU study (Chapter 4).

Another limitation of the case study was that annotations were only recorded to the nearest minute, while raw data values were collected every five seconds. This difference mandates that we will incorrectly label some cases simply because we are not sure precisely when, during the recorded minute, the true alarm actually occurred. For this study, we tried to compensate for the granularity differences by using a time 'window' of one minute on either side of each annotation's recorded time of occurrence. Areas of overlapping windows were then not used for machine learning. Section 6.2.1 also discusses windowing compensation methodology for inadequate annotations. Future work in this area should pay caution to recording annotations with the same time granularity as available raw data.

Annotations are additionally subject to inter-observer and intra-observer biases. For the described ICU case study, two trained observers recorded all of the 1996 annotations, a third trained observer recorded all of the 1995 annotations, and a fourth trained observer recorded all of the 1997 annotations. For each annotation, the bedside nurse who was present that day, in that bedspace, validated the annotation. It is not possible that all of the MICU nurses and all four trained observers interpreted or recorded all alarm occurrences in the same manner (inter-observer bias). Moreover, the same nurse or the same trained observer would also likely not record every type of alarm occurrence in the same manner (intra-observer bias). One possibility for future annotation methodology is to explore the idea of being able to record and recognize scenarios in the ICU [49].

Finally, there were generally not enough numbers of event class examples for adequate supervised machine-learning. This was especially true for all signals except systolic blood pressure. Even for systolic blood pressure, the number of clinically-relevant true alarms was quite low given the quantity of data. Figures 5-34 through 5-37 show scatter plots of MICU data attributes, two at a time, for systolic blood pressure true alarm detection. The attributes chosen for these plots are attributes that are present in the systolic BP true alarm decision tree model, which was shown in Figure 5-4.

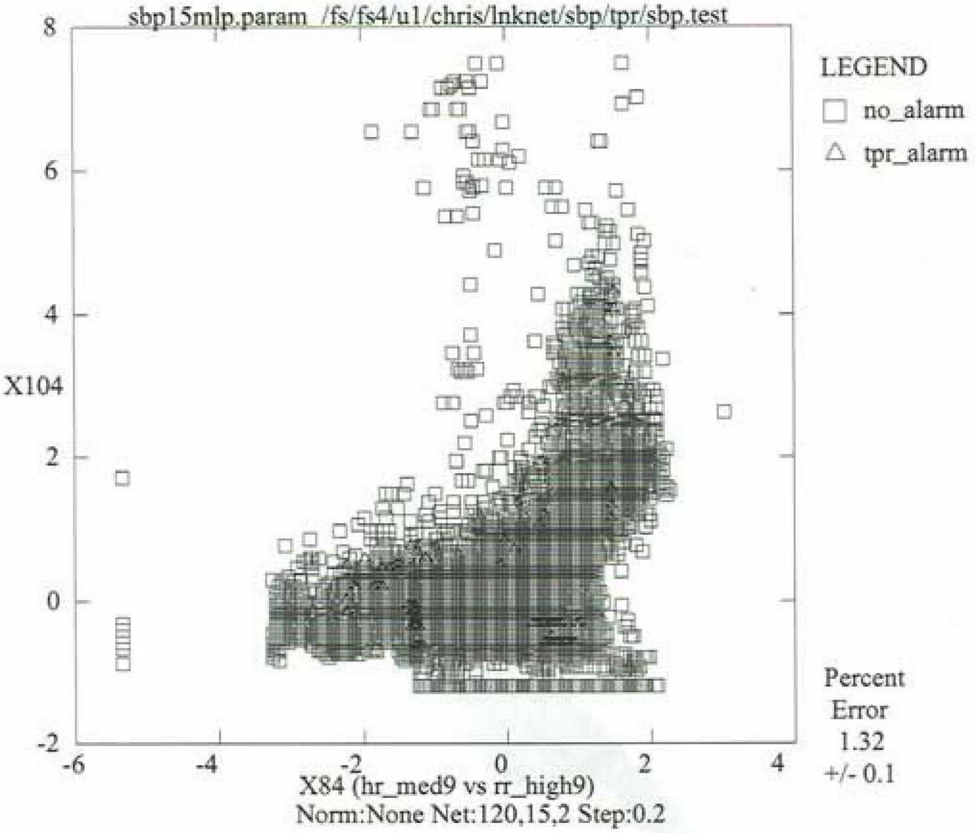


Figure 5-34: Scatter plot for systolic blood pressure true alarm detection data cases for two attribute values, hr_med9 and rr_high9 (shown normalized).

As can be seen, the very few true alarm cases are inundated by the no_alarm cases.

Despite its limitations, however, the ICU alarm example has provided a useful demonstration of how data-intensive medical time-series data may be useful, even when the underlying knowledge about how they relate to particular events is not well understood. Especially at a time when information technology is making available enormous amounts of clinical data, methods for taking advantage of these data need to be explored. The event discovery paradigm may be one technique that can assist in learning from these data.

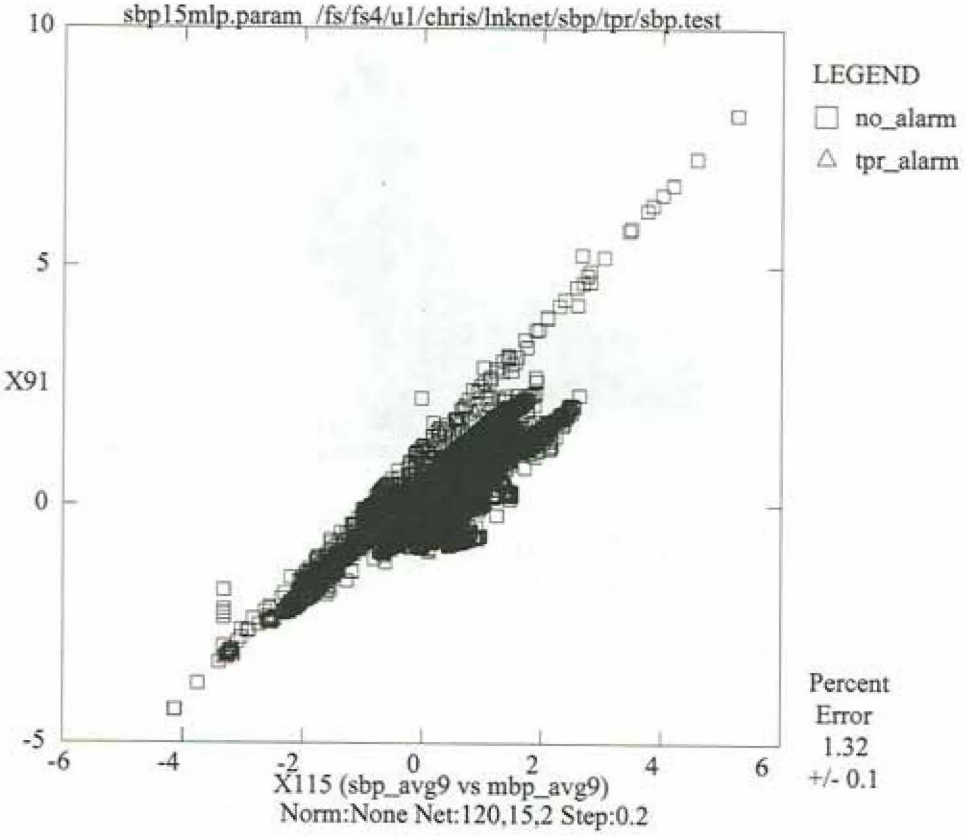


Figure 5-35: Scatter plot for systolic blood pressure true alarm detection data cases for two attribute values, sbp_avg9 and mbp_avg9 (shown normalized).

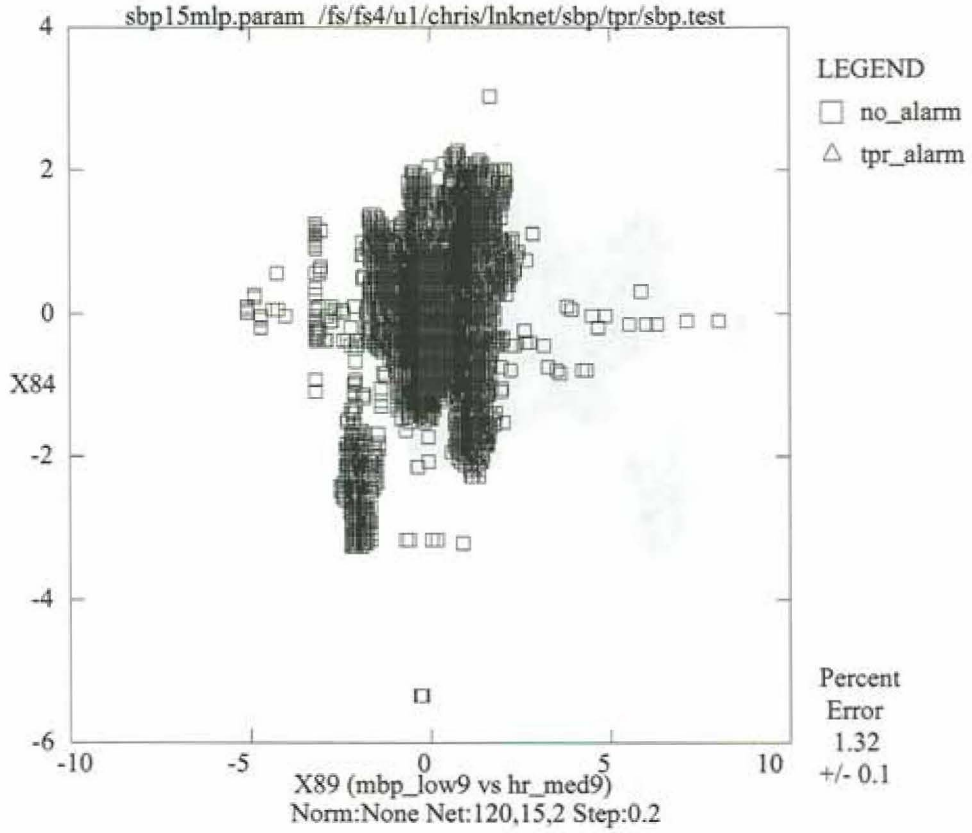


Figure 5-36: Scatter plot for systolic blood pressure true alarm detection data cases for two attribute values, mbp_low9 and hr_med9 (shown normalized).

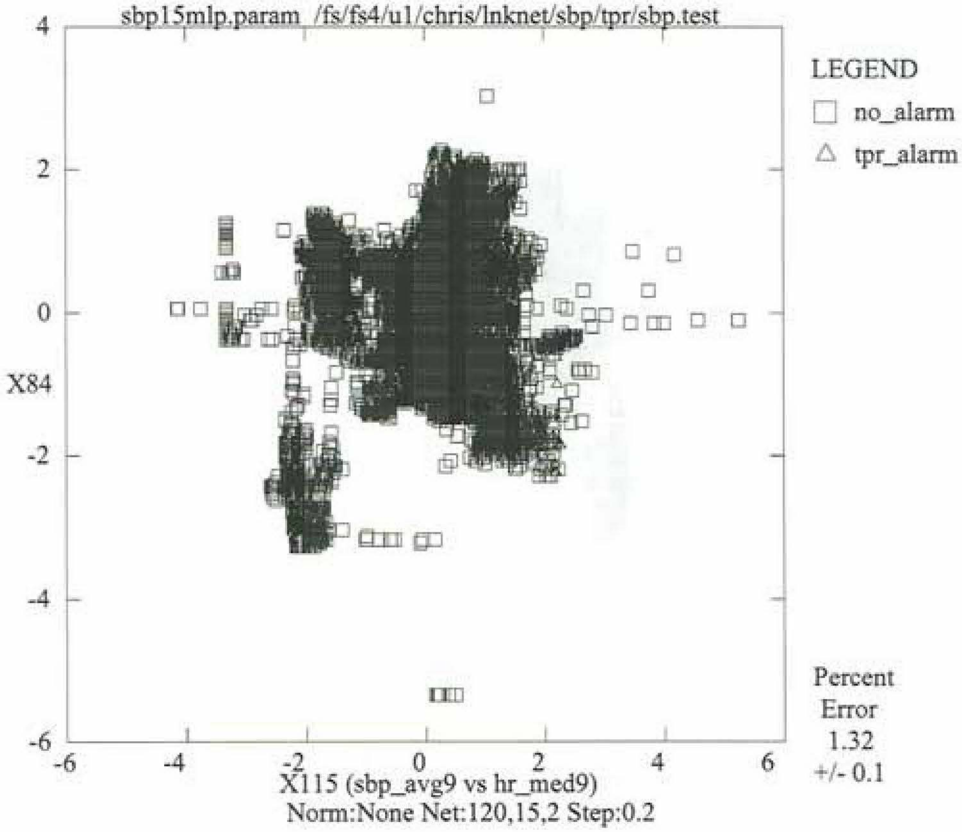


Figure 5-37: Scatter plot for systolic blood pressure true alarm detection data cases for two attribute values, sbp_avg9 and hr_med9 (shown normalized).

Chapter 6

TrendFinder Alternatives for Reducing ICU False Alarms

This chapter explores two alternative studies we performed to better understand ICU alarms and ways for their elimination. The first study is a case study on a 'solution' from industry. We compare the Nellcor N-200 pulse oximeter, which was used in the background study described in Chapter 3, to the Nellcor N-3000 pulse oximeter, which is a solution proposed by the company to decreasing false alarms due to motion artifact. The second study in this chapter examines the effects of non-machine-learned single-signal filter algorithms on false versus true alarms to help gain an understanding of whether these filters might be useful ways of describing time-series ICU data.

6.1 A Solution from Industry

Studies have shown that, of the large percentage of false alarms that sound in the pediatric intensive care unit, the pulse oximeter is the largest contributor. The goal of improved monitor alarms should be to decrease false alarms without missing true alarms. Clearly, any system can result in a decreased number of false alarms simply by never alarming; however, this inadequate method would be reflected by a parallel decrease in sensitivity. This study examines a newer model pulse oximeter in comparison with an older model to determine whether the correct goal is being achieved.

6.1.1 Background

To reduce the number of motion artifact false alarms seen in the N-100 pulse oximeter, Nellcor coupled the ECG signal to the oximeter in order to synchronize detection to heart rate; this change was implemented in their then newer N-200 model [185]. In a study of ICU monitor alarms, however, the N-200 was still found to be the source of the largest number of alarms, with as many as 91% being false [206]. In clinical practice, the synchronization feature was infrequently used, and thus was not included in their even newer N-3000 model [12], which Nellcor has promoted as a device that better handles motion artifact. The N-3000 system relies on differential amplification of the input signal, application of a series of signal-quality tests, and motion-detection testing of identified pulses [1]. Signal quality is assessed, for example, by comparing the identified pulse to a broadly defined 'normal' pulse and to the moving average of the preceding few pulses. The N-200 and N-3000 oximeters are representative of second- and third-generation pulse oximeter technology, respectively [12]. Performing a prospective clinical trial of these two models to determine their respective false positive and false negative rates might be one way to gauge the success of industry in solving this problem.

6.1.2 Methods

In a large urban children's hospital ICU, on patients with one sensor already in use with a Nellcor N-200 pulse oximeter, an additional, new sensor was placed on the patient in a corresponding location as follows: if palm, other palm; if sole, other sole; if big toe, other big toe; if thumbnail, other thumbnail; if ear, other ear; if finger, another finger of the same hand. On patients with no pulse oximeter sensors, the nurse placed two new sensors in corresponding locations. The nurse then flipped a coin to randomize: for heads, the old sensor remained with the N-200 while the new sensor was attached to the Nellcor N-3000 pulse oximeter; for tails, the old sensor was attached to the N-3000 while the new sensor was attached to the N-200. A trained observer then sat by the bedside with a custom-designed Microsoft Access database in which to record all oxygen saturation (S_pO_2) and heart rate (HR) alarm soundings for the two pulse oximeters, along with annotations of alarm appropriateness.

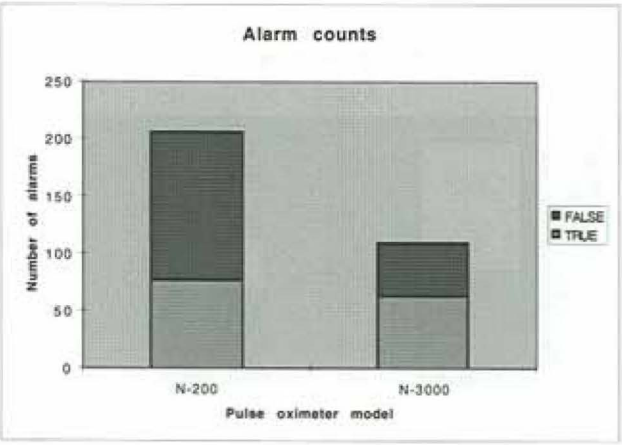


Figure 6-1: Total alarm counts for the N-200 and N-3000.

6.1.3 Results

During one month, 85 hours of data were collected. Total alarm counts are shown in Figure 6-1. For S_pO_2, the N-200 sounded 68 true alarms and 79 false alarms, while the N-3000 sounded 57 true and 33 false alarms. These alarm soundings comprised 30 alarm events. Of these, the N-200 missed none, while the N-3000 missed six. For HR, the N-200 sounded nine true and 51 false alarms, while the N-3000 sounded six true and 13 false alarms. No HR alarm events were missed by either pulse oximeter. Sensitivity and positive predictive value of the N-200 S_pO_2 signal were 100% and 46%, respectively, while for the N-3000 were 80% and 63%, respectively. Figures 6-2 and 6-3 show the decomposition of alarms by type for each of the monitors and each of the signals.

6.1.4 Conclusions

The N-3000's S_pO_2 had a higher positive predictive value than the N-200 but only at the cost of a lower sensitivity. While efforts at decreasing false alarms via these techniques did result in fewer false alarms, a parallel decrease in true alarms points to a need for better solutions.

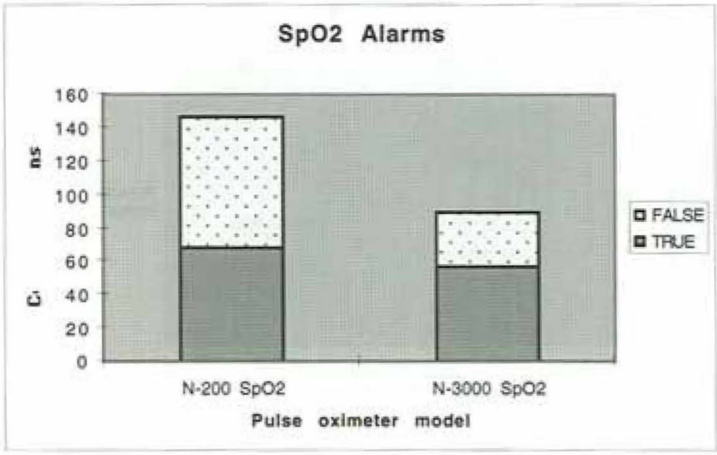


Figure 6-2: Breakdown of oxygen alarms by type for N-200 and N-3000.

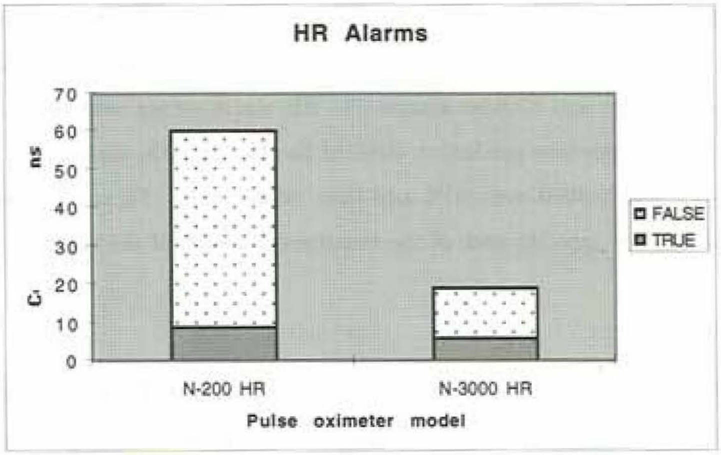


Figure 6-3: Breakdown of heart rate alarms by type for N-200 and N-3000.

6.2 Single-Signal Filters

To be able to develop better alarm algorithms requires an understanding of the effects that different filter algorithms have on different types of alarms. An ideal filter–one that substantially reduces false alarms without missing any true alarms–is neither obvious nor easy to design. A systematic comparison of the different effects of various filters on false versus true alarms can increase understanding and aid in algorithm development. Such a comparison has been performed on an actual ICU data set to explore the four filters: moving average, moving median, delay, and sampling rate.

6.2.1 Methods

A systematic comparison of algorithms requires having: annotated ICU data (i.e., ICU data in conjunction with notes of alarm soundings that occurred during the data collection time), specific filter algorithms to test, and a data processing system for first applying the filters to the data and then reporting the results. Each of these three components will be described more thoroughly.

Annotated ICU Data

The annotated ICU data used for this study were collected in the multidisciplinary ICU of Children's Hospital in Boston over the course of 12 weeks. The physiologic values derived from patient monitoring devices in a given bedspace were connected to the SpaceLabs bedside monitor (SpaceLabs Medical, Redmond, WA). A laptop computer placed at the bedside recorded values transmitted via a serial line from the SpaceLabs monitor approximately every five seconds. Annotations to the physiologic data were made by a trained human observer into an Access (Microsoft, Redmond, WA) database program running on the laptop. For each alarm that sounded in the bedspace under surveillance, the trained observer created a time-stamped note consisting of alarm origin (e.g., systolic blood pressure), alarm appropriateness (i.e., false alarm, clinically-relevant true alarm, or clinically-irrelevant true alarm), and alarm cause (e.g., patient movement). The trained observer, in addition, would have each annotation verbally verified by the bedside nurse. Furthermore, the trained observer would record the upper and lower threshold limits for each physiologic parameter being tracked.

Resulting annotated data consist of raw data files (one file for every patient monitoring period during which one set of threshold limits were in use); a database of all alarm annotations for those monitoring periods; and a database of all alarm threshold settings for each monitoring period (i.e., for each raw data file).

Four Filter Algorithms

A basic algorithm used by patient monitoring devices is the standard threshold filter (STF), which sounds an alarm as soon as a monitored value goes above the set upper limit or below the set lower limit. Granted, some filters in use may be more sophisticated than this. Nevertheless, as mentioned earlier, existing monitors have exorbitant false alarm rates; thus, regardless of how much more sophisticated these filters are, improvement is still needed. For purposes of this study, each filter analyzed is compared in two ways. First, each filter is uniformly compared against the performance of the standard threshold filter on the collected data; this produces a basis for relative comparison amongst different filter types. Second, each filter is compared against the number of alarm annotations that were recorded at the bedside; the purpose is to elucidate potential benefits or shortcomings of this type of analysis. Each of the four filters tested–moving average, moving median, delay, and sampling rate–are described in more detail below.

Moving Average Algorithm The moving average algorithm takes as input the number of values (N) for which an average should be computed, along with the upper and lower threshold limits that were in use for the given data set. The first (N-1) values read are not compared against the threshold limits. Once N values have been read, an average (i.e., the arithmetic mean value, computed by adding all N values and then dividing by N) is calculated and compared against the threshold limits. If the average value is outside the limits, the most recent time-stamp value is recorded along with the out-of-bounds average value and the physiologic signal type (e.g., 11:10:52, 193, systolic blood pressure). If the average value is within the limits, nothing is recorded and the processing program continues. Upon reading another value, the N+1st value, an average is taken of the N values starting from the second value read to the N+1st value. This newly computed average is then compared against threshold limits, and so on. The parameter, N, can range from 1 (which is equivalent to applying the standard threshold filter) to any higher integer value. An N value of 12, for example, would be equivalent to taking an average over every approximately one minute.

Moving Median Algorithm The moving median algorithm works in the same manner as the moving average algorithm. The only difference is that once N values have been read, the median value (rather than the mean) of the N values is used for checking against threshold limits. In cases in which N is an odd integer, the median value is the (N+1)/2-positioned value in a list of the N values that is sorted by increasing value. When N is an even integer, the median value is the mean of the N/2-positioned and the (N/2+1)-positioned values in the ordered list.

Delay Algorithm The delay algorithm takes as parameter an integer, N, and only 'sounds' an alarm when N sequential values have been above or N sequential values have been below the set thresholds. The sounding of an alarm is thus 'delayed' until the out-of-bounds value has persisted for a particular length of time. The time recorded for the alarm is the time-stamp of the first value that went out of range.

Sampling Rate Algorithm The sampling rate algorithm is almost identical to the standard threshold filter, except that sampling rate of values is controlled by the input parameter, N. If N is equal to one, each and every recorded data value (approximately one per five seconds) is tested against the upper and lower threshold limits. For N greater than one, only one value in each N values will be tested against the limits. Essentially, the sampling rate of the data values is decreased as N is increased. For example, for N equal to two, sampling rate is approximately once per 10 seconds; for N equal to three, sampling rate is approximately once per 15 seconds.

For each of the four algorithms presented, 28 N values are tested: one through 24, inclusive, as well as the values 27, 30, 33, and 36. These values test each filter on a range of approximately five seconds' worth to approximately three minutes' worth of data values.

Data Processing System

The analysis system consists of a series of data processing programs which take as input: physiologic values collected from the bedside patient monitor; corresponding time-stamped annotations of the alarms that sounded; upper and lower threshold listings for each physiologic signal being tracked; the filter-specific parameter, N; and a value called the 'windowsize,' which will be described below. The final output consists of a listing of each algorithm type (e.g., moving average); its algorithm-specific information (e.g., N, the number of data values over which the moving average was applied); the effects of this filter on alarms as compared to the total number of alarms that had originally been recorded in the annotations; and the effects of this filter on alarms as compared to the number of alarms found by the standard threshold filter. Figure 6-4 illustrates the data processing system.

For example, say that the trained observer recorded 10 clinically-relevant true alarms and 50 false

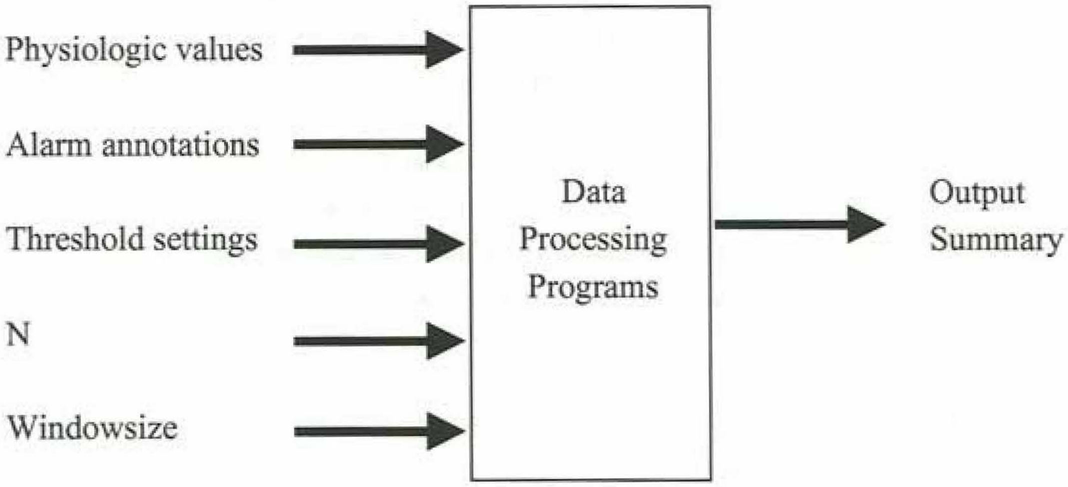


Figure 6-4: Data processing system for experimenting with single filter algorithms.

Table 6.1: Sample filter comparison results.

	Vs. Annotations:	Vs. Annotations:	Vs. Standard Threshold Filter:	Vs. Standard Threshold Filter:
Filter type	TP-R No.found/ No.expected	FP No.found/ No.expected	TP-R No.found/ Max.No.found	FP No.found/ Max.No.found
STF	80%	60%	100%	100%
avg 4	60%	20%	75%	33%

alarms. During the same period of time, data values were collected once every five seconds. The standard threshold filter, which is then run over this collected data, finds eight clinically-relevant true alarms (8/10=80%) and 30 false alarms (30/50=60%). (This discrepancy in alarm counts arises because the bedside monitors use algorithms that are slightly different than the standard threshold filter, and because the collected data points are more sparse than the original bedside monitor data). A moving average over four data points may find six of the clinically-relevant true alarms found by the standard threshold filter (6/10= 60%), while finding 10 of the false alarms found by the standard threshold filter (10/50=20%). These results would be reported as in Table 6.1.

A monitoring period is a period when all alarm threshold settings remain unchanged; it is defined

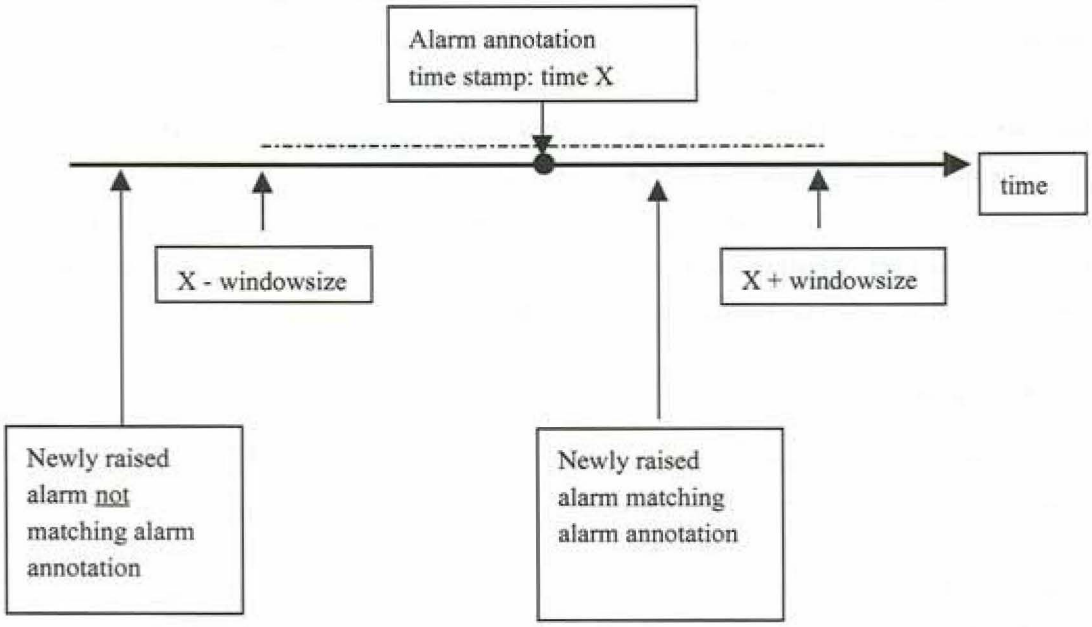

Alarm annotation
time stamp: time X
time
X - windowsize
X + windowsize
Newly raised alarm not matching alarm annotation
Newly raised alarm matching alarm annotation

Figure 6-5: The notion of 'windowsize' calculated for each alarm annotation to aid automated alarm filter processing.

by a start time (and date) and an end time (and date). Data processing occurs by first reading the alarm limits, and the start and end times, for a given monitoring period. Then, the database of alarm annotations is read, and each annotation that occurred between the start and end times of the monitoring period under analysis is stored by the processing program. Next, all physiologic data values occurring within the start and end times are read by the processing program, which applies the appropriate filter on the appropriate number (N) of data values before testing the resulting value against the current threshold limits. Any time a resulting value surpasses a threshold (i.e., 'an alarm is raised'), the program searches the stored list of alarm annotations for the annotation corresponding to the newly raised alarm. This automated search is made possible by first calculating a time 'window' around each alarm annotation; this is illustrated in Figure 6-5. The program then determines whether the time-stamp of the newly raised alarm is within the time window of any stored alarm annotation. The 'windowsize,' which can be specified at program run-time, indicates the number of seconds that should be added to both sides of an alarm annotation's time-stamp for this search. The analyses performed use windowsize values of 60, 90, and 120 seconds.

After all filter type and parameter value combinations have been tested, composite listings of results are created. One set of listings is created for each different windowsize used. Each set consists of 11 listings, one for a composite of all alarms, and one each for results tallied by: electrocardiogram heart rate high threshold alarms; ECG heart rate low threshold alarms; systolic blood pressure high alarms; systolic blood pressure low alarms; mean blood pressure high alarms; mean blood pressure low alarms; respiratory rate high alarms; respiratory rate low alarms; arterial oxygen saturation high alarms; and arterial oxygen saturation low alarms. Within each listing, alarms are grouped by type (e.g., clinically-relevant true alarms, or false alarms).

6.2.2 Results

Over the twelve-week data collection period, 223 monitoring periods were followed, totaling 35,118 minutes. For each of these periods, corresponding alarm threshold settings, alarm annotations, and physiologic data values were recorded. Overall, 2435 alarms were recorded in the annotations. Of these, 221 (9%) were clinically-relevant true alarms; 437 (18%) were clinically-irrelevant true alarms; and 1777 (73%) were false alarms.

The analysis processing programs were then run for each monitoring period. The filters tested included 28 variations each of moving average, moving median, delay, and sampling rate, for a total of 112 filter types. (Note, though, that moving average of one value, moving median of one value, delay of one value, and sampling rate every one value, are equivalent to each other as well as to the standard threshold filter.)

At least three types of conclusions can be drawn from the processing results: the filter types that are closest to the 'ideal' for each physiologic signal type can be postulated; trends that exist within one kind of algorithm (e.g., moving average) as the value of N is varied can be observed; and the effect of altering the windowsize value can be determined.

Filter Types Closest to 'Ideal'

Recall that an 'ideal' filter is one that substantially decreases false alarms without missing true alarms. A modified ranking is used in comparing filter types for this study: first, only filters retaining at least 90% of the clinically-relevant true alarms that can be found by the standard threshold filter are considered. Of these, the filter that retains the smallest percentage of false alarms that are recorded in the annotations is 'closer to ideal.' In cases where these latter percentages are identical for several filters, the smaller N values are favored. Table 6.2 lists the most favorable filters for each type of algorithm according to each of the signal types analyzed with a windowsize of 120 seconds. (No data is shown for 'arterial oxygen saturation high' because there were no alarms of this type in the annotations and no alarms of this type found by the filters.)

Trends within a Particular Algorithm Type

Another result of this study is the ability to see how the effects of the four algorithms, applied to the ICU data, vary as the parameter, N, changes. Table 6.3 lists the differences between two percentages: the percent of the clinically-relevant true alarms (which were found by the standard threshold filter) that a particular filter can find; and the percent of false alarms (which were found by the standard threshold filter) that a particular filter can find. For the moving average algorithm, this difference value increases monotonically until the value of N is equal to nine, and then decreases. For moving median, this difference is relatively uniform, and negative. The delay algorithm shows a trend similar to that of moving average, but with a peak at N=6. Finally, no trend is immediately discernible from the difference values for the sampling rate algorithm.

Effect of Different Windowsize Values

Varying the windowsize value (60, 90, and 120 seconds) during data processing appears to have no significant effect. For example, the composite pages, from processing with each of the three windowsize values shown, have nearly identical most 'ideal' filters of moving average with N=3, moving median with N=3, delay with N=1, and sampling rate with N=3. The only exception is that moving median with N=4 has slightly better results than with N=3 for the windowsize value of 120 seconds.

6.2.3 Discussion

There is no question that high false alarm rates are less than desirable in the ICU. Chapter 3 discussed these issues in detail. This study has offered a glimpse at one method of beginning a systematic exploration of the vast realm of possibilities. By having a methodical analysis process, comparisons of different filters can be made with greater confidence in the validity of results.

What has also become evident again, as in Chapter 5, is the necessity for accurate annotation recording. Without annotations that are accurately time-stamped, automatic searches for alarms within an annotation database become impossible, thus making alarm algorithm comparison difficult.

Table 6.2: Most 'ideal' filters for each signal type.

Signal	Filter type	Vs. Annotations: TP-R No.found/ No.expected	Vs. Annotations: FP No.found/ No.expected	Vs. Standard Threshold Filter: TP-R No.found/ Max.No.found	Vs. Standard Threshold Filter: FP No.found/ Max.No.found
STF		89%	41%	100%	100%
COMPOSITE	avg 3	81%	33%	92%	81%
	med 4	81%	41%	91%	99%
	del 1	89%	41%	100%	100%
	sam 3	82%	34%	92%	83%
ECG HR HIGH	avg 6	78%	9%	90%	30%
	med 5	78%	7%	90%	24%
	del 2	78%	11%	90%	34%
	sam 9	81%	12%	94%	39%
ECG HR LOW	avg 1	89%	41%	100%	100%
	med 1	89%	41%	100%	100%
	del 1	89%	41%	100%	100%
	sam 1	89%	41%	100%	100%
SYS BP HIGH	avg 2	87%	59%	97%	93%
	med 2	87%	59%	97%	93%
	del 1	90%	63%	100%	100%
	sam 2	88%	58%	98%	92%
SYS BP LOW	avg 20	100%	53%	100%	69%
	med 1	100%	77%	100%	100%
	del 1	100%	77%	100%	100%
	sam 16	100%	53%	100%	69%
MEAN BP HIGH	avg 15	100%	77%	100%	80%
	med 4	100%	87%	100%	90%
	del 3	100%	87%	100%	90%
	sam 11	100%	77%	100%	80%
MEAN BP LOW	avg 21	100%	0%	100%	0%
	med 13	100%	0%	100%	0%
	del 6	100%	0%	100%	0%
	sam 7	100%	0%	100%	0%
RR HIGH	avg 3	95%	55%	95%	90%
	med 4	95%	42%	95%	68%
	del 2	95%	55%	95%	90%
	sam 9	95%	37%	95%	60%
RR LOW	avg 1	89%	41%	100%	100%
	med 1	89%	41%	100%	100%
	del 1	89%	41%	100%	100%
	sam 18	100%	18%	100%	35%
O_2 SAT LOW	avg 6	78%	32%	93%	73%
	med 1	89%	41%	100%	100%
	del 2	78%	34%	93%	80%
	sam 5	81%	29%	97%	69%

Table 6.3: Difference between percent of clinically-relevant true alarms versus percent of false alarms found by filter.

N value:	Average	Median	Delay	Sampling
1	0	0	0	0
2	9	-15	13	5
3	11	-11	17	9
4	12	-8	18	5
5	15	-13	20	13
6	15	-17	24	12
7	17	-12	19	14
8	26	-12	17	11
9	27	-13	12	18
10	27	-12	11	15
11	23	-12	10	9
12	23	-14	9	7
13	20	-12	8	5
14	20	-14	7	5
15	20	-11	8	15
16	22	-14	7	13
17	19	-14	6	19
18	20	-14	6	15
19	18	-15	1	11
20	19	-15	-1	12
21	19	-14	0	5
22	18	-14	2	11
23	18	-12	3	16
24	19	-13	3	4
27	17	-14	2	12
30	11	-13	2	2
33	14	-19	-2	5
36	14	-20	-2	15

Ability to adjust the 'windowsize' was explored as a potential compensatory mechanism for imperfect annotations, but seemed to have little overall effect.

Another potential source of inaccuracy is the limited data collection rate of one per five seconds; without a higher sampling rate, certain alarms become 'lost' in the recording process and thus are not particularly amenable to analysis.

In considering the results, another realization is that more annotated data are needed if analyses are to be trusted. For example, only the composite listing, ECG high heart rate alarm, arterial oxygen saturation low alarm, and systolic blood pressure high and low alarms, each had more than 30 clinically-relevant true positive alarm annotations recorded. When there are very few alarms for a particular physiologic signal type, significance of results is difficult to ascertain. Thus, in Table 6.2 for example, the filters that appear to retain 100% of true alarms while decreasing the percentage of false alarms are deceptive because there were too few alarms in each of those categories. The situation is somewhat better for false alarms: all signals except the mean blood pressure low alarm and the arterial oxygen saturation high alarm had more than 30 alarm annotations.

In all analyses performed in this study, the clinically-irrelevant true alarms have not been taken into account. This omission has been deliberate because it is unclear whether it would be desirable to change the frequency of these alarms one way or the other. Clinically-irrelevant alarms have previously been found to occur mostly during patient interventions [206], meaning that a caregiver is attending to the patient when the alarm sounds. Because a caregiver is present, and can in fact cause the alarm (e.g., while performing a procedure), it might be argued that the caregiver can immediately and correctly react to the alarm sounding, with the implication that such alarms are not particularly problematic.

Finally, while none of the tested filters would be viable as a complete solution to developing improved alarm algorithms, the results of this study can direct attention to the single-signal filters that may prove more effective when used in conjunction with other filter methods or as preprocessing steps for multi-signal algorithms.

Chapter 7

Related Work

In addition to the TrendFinder alternatives presented in the previous chapter, there are also several other possibilities that have been proposed or explored for various aspects of the work we've described in this thesis. Related work can roughly be grouped into five areas: analysis of time-series data, application of machine learning-based techniques to other clinical problems, understanding of monitoring devices and alarms, use of single-signal algorithms to improve monitoring, and use of multi-signal algorithms to improve monitoring. There are necessarily areas of overlap among these groups.

7.1 Time-Series Analysis

Time-series analysis and prediction is a large field in and of itself. We have purposely avoided the idea of predicting next values in a time series, though some have chosen to focus upon prediction [19, 168, 218] and the topic is no doubt important. Several other studies have also explored the idea of learning rules [44] or patterns from time-series data. Das *et al.* describe a method that is similar to ours in that they try to learn from the data itself and use a sliding window like we do to describe successive pieces of the time-series data. Their work is different, however, in that they have no sense of when particular events occur, as we do from our annotations. They choose to use clustering of time intervals; with an unknown goal, their system produces numerous candidate rules for a user to peruse. They do not know, however, which rules are meaningful. Keogh and Pazzani are also interested in classifying information from time-series data; their approach, however, is to use piecewise linear representations of the data with weight vectors for assigned importance of each piece [102]. A relevance feedback system then requires the user to rate choices of how similar different pieces are. Guralnik and Srivastava have a similar goal of detecting events in time series when rules are ill-understood [75]. Their method, however, assumes that time series can be modeled mathematically and parametrically; it is based upon finding 'change-points' in these models, and is only for single-signal analysis.

Fawcett and Provost are interested in monitoring over time for events requiring action, such as credit card fraud [58]. Their 'event' of interest, however, is an ill-defined notion of 'activity' such that 'true negative' periods are not well defined. Additionally, their method would not lend itself to detecting artifacts, for example. Zhang proposes an approach for discovering temporal structures [229]. The method, however, requires that the user know how to define all events of interest and their interrelationships. Haimowitz proposes the use of 'trend templates' in which one uses a trend template language to define trends [78]; one can then search for occurrences of these trends in one's data. The system, however, requires a priori understanding of the nature of the trends of interest. Hunter and McIntosh use a marking software workbench to add knowledge of events to data [86]; this is like the system that was used for creating annotations for our neonatal ICU experiments of Chapter 4. They then write rules from these marked data regions, which are subsequently used by a pattern matcher on raw data.

Guralnik *et al.* are also interested in patterns in sequence data [76]; the pattern, however, must be understood a priori and specified by the user in their constraint language. Their focus is to see if the user's specified pattern has frequent episodes in the data. Keogh and Smyth also depend on user-specified patterns for searching archived time-series data by subsequence matching [101]. Han *et al.* try to find cyclic patterns in data by looking at segment-wise periodic patterns [80].

Oates tries to learn what makes a time-series sequence different from other sequences from the same time-series using clustering [149]. Kadous also uses clustering of events as a basis for classifying, for example, sign language hand movements[97]. Padmanabhan and Tuzhilin use temporal logic to express patterns in temporal databases, though their methods are for categorical (symbolic) data [154].

In the area of medicine, several computer programs have been developed to interpret electrocardiograms, for example, using statistical, heuristic, or deterministic approaches [224]. Balm experiments with sampling rate, time and amplitude normalization, and cross-correlation to decide whether an ECG segment is normal or not normal [11]. Differences in performance of various approaches to

ECG interpretation could be informative for understanding monitor alarms as well.

Kaplan explains that one of the goals of analyzing variability in serial physiologic measurements is to convert from a large series of values to a single number or small set of numbers that effectively represent the original series [98]. He contrasts "measures of central tendency," such as mean, mode, and median, from "measures of variability," such as standard deviation, variance, range from maximum to minimum, and range from, for example, 10th to 90th percentile. He emphasizes that none of these measures capture the ordering of values. On the other hand, he suggests that analyzing the autocorrelation function and/or power spectrum of a time series could provide new insights into physiological functioning. Our studies have used feature attribute encompassing both "measures of central tendency" and "measures of variability;" in addition, our slope attributes give an idea about the ordering of values.

Mormino *et al.* describe their computer analysis of continuously recorded blood pressure signals [144]. Their data processing includes determination of the maximum and minimum values over a specified time interval and determination of the mean and standard deviation of values for 3-minute, 30-minute, and 24-hour periods. These feature attributes are similar to a subset of ours.

Hitchings *et al.* present a method for detecting trends in monitor data using "exponentially mapped past statistical variables" [85]. They explain that because most physiological variables are pseudo-random in nature, special techniques based, for example, on their power spectral density and probability distribution function, need to be used in determining their trends. This method, however, requires initial knowledge about the normal biological variation of the signal being monitored.

Cao and McIntosh identify false alarms in neonatal ICU physiologic signals by deriving a 'drop' stream and a 'rise' stream from which to predict future drops and rises in signal values by regression [29]. The results, however, are difficult to interpret given the limited testing performed thus far.

Ridges *et al.* use computers to analyze atrial pressures [171]. First, the waveforms are digitally filtered to remove the higher frequency artifacts while preserving the low frequency information. Second, the pressure waveforms within one standard deviation of the mean cycle length for the recorded data are averaged in order to remove random noise without removing the repeating cardiovascular waveform. They found these methods to work reasonably well for waveform interpretation. Gibby and Ghani describe a computer algorithm for monitoring precordial Doppler audio in order to identify possible presence of venous air embolus during surgery [69]. The algorithm is based on fast Fourier transform spectral analysis of the Doppler signals.

While we have not specifically experimented with frequency domain methods in our work, many others have described both Fourier [23, 26] and wavelet techniques for signal analysis [134, 135]. As Meyer explains, there are stationary signals, whose properties are statistically invariant over time, and nonstationary signals, in which transient events (that are not predictable) occur [135]. He further explains that the Fourier transform is ideal for studying stationary signals, and that wavelets are useful for studying nonstationary signals. Our data signals appear to be nonstationary ones. That we examine each signal by small 'pieces' is akin to the idea of 'time-frequency' wavelets, while our use of multiple time intervals represents a type of multi-scale analysis akin to the idea of 'time-scale' wavelets. A useful addition to our work, for example, might be to use Fourier analysis techniques on individual time interval segments to derive additional feature attributes.

Price discusses the use of computers in neurosurgical intensive care; he suggests that dense time-series data can be compressed by averaging and storing only the mean and its standard deviation, by using sequential histogram analyses to follow the median and distributional skew, and by spectral (Fourier) analysis to demonstrate changes in component frequencies [163]. We have found through actual experimentation that compression by averaging is in fact a useful means of looking at dense neonatal ICU time series data in such a way that artifacts are retained. In contrast, Young *et al.* have suggested using a median filter over 3 minutes for data reduction that leads to rejection of artifacts [228].

Dietterich and Michalski propose data transformations such as 'segmenting,' which divides the original sequence into non-overlapping segments; 'splitting,' which breaks a single sequence into several subsequences and could thus enable a system to discover periodic rules; and 'blocking,' which breaks the original sequence into overlapping segments for which new attributes can be derived [48]. Our method of deriving feature attributes from successively overlapping subsequences is similar to the described 'blocking' method, while our method of exploring multi-phase learning is somewhat reminiscent of the described 'segmenting' method.

Noise on monitored signals is usually an undesirable hurdle. Gamberger *et al.* propose a method for noise filtering by disregarding values that are non-typical, including both errors and outliers [66]. This type of system, however, would not, for example, be able to detect clinically-relevant true alarms that happen to manifest in the sensors as atypical values.

Others have also proposed different techniques for learning from time-series data [14, 28, 138].

7.2 Machine Learning Applied to Similar Domains

The problem of understanding patient monitor time series data can in some ways be thought of as a pattern recognition problem, and hence past work in the pattern recognition field [51] may provide insights. For example, useful ideas may include data smoothing (e.g., using a seven-point least square error parabolic fit smoothing technique), convolution, factor analysis (a technique for generating new properties which are linear combinations of the old ones), and Fourier series analysis [73]. Pattern recognition techniques in speech recognition or identification may also be informative [119, 232]. Another pertinent area is that of automatic feature selection [31].

Machine learning techniques [137, 139] have recently been more widely used to tackle pattern recognition problems. Closely related to machine learning is the fairly new field of data mining [5, 8, 117, 219, 220, 233], in which large amounts of data are used to extract new knowledge. Methods used in data mining, such as data preparation and data reduction, might be useful in the ICU monitoring domain as well. Machine learning methods have already been used in various medical domains [50, 63, 99, 143, 209], such as diagnosing myocardial infarction in emergency room patients with chest pain [208], predicting pneumonia mortality [40], and interpreting digital images in radiology [157], just to name a few. Most applications, however, have not involved high density time series data.

One similar example of the use of machine learning for time series data is the work of Orr *et al.*. They developed a prototype intelligent alarm system that uses a neural network to process CO_2, flow, and pressure values in the operating room in order to identify critical events [56, 57, 88, 151, 152, 172, 221]. Their system was made from simulating anesthesia faults using dogs. They report, for example, that in the operating room, their system was able to find 54 out of 57 faults with 74 false alarms [152].

Neural networks have also been tried with promising results for recognizing patterns in cardiotocograms used for fetal monitoring [212]. A number of machine learning techniques and mechanisms for using them in medicine have recently been described by Lavrac [117].

For a very different application area, Ginzburg and Horn proposed a combined system that uses two sequential neural networks to perform time series prediction of sunspots [70]. The first network is trained to predict the next element based on previous elements in the time series, while the second network is trained to predict the errors of the first network in order to correct the original predictions.

7.3 Understanding Patient Monitors and Alarms

Several researchers have studied ICU monitoring devices and their alarms. Similar monitors are used in other hospital areas, such as the operating room and the post-anesthesia care unit (PACU), and have also been studied. Relevant findings from all hospital areas will be discussed. Particular emphasis is given to previous work related to pulse oximetry because of the widely acknowledged high false alarm rate of this device.

Computer analysis of the electrocardiogram was described at least as early as 1963 [159] and computer systems for acquiring, processing, and displaying data from critically-ill and/or post-surgical patients have been described at least as early as 1966 [82, 89, 90, 93, 145, 190, 214, 215]. At that time, it was already felt that a computerized monitoring system could help to interpret the large quantities of data available from bedside monitors, as well as facilitate research in understanding complex physiological mechanisms associated with different disease states [93]. In 1974, Sheppard and Kirklin described a computer system that had been in use for seven years which assisted in both patient monitoring and closed-loop control of blood infusion [189], and later, in closed-loop control of drug infusion as well [188]. They believe that use of the computer system, by performing repetitive and time-consuming routine tasks, has led to a 25% increase in nursing time available for direct patient care [188]. More recently, an automated vital signs monitor has made it possible for certain patients to transfer from the ICU to a medical/surgical unit [126]. It was realized early on, however, that efforts were needed to decrease the rate of equipment failure and false alarms [169].

Others have also discussed intelligent patient monitoring [24, 37, 46, 87, 105, 110, 122, 195, 211] and control [32, 38, 92, 125, 141, 184], the role of emerging technologies [55, 79, 115, 127, 146, 163, 198, 216], and the future of critical care [6, 35, 133, 175, 192]. Smith proposes that development and implementation of new clinical monitoring devices might be accelerated with greater collaboration between industry and academia [194].

The pulse oximeter, developed in the early 1970s but not widely used until the mid-1980s [148], is now thought of as an invaluable device for continuous non-invasive measurement of arterial oxygen saturation [18, 176, 179, 181] and thus detection of hypoxemia. However, it is also well known to have a high false alarm rate [155, 206, 223, 225]. Various efforts have been made to improve this device or decrease the occurrence of false alarms but the problem does not appear to be fully solved. The high false alarm rate of pulse oximetry is believed to be due in most cases to a low signal-to-noise ratio [185]. Low signal includes low perfusion (e.g., from relative hypothermia or shock conditions) and improper probe placement, while high noise may be due to motion, ambient light, abnormal hemoglobinemias, and venous pressure waves. Some say that pulse oximetry is insufficient for certain patients who are likely to have impaired CO_2 removal, and thus suggest the use of continuous intra-arterial blood gas monitoring [156]. Clearly this would not be practical for the majority of patients who are currently being monitored by pulse oximetry.

High false alarm rates have not been limited to one model of pulse oximeter. Datex Instrumentarium's Cardiocap CH IIS (Helsinki, Finland) [223] and Nellcor's N-100 [225] as well as N-200 [206] (Hayward, CA), for example, all have very high false alarm rates. Barrington *et al.* evaluated the performance of the N-100, N-200, Novametrix 500 (Wallingford, CT), and Ohmeda Biox 3700 (Boulder, CO) in the neonatal ICU; they found the oximeters to be very sensitive to motion artifact, with some improvement from the N-200's use of ECG synchronization [106]. Bentt *et al.* have suggested correlating pulse oximetry heart rate with ECG-derived heart rate [17]. Recording both the oxygen saturation signal and the pulse signal itself from an oximeter might also be useful in validation of the saturation signal [147].

Similar problems have been reported both for routine anesthesia monitoring, in which 75% of all auditory alarms were spurious alarms and only 3% represented any patient risk [104], and for home monitoring, in which only 8% of alarm events were classified as true events [217]. Likewise, 77% of pulse oximetry alarms in the PACU were found to be false [223], and 87–90% of pulse oximeter-detected desaturations during dental sedation were due to patient movement [225]. Home apnea monitors for neonates have also been demonstrated to generate a large number of false alarms [173].

Most existing patient monitors use limit alarms which sound when the signal value has surpassed a preset or operator-set threshold value. Beneken and van der Aa suggest that it would be useful to have a method for automatically determining these threshold levels, for example from statistical data collected from different groups of patients [15]. Moreover, Grieve *et al.* showed that different pulse oximeters (e.g., Nellcor and Ohmeda) can consistently report higher or lower saturation values and thus different threshold levels should be employed depending on which company's monitor is in use [74].

Another possibility for determining threshold levels is by using statistical process control (SPC) methods. Fackler *et al.* emphasize, however, that traditional SPC methods do not work well when data are auto-correlated or are rounded to whole numbers, such as is the case with pulse oximetry monitors [54]. Laffel *et al.* use SPC in the way of control charts [160] for analyzing serial patient-related data [114]; it allowed them to get an idea about a patient's condition, but it is not clear how effective control charts would be for alarming purposes. DeCoste suggests a different method, large-scale linear regression, to learn the high and low limit bounds for spacecraft sensors [47].

In all efforts to decrease false alarms, it is imperative to keep in mind the inverse relationship between sensitivity and specificity. Stated in an extreme manner: all false alarms could be eliminated (100% specificity) simply by never sounding any alarms (0% sensitivity); this system, however, would not be very useful. A continuous jugular bulb oxygen saturation catheter monitor, for example, was found to have an impressively low false positive alarm rate of 2%; this value can only be meaningfully interpreted, however, when the false negative alarm rate (54% in this case) is also known [41]. A more informative measure of performance, as we have described and used in our studies, would be to determine the area under the receiver operating characteristic (ROC) curve [81], which takes into account the full range of possible sensitivity-specificity pairs.

Not all monitoring devices are necessarily so problematic. Anderson *et al.* evaluated a cardiorespiratory rate monitor and found there to be no false alarms during 48 hours of monitoring [7]. Study of these devices might provide insight for how to improve monitors with high false alarm rates.

Ford *et al.* proposed and developed a respiratory alarm monitor for use with mechanically ventilated ICU patients which is purported to detect several specific types of alarms [61]. However, they have not to date performed a prospective trial and/or published results of how well the monitor works.

Klaas and Cheng found that bedside monitors telemetrically linked to a central nursing station could be useful in alerting medical personnel to hypoxemic events otherwise missed in a well-staffed ICU [107]. They believe that noise pollution and the numerous other ICU alarm soundings could contribute to the missing of bedside alarms. Jans *et al.* discuss the potential value of an electrocardiogram central station for ICU monitoring [91].

Other ICU alarm soundings include, for example, those from mechanical ventilators, which can alarm for one of a large number of reasons (e.g., high-pressure, low-pressure, low exhaled tidal volume, low exhaled minute volume, apnea, oxygen) [130]. Hayes, in a review about mechanical ventilation, suggests that future ventilators provide more informative alarms to medical personnel [83].

Along those lines, some have suggested that newer alarm systems should adopt a three-level priority system in which alarms are differentiated based on the seriousness of the event and the response required by the caregiver [25, 95, 182]. The alarm type (loud or soft, audible or visual) for each level would be commensurate with the seriousness of the event.

Scuderi *et al.* suggest that pulse oximeters may become more useful devices with the development of improved trending and analysis of S_pO_2 data, for example, using stochastic analysis techniques from the econometrics literature [183]. Morgan *et al.* propose to collect an annotated data library of ICU monitor signals to use in future development of patient monitoring aids [142]. Metnitz *et al.* also describe a system for collecting, presenting, and storing ICU data [132]; however, they do not have a method for annotating the data to indicate its validity in retrospective analysis.

Shapiro supports having quality improvement standards implemented for ICU monitors, but argues against employing standards previously passed for 'analyzers' instead of 'monitors' [187]. Such standards for patient monitoring during anesthesia have already been developed at Harvard [53].

7.4 Single-Signal Algorithms for ICU Monitoring

Single-signal algorithms refer to any methods of decreasing false alarms which only involve a single monitored signal (e.g., heart rate). This could include both alterations in alarm threshold settings or in processing of the signal itself to help decide the alarm status of that same signal without the use of knowledge about other monitored signals.

One type of single-signal algorithm would be to use rules involving signal thresholds and simple trending. As early as 1976, Frost *et al.* experimented with computerized alarms that imposed criteria such as: bradycardia alarm if heart rate is 50 beats per minute or less and there is a decrease in the average heart rate of 20 beats per minute or more; atrial tachycardia alarm if there is a sequence of 10 or more consecutive premature atrial beats at a rate of 120 per minute or more. They compared this computerized arrhythmia alarm system to the conventional analog monitoring system of the time and found that 53% of computer alarms were true positive alarms while only 8% of analog alarms were true positives [64]. Furthermore, there were four times as many false alarms from the analog system as from the computerized system. Most false positives in both systems were due to patient movement.

Two other single-signal methods that have been tried are: 1) requiring a delay time between when a signal value surpasses a threshold and when the alarm sounds (to be referred to as 'prealarm delay' [124]); and 2) varying the alarm threshold values. Pan and James [155] combined these methods, experimenting with altering alarm limit settings and incorporating a 60-second prealarm delay (which they call a 'wait-period'). They report a dramatic decrease in "maximum frequency of alarms" (from 324 to four alarms per 24 hours per patient). However, they do not report another crucial piece of information: the breakdown of false and true alarms affected. Thomas describes a PC-based monitoring and alarm system for biotechnology laboratories in which each monitored input also has its own programmed delay time to allow for brief out-of-range situations that are still considered normal [200].

Rheineck-Leyssius and Kalkman experimented with using different pulse oximeter lower alarm limits, but found that the lower alarm limit was actually associated with a larger number of hypoxemia events. They suggest that this could be due to earlier alarms (and resultant interventions) being effective in preventing longer and more severe desaturations later [170].

Benis *et al.* present a 'two-variable' (mean arterial and left atrial pressures) 'trend detection alarm.' Their system, however, actually evaluates each of the two variables separately against set limit values [16], so can be considered as a single-signal algorithm. Furthermore, their alarm algorithm essentially works by sounding an alarm only when three consecutive values (one sample per minute) are beyond limits, thus is similar to the other prealarm delay methods.

Makivirta *et al.* also experimented with varying the prealarm delay time and found that a 10-second delay could reduce the alarm rate by 26%. They did not know, however, how many true alarms were missed by this method. They additionally looked at using a median filter to remove transient changes in data signals [123, 124], which they believe can remove clinically irrelevant brief signal transients and thus increase alarm reliability.

There have also been more complicated single-signal algorithms proposed. De Beer *et al.*, for example, evaluate a method for automatically detecting artifacts in auditory evoked potential monitoring [45]. They first derive four variables per 'sweep' (neurophysiologic signal recorded during a fixed period after a stimulus): maximum amplitude, minimum amplitude, amplitude range, and the slopes between successive samples. If one or more of the variables deviates strongly from their pre-calculated normal range of values, then the sweep is considered to contain artifacts.

Adams *et al.* describe a computer algorithm for automatically recognizing pulse oximeter waveforms as normal or distorted based on a relative constancy of the systolic upstroke time which they observed in the normal waveforms [4]. They achieved a 92% sensitivity, but did not report the specificity. Their algorithm also unfortunately requires direct access to the oximeter's analog waveforms, which is not commonly available.

Bosnjak *et al.* [21] propose using a Kalman filter for detecting ischemia based on ECG analysis. The Kalman filer [33] is a method of performing trend analysis by recursively estimating the mean and slope of an input data stream. They report a sensitivity of 89.58%. Sittig also discusses a Kalman filtering algorithm for detecting physiologic trends, although his focus is on its implementation using a parallel architecture [193].

Yien *et al.* observed trends in the spectral components of systemic arterial pressure and heart rate signals that correlated with patient outcome [227]. Specifically, they found progressive increases in the power density values of the very low-frequency and low-frequency components of the systemic arterial pressure and heart rate signals in patients who went on to recover, but progressive decreases in patients who went on to have poor outcome. Power density values were calculated for 30-minute periods once per day.

Fuzzy logic has also been suggested for pattern recognition [231]. Sun *et al.* use fuzzy logic as a method of monitoring arterial oxygen saturation in mechanically ventilated newborns [197]. They first derive fuzzy sets for change in S_pO_2 and slope of S_pO_2, and then use a rule-based system to determine desired change in F_iO_2 (fractional inspired oxygen).

Shabot *et al.* [186] describe a system within a comprehensive ICU patient data management system for automatically identifying critically abnormal laboratory and blood gas values rather than vital signs monitor data like most of the other work discussed. One part of their system, however, is the detection of adverse laboratory data trends. It works by considering, for sequential values, the amount of change, rate of change, time span between values, and proximity of current value to a defined critical limit.

Statistical process control methods, as we discussed in the previous section, are also a possibility for single-signal improvements.

7.5 Multi-Signal Algorithms for ICU Monitoring

Multi-signal algorithms refer to processing of more than one monitored signal (e.g., heart rate, blood pressure, and oxygen saturation) to help decide the alarm status of one of those signals (e.g., heart rate). Researchers have been investigating approaches such as trend detection [78, 108, 193], belief networks [161, 177], rule-based systems [2, 10, 67, 68, 178], knowledge-based systems [3, 30, 65, 94, 112, 196, 201], fuzzy logic [13, 72, 226], and multiple parameter detection [43, 167]. Orr and Westenskow point out that the disadvantages of rule-based or knowledge-based systems are that they generally only use very simple and the most apparent relationships between sensors and alarms, that they require medical knowledge that is often not fully understood, and that they require finding an expert who has sufficient time to write the rules [152]. Uckun reviews several model-based efforts in biomedicine, including intelligent monitoring systems (Guardian [84], SIMON) and temporal reasoning systems (TUP, DYNASCENE, TCS) [210].

Artioli *et al.* use a linear transformation based on the Karhunen-Loeve expansion to map from 13 primitive features to two new variables that are still able to separate the two classes of their data quite well [9]; the classes are normal and high-risk patients in the post-cardiosurgical ICU, and the primitive features are physiologic measurements. The technique appears promising, though they did not apply it to time-series data. Also, that the new variables do not have a direct physiological meaning may or may not affect the technique's usefulness and acceptance.

Makivirta *et al.* experimented with a 'vector median filter' to process the systolic, mean, and diastolic values of systemic and pulmonary arterial pressures as values of a multidimensional signal [124]. They report that this type of filter might be able to provide a modest reduction in alarm rate.

Sukuvaara *et al.* present a knowledge-based alarm system that first determines the qualitative trends of median-filtered signals and then uses multi-signal, rule-based reasoning to estimate the patient's state [112, 196]. The qualitative trend is established by linear regression over a 30-minute period, and the median filtering is applied to a data window of 48 minutes. Their knowledge base consists of 87 rules derived from clinician interviews; this could make the system a difficult one to expand.

Visram *et al.* identify pulse oximetry movement artifact by analyzing the pattern of the pulse signal amplitude preceding a desaturation and comparing that pattern to a template that they derived [213]. They report a 96% sensitivity and 60% specificity in identifying artifacts. Their method for gold standard classification, however, is retrospective analysis of saturation data from two pulse oximeters, one attached to the finger and one attached to the toe; the accuracy of this gold standard is unknown.

Dumas *et al.* report on a prototype Masimo signal extraction technology (SET) pulse oximeter (Mission Viejo, CA) that is purported to better handle motion artifact and low perfusion conditions by using mathematical methods on the pulse oximeter's red and infrared light signals to measure and then subtract out the noise components from the saturation signal [52]. Barker and Shah compare this Masimo pulse oximeter to the Nellcor N-200 and N-3000 during motion in volunteers; they find that the Masimo oximeter appears to perform significantly better than the Nellcor oximeters [12].

As integrated ICU workstations become available in the future [77], it should become easier to have access to multiple physiologic signals and thus to implement multi-signal intelligent alarm algorithms.

Chapter 8

Summary of Contributions

In this thesis, we have attempted to first describe various data collection, preprocessing, and analysis techniques within a paradigm for event discovery in numerical time-series data, and then demonstrate the application of these techniques to intensive care unit monitoring. We summarize our work and findings here.

We began in Chapter 2 by presenting the TrendFinder paradigm as a general strategy for learning to detect trends of interest from time-series data. Examples of events from business, medicine, and engineering disciplines that could be learned by this method were given. Particular emphasis was placed on data annotation and preprocessing techniques; their importance has previously been understated. Model derivation by machine-learning methods was the next component in the process. Finally, performance evaluation, not only with use of separate evaluation and test sets but also with completely different data, was described. Others have reported that attempts to use traditional machine learning methods on time-series data have not had much success [102]. The techniques of the TrendFinder paradigm may be one feasible approach.

In Chapter 4, we next applied the learning paradigm to detection of signal artifact in the neonatal intensive care unit. We found that non-linear classifiers performed quite well while a linear classifier did not. This made sense once we realized the nature of the data we were working with; namely, artifacts occur in a discontinuous fashion on the spectrum of monitored signal values. Amongst non-linear classifiers, we found that Quinlan's notion of S-type data and P-type data did not completely lend itself to our domain. While the preprocessed neonatal ICU feature vectors were arguably more S-type, we found that the neural networks (which should be better with P-type data, while decision trees should be better with S-type data) in fact out-performed the decision trees. This showed that neural network-like classifiers in fact can do well with S-type data.

We also explored issues of data granularity and data compression in Chapter 4. We found that data compression of temporal data by arithmetic mean can be an effective method for decreasing knowledge discovery processing time without compromising learning.

We then presented detailed experiments in class labeling methods for preprocessing of data before applying machine learning methods. We found that front-labeling and end-labeling resulted in models with essentially equal performance such that the method chosen should be based upon the target application. For ICU monitoring, this meant that end-labeling should be used in order to be able to detect an event as quickly as possible when it occurs. Comparisons of different strictness levels in class labeling taught us that more robust models are likely to be developed when the least strict labeling methods are employed.

In Chapter 4, we also presented a technique for learning multi-phase trends of interest using piecewise phase characteristics as feature attributes given to machine learning methods. This was an effective method, yet uncomplicated and extendable.

We then applied the TrendFinder paradigm in Chapter 5 to the detection of clinically-relevant true alarms on monitored signals in the multidisciplinary medical ICU. We compared multi-signal learning to single-signal learning and found that the multi-signal machine-learned models out-performed single-signal methods (both single-signal machine-learned models and single-signal limit thresholding methods).

In that chapter, we also presented a more principled manner of choosing time intervals for feature attribute derivation. This was based upon a straightforward comparison of numbers of errors occurring with models containing only attributes derived from a single time interval. The best performers were used together in a multi-time interval model. Results showed that the models developed in this manner did at least as well in one case and much better in the other case than models developed from arbitrarily chosen time intervals.

We also introduced the idea of post-model threshold refinement for new populations in Chapter 5. Just as traditional ICU threshold limit alarms have factory-set threshold levels that can be optionally adjusted by the caregiver for a particular patient, post-model refinement also allows this flexibility for refinement of thresholds within a machine-learned detection model. We found, for example, that adapting six out of nine signal thresholds in the systolic blood pressure true alarm detection model, and otherwise keeping the model structure identical, increased model performance significantly when applied to a new population.

Finally, we proposed an alternative type of hybrid committee classifier to use both true alarm detection models and false alarm detection models in conjunction in any one of an unlimited number of ways. Our example demonstrated the feasibility of such a classifier system.

From a medicine standpoint, our contributions have been to gain a better understanding of the clinical problems of false alarms in the ICU and ways for their elimination. Our prospective study of ICU alarms, presented in Chapter 3, represents one of the most thorough alarm studies available. It has helped to further understanding of alarms, as well as their type, frequency, cause, silencing, and relationship to interventions. In Chapter 6, our case analysis of an 'improved' monitor from industry and its performance compared to its predecessor helps to demonstrate the need for still further work in this domain.

Overall, we have developed and presented several techniques to be used within an organized framework which enable successful learning of interesting trends from numerical time-series data. Moreover, we have demonstrated the feasibility and advantage of applying such techniques to ICU monitoring, an area that is extremely important, challenging, and in need of improvement.

Appendix A

List of Abbreviations

Tables A.1 and A.2 list the abbreviations and their meanings for terms used in this thesis.

Table A.1: List of abbreviations used (A–M).

abbreviation	meaning
abs_slope	absolute value of linear regression slope
Art	arterial line
Art BP	arterial line blood pressure
avg	moving average algorithm, or moving average
avg 4	moving average algorithm with N=4
BP	blood pressure
bpm	beats per minute (for heart rate)
C	positive constant used in solution to finding decision surface for SVMs
c	'confidence level' input option to c4.5, used during pruning
CO_2	carbon dioxide
d	degree of a polynomial, for example for an SVM classifier
del	delay algorithm
dia	diastolic
ECG HR	electrocardiogram heart rate
F_iO_2	fractional inspired oxygen
FN	false negative alarm
FP	false positive alarm
high	maximum value
ICP	intracranial pressure
ICU	intensive care unit
K	kernel function for the mathematical form of an SVM classifier
LEAD	electrocardiogram lead
low	minimum value
LR	logistic regression
m	minimum number of cases needed in at least two outcomes of a node for c4.5
MBP	mean blood pressure
med	moving median algorithm, or moving median
MICU	medical ICU, also multidisciplinary ICU
MLP	multi-layer perceptron
mmHg	millimeters of mercury

Table A.2: List of abbreviations used (N–Z).

abbreviation	meaning
Neg	negative
NICU	neonatal intensive care unit
NN	neural network
No.found/Max.No.found	Number found divided by maximum number found
No.found/No.expected	Number found divided by number expected
O_2	oxygen
PEEP	positive end-expiratory pressure
PO1 HR	pulse oximeter 1 heart rate
PO1 sat	pulse oximeter 1 oxygen saturation
PO2 HR	pulse oximeter 2 heart rate
PO2 sat	pulse oximeter 2 oxygen saturation
Pos	positive
PPV	positive predictive value
RBF	radial basis function
Resp. rate	respiratory rate
ROC	receiver operating characteristic
RR	respiratory rate
sam	sampling rate algorithm
SBP	systolic blood pressure
S_aO_2	arterial oxygen saturation
slope	linear regression slope
S_pO_2	arterial oxygen saturation measured by pulse oximetry
std_dev	standard deviation
STF	standard threshold filter
SVM(s)	support vector machine(s)
sys	systolic
TN	true negative alarm
TP-I	true positive, clinically irrelevant alarm
TP-R	true positive, clinically relevant alarm
TPR	true positive, clinically relevant alarm
Tree	decision tree
Vs.	versus

Bibliography

[1] Technology overview: SpO2 monitors with oxismart advanced signal processing and alarm management technology. Reference Note Pulse Oximetry Note Number 9, Nellcor Puritan Bennett, 1998.

[2] A Hosseinzadeh. *A rule-based system for vital sign monitoring in intensive care.* PhD thesis, McGill University, November 1993. Master of Engineering thesis.

[3] A Jiang. *The design and development of a knowledge-based ventilatory and respiratory monitoring system.* PhD thesis, Vanderbilt University, December 1991.

[4] JA Adams, IM Inman, SA Abreu, IA Zabaleta, and MA Sackner. A computer algorithm for differentiating valid from distorted pulse oximeter waveforms in neonates. *Pediatric Pulmonology*, 19:307–311, 1995.

[5] Pieter Adriaans. *Data Mining.* Addison-Wesley, 1996.

[6] S Alpert. Physiologic monitoring devices [editorial]. *Crit Care Med*, 23(10):1626–1627, 1995.

[7] W Anderson, AJ Brock-Utne, JG Brock-Utne, and JB Brodsky. Evaluation of a respiratory rate monitor in postsurgical patients. *J Clin Anesth*, 4(4):289–291, July 1992.

[8] C Apte. Data mining - an industrial research perspective. Research Report RC 20851 (92270), IBM Research Division, 1997.

[9] E Artioli, G Avanzolini, and G Gnudi. Extraction of discriminant features in post-cardiosurgical intensive care units. *Int J Biomed Comput*, 39(3):349–358, Jun 1995.

[10] SJ Aukburg, PH Ketikidis, DS Kitz, TG Mavrides, and BB Matschinsky. Automation of physiologic data presentation and alarms in the post anesthesia care unit. In *Symposium on Computer Applications in Medical Care*, pages 580–582. American Medical Informatics Association, 1989.

[11] GJ Balm. Crosscorrelation techniques applied to the electrocardiogram interpretation problem. *IEEE Transactions on Bio-Medical Engineering*, BME-14(4):258–262, Oct 1967.

[12] SJ Barker and NK Shah. Effects of motion on the performance of pulse oximeters in volunteers. *Anesthesiology*, 85(4):774–781, Oct 1996.

[13] K Becker, B Thull, H Kasmacher-Leidinger, J Stemmer, G Rau, G Kalff, and H-J Zimmermann. Design and validation of an intelligent patient monitoring and alarm system based on a fuzzy logic process model. *Artif Intell Med*, 11(1):33–53, Sep 1997.

[14] R Bellazzi, C Larizza, P Magni, S Montani, and G De Nicolao. *Intelligent analysis of clinical time series by combining structural filtering and temporal abstractions*, pages 261–270. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[15] JEW Beneken and JJ van der Aa. Alarms and their limits in monitoring. *J Clin Monit*, 5(3):205–210, 1989.

[16] AM Benis, HL Fitzkee, RA Jurado, and RS Litwak. Improved detection of adverse cardiovascular trends with the use of a two-variable computer alarm. *Crit Care Med*, 8(6):341–344, Jun 1980.

[17] LR Bentt, TA Santora, BJ Leverle, M LoBue, and MM Shabot. Accuracy and utility of pulse oximetry in the surgical intensive care unit. *Current Surgery*, pages 267–268, Jul-Aug 1990.

[18] MI Bierman, KI Stein, and JV Snyder. Pulse oximetry in the postoperative care of cardiac surgical patients. *Chest*, 102(5):1367–1370, Nov 1992.

[19] G Bontempi, M Birattari, and H Bersini. Local learning for iterated time series prediction. In *Machine Learning Proceedings of the Sixteenth International Conference (ICML '99). I Bratko, S Dzeroski, eds.*, pages 32–38, 1999.

[20] BE Boser, IM Guyon, and VN Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of Fifth Annual ACM Workshop on Computational Learning Theory*, pages 144–152, 1992.

[21] A Bosnjak, G Bevilacqua, G Passariello, F Mora, B Sanso, and G Carrault. An approach to intelligent ischaemia monitoring. *Med & Biol Eng & Comput*, 33:749–756, November 1995.

[22] DL Bowton, PE Scuderi, L Harris, and EF Haponik. Pulse oximetry monitoring outside the intensive care unit: Progress or problem? *Ann Intern Med*, 115:450–454, 1991.

[23] RN Bracewell. *The Fourier transform and its applications*. McGraw-Hill Book Company, 1986.

[24] KE Bradshaw. Computerized alerting system warns of life-threatening events. In *Symposium on Computer Applications in Medical Care*, pages 403–410. American Medical Informatics Association, 1986.

[25] RD Branson. Monitoring ventilator function. *Critical Care Clinics*, 11(1):127–143, 1995.

[26] WL Briggs and VE Henson. *The DFT an owner's manual for the discrete Fourier transform*. Society for Industrial and Applied Mathematics, 1995.

[27] CJC Burges. Simplified support vector decision rules. In L Saitta, editor, *Macine Learning, Proceedings of the 13th International Conference (ICML '96)*, pages 71–77, 1996.

[28] D Calvelo, M Chambrin, D Pomorski, and P Ravaux. *ICU patient state characterization using machine learning in a time series framework*, pages 356–360. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[29] C Cao and N McIntosh. Empirical study on artifact identification in clinical monitoring data [abstract]. In *Proceedings of the American Medical Informatics Association Fall Symposium*, 1998. (in press).

[30] MC Chambrin, P Ravaux, C Chopin, J Mangalaboyi, P Lestavel, and F Fourrier. Computer-assisted evaluation of respiratory data in ventilated critically ill patients. *Int J Clin Monit Comput*, 6(4):211–215, Dec 1989.

[31] CH Chen, LF Pau, and PSP Wang, editors. *Handbook of pattern recognition & computer vision*. World Scientific, 1993.

[32] HJ Chizeck. Modelling, simulation and control in a data rich environment. In *Symposium on Computer Applications in Medical Care*, pages 65–69. American Medical Informatics Association, 1986.

[33] CK Chui and G Chen. *Kalman Filtering with Real-Time Applications*. Springer-Verlag, 1987.

[34] PT Chui and T Gin. False alarms and the integrated alarm system: report of a potential hazard [letter]. *Anesth Analg*, 79:192–193, 1994.

[35] Coalition for Critical Care Excellence. Standards of evidence for the safety and effectiveness of critical care monitoring devices and related interventions. *Crit Care Med*, 23(10):1756–1763, 1995.

[36] MH Coen. Design principles for intelligent environments. In *Proceedings Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 547–554, 1998.

[37] E Coiera. Intelligent monitoring and control of dynamic physiological systems. *Artificial Intelligence in Medicine*, 5:1–8, 1993.

[38] JR Colvin and GNC Kenny. Microcomputer-controlled administration of vasodilators following cardiac surgery: technical considerations. *J Cardiothorac Anesth*, 3(1):10–15, Feb 1989.

[39] C Combi, L Portoni, and F Pinciroli. *Visualizing temporal clinical data on the WWW*, pages 301–311. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[40] GF Cooper, CF Aliferis, R Ambrosino, J Aronis, BG Buchanan, R Caruana, MJ Fine, C Glymour, G Gordon, BH Hanusa, JE Janosky, C Meek, T Mitchell, T Richardson, and P Spirtes. An evaluation of machine-learning methods for predicting pneumonia mortality. *Artificial Intelligence in Medicine*, 9:107–138, 1997.

[41] WM Coplin, GE O'Keefe, MS Grady, GA Grant, KS March, HR Winn, and AM Lam. Accuracy of continuous jugular bulb oximetry in the intensive care unit. *Neurosurgery*, 42(3):533–539, Mar 1998.

[42] C Cortes and V Vapnik. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[43] AD Crew, KDC Stoodley, R Lu, S Old, and M Ward. Preliminary clinical trials of a computer-based cardiac arrest alarm. *Intensive Care Med*, 17:359–364, 1991.

[44] G Das, K Lin, H Mannila, G Renganathan, and P Smyth. Rule discovery from time series. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 16–22, 1998.

[45] NAM de Beer, M van de Velde, and PJM Cluitmans. Clinical evaluation of a method for automatic detection and removal of artifacts in auditory evoked potential monitoring. *J Clin Monit*, 11:381–391, 1995.

[46] P Guedes de Oliveira, JP Cunha, and A Martins da Silva. The role of computer based techniques in patient monitoring: technical note. *Acta Neurochir*, 55 Suppl:18–20, 1992.

[47] D DeCoste. Mining multivariate time-series sensor data to discover behavior envelopes. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. D Heckerman, H Mannila, D Pregibon, R Uthurusamy, eds.*, pages 151–154, 1997.

[48] TG Dietterich. *Learning to predict sequences*, pages 63–106. Morgan Kaufmann Pulishers, Inc., 1986. In: Machine Learning. An Artificial Intelligence Approach. Volume II. RS Michalski, JG Carbonell, TM Mitchell, eds.

[49] M Dojat, N Ramaux, and D Fontaine. Scenario recognition for temporal reasoning in medical domains. *Artificial Intelligence in Medicine*, 14:139–155, 1998.

[50] G Dorffner, E Leitgeb, and H Koller. *A comparison of linear and non-linear classifiers for the detection of coronary artery disease in stress-ECG*, pages 227–231. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[51] RO Duda and PE Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[52] C Dumas, JA Wahr, and KK Tremper. Clinical evaluation of a prototype motion artifact resistant pulse oximeter in the recovery room. *Anesth Analg*, 83:269–272, 1996.

[53] JH Eichorn, JB Cooper, DJ Cullen, WR Maier, JH Philip, and RG Seeman. Standards for patient monitoring during anesthesia at Harvard Medical School. *JAMA*, 256:1017–1020, 1986.

[54] J Fackler, C Tsien, W Beatty, and SM Zimmerman. Experimental design: one observation out-of-specification limits system versus SPC methods for patient vital sign management. In *Proceedings of International Conference on Industry, Engineering, and Management Systems*, 1997.

[55] M Factor, DF Sittig, and AI Cohn. A parallel software architecture for building intelligent medical monitors. In *Symposium on Computer Applications in Medical Care*, pages 11–16. American Medical Informatics Association, 1989.

[56] RM Farrell, JA Orr, K Kuck, and DR Westenskow. Differential features for a neural network based anesthesia alarm system. *Biomed Sci Inst*, 28:99–104, 1992.

[57] RM Farrell, JA Orr, and DR Westenskow. Improving the performance of a neural network based alarm system. *J Clin Monit*, 9(3):223–224, 1993.

[58] T Fawcett and F Provost. Activity monitoring: noticing interesting changes in behavior. In *Proceedings The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. S Chaudhuri and D Madigan, eds.*, pages 53–62, 1999.

[59] U Fayyad, G Piatetsky-Shapiro, and P Smyth. Knowledge discovery and data mining: towards a unifying framework. In *Proceedings Second International Conference on Knowledge Discovery and Data Mining. E Simoudis, J Han, U Fayyad, eds.*, pages 82–88, 1996.

[60] GA Finley and AJ Cohen. Perceived urgency and the anaesthetist: responses to common operating room monitor alarms. *Canadian Journal of Anesthesia*, 38:958–964, 1991.

[61] PM Ford, DJ Hoodless, and GR Plant. Respiratory alarm monitor. *Lancet*, 2(7666):246–247, Aug 1 1970.

[62] W Friedsdorf, S Konichezky, F Gross-lltag, A Fattroth, and B Schwilk. Data quality of bedside monitoring in an intensive care unit. *Int J Clin Mon Comp*, 11(2):123–128, 1994.

[63] M Frize, FG Solven, M Stevenson, B Nickerson, T Buskard, and K Taylor. Computer-assisted decision support systems for patient management in an intensive care unit. In *MEDINFO 95 Proceedings*, pages 1009–1012. IMIA, 1995.

[64] DA Frost, FG Yanowitz, and TA Pryor. Evaluation of a computerized arrhythmia alarm system. *Am J Cardiol*, 39(4):583–587, Apr 1977.

[65] Y Fukui and T Masuzawa. Knowledge-based approach to intelligent alarms. *J Clin Monit*, 5(3):211–216, 1989.

[66] D Gamberger, N Lavrac, and C Groselj. Experiments with noise filtering in a medical domain. In *Machine Learning Proceedings of the Sixteenth International Conference (ICML '99). I Bratko, S Dzeroski, eds.*, pages 143–151, 1999.

[67] D Garfinkel, PV Matsiras, JH Lecky, SJ Aukburg, BB Matschinsky, and TG Mavrides. PONI: an intelligent alarm system for respiratory and circulation management in the operating rooms. In *Symposium on Computer Applications in Medical Care*, pages 13–17. American Medical Informatics Association, 1988.

[68] D Garfinkel, PV Matsiras, T Mavrides, J McAdams, and SJ Aukburg. Patient monitoring in the operating room: validation of instrument reading by artificial intelligence methods. In *Symposium on Computer Applications in Medical Care*, pages 575–579. American Medical Informatics Association, 1989.

[69] GL Gibby and GA Ghani. Computer-assisted doppler monitoring to enhance detection of air emboli. *J Clin Monit*, 4(1):64–73, Jan 1988.

[70] I Ginzburg and D Horn. Combined neural networks for time series anaylsis. In *Advances in Neural Information Processing Systems 6*, pages 224–231, 1994. J Cowan, G Tesauro and J Alspector, eds.

[71] F Girosi. An equivalence between sparse approximation and support vector machines. A.I. Memo/C.B.C.L. Paper 1606/147, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, and Center for Biological and Computational Learning, 1997.

[72] JM Goldman and MJ Cordova. Advanced clinical monitoring: considerations for real-time hemodynamic diagnostics. In *Proceedings – the Annual Symposium on Computer Applications in Medical Care*, pages 752–756. American Medical Informatics Association, 1994.

[73] EE Gose. *Introduction to biological and mechanical pattern recognition*, pages 203–252. Academic Press, NY, 1969. In: 'Methodologies of Pattern Recognition,' S Watanabe, ed.

[74] SH Grieve, N McIntosh, and IA Laing. Comparison of two different pulse oximeters in monitoring preterm infants. *Crit Care Med*, 25(12):2051–2054, Dec 1997.

[75] V Guralnik and J Srivastava. Event detection from time series data. In *Proceedings The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. S Chaudhuri and D Madigan, eds.*, pages 33–42, 1999.

[76] V Guralnik, D Wijesekera, and J Srivastava. Pattern directed mining of sequence data. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 51–57, 1998.

[77] J Hahnel, W Friesdorf, B Schwilk, T Marx, and S Blessing. Can a clinician predict the technical equipment a patient will need during intensive care unit treatment? An approach to standardize and redesign the intensive care unit workstation. *J Clin Monit*, 8:1–6, 1992.

[78] IJ Haimowitz. Intelligent diagnostic monitoring using trend templates. In *Symposium on Computer Applications in Medical Care*, pages 702–708. American Medical Informatics Association, 1994.

[79] GL Hall and PB Colditz. Continuous physiological monitoring: an integrated system for use in neonatal intensive care. *Australasian Physical & Engineering Sciences in Medicine*, 18(3):139–142, 1995.

[80] J Han, W Gong, and Y Yin. Mining segment-wise periodic patterns in time-related databases. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 214–218, 1998.

[81] JA Hanley and BJ McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.

[82] PJ Haug, RM Gardner, KE Tate, RS Evans, TD East, G Kuperman, TA Pryor, SM Huff, and HR Warner. Decision support in medicine: examples from the HELP system. *Computers and Biomedical Research*, 27:396–418, 1994.

[83] B Hayes. Ventilation and ventilators–an update [review]. *Journal of Medical Engineering & Technology*, pages 197–218, Sep 1988.

[84] B Hayes-Roth, S Uckun, JE Larsson, and J Drakopoulos. Guardian: an experimental system for intelligent ICU monitoring [abstract]. In *Proceedings - the Annual Symposium on Computer Applications in Medical Care*, page 1004. American Medical Informatics Association, 1994.

[85] DJ Hitchings, MJ Campbell, and DEM Taylor. Trend detection of pseudo-random variables using a exponentially mapped past statistical approach: an adjunct to computer assisted monitoring. *Int J Biomed Comput*, 6(2):73–88, Apr 1975.

[86] J Hunter and N McIntosh. *Knowledge-based event detection in complex time series data*, pages 271–280. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[87] J Irazuzta. Monitoring in pediatric intensive care [review]. *Indian J Pediatr*, 60:55–65, 1993.

[88] JA Orr. *An anesthesia alarm system based on neural networks.* PhD thesis, The University of Utah, June 1991.

[89] JE Jacobs, FJ Lewis, and RL Stout. A hybrid computer for use in patient monitoring. *Biomed Sci Instrum*, 3:207–214, 1967.

[90] Computers and clinicians. *JAMA*, 196(2):29–36, Apr 11 1966.

[91] R Jans, AG Elder, RD Jones, WR Fright, and PJ Bones. A low cost ECG central station for intensive care. *Australas Phys Eng Sci Med*, 13(1):31–35, Mar 1990.

[92] M Jastremski, C Jastremski, M Shepherd, V Friedman, D Porembka, R Smith, E Gonzales, D Swedlow, H Belzberg, R Crass, T Jannett, E Richards III, D Thys, and D Woods. A model for technology assessment as applied to closed loop infusion systems. *Crit Care Med*, 23(10):1745–1755, 1995.

[93] RE Jensen, H Shubin, PF Meagher, and MH Weil. On-line computer monitoring of the seriously-ill patient. *Med Biol Eng*, 4(3):265–272, May 1966.

[94] JJ van der Aa. *Intelligent alarms in anesthesia*. PhD thesis, May 1990.

[95] D Jones, A Lawson, and R Holland. Integrated monitor alarms and 'alarm overload'. *Anaesth Intens Care*, 19:101–102, 1991.

[96] JM Bonifacio Jr, AM Cansian, ACPLF de Carvalho, and ES Moreira. Neural networks applied in intrusion detection systems. In *The 1998 IEEE International Joint Conference on Neural Networks Proceedings*, pages 205–210, 1998.

[97] MW Kadous. Learning comprehensible descriptions of multivariate time series. In *Machine Learning Proceedings of the Sixteenth International Conference (ICML '99). I Bratko, S Dzeroski, eds.*, pages 454–463, 1999.

[98] DT Kaplan. The analysis of variability. *J Cardiovasc Electrophysiol*, 5:16–19, 1994.

[99] RD Keith, J Westgate, GW Hughes, EC Ifeachor, and KR Greene. Preliminary evaluation of an intelligent system for the management of labour. *Journal of Perinatal Medicine*, 22(4):345–350, 1994.

[100] RL Kennedy, HS Fraser, LN McStay, and RF Harrison. Early diagnosis of acute myocardial infarction using clinical and electrocardiographic data at presentation: derivation and evaluation of logistic regression models. *EHJ*, 17:1181–1191, 1996.

[101] E Keogh and P Smyth. A Probabilistic approach to fast pattern matching in time series databases. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. D Heckerman, H Mannila, D Pregibon, R Uthurusamy, eds.*, pages 24–30, 1997.

[102] EJ Keogh and MJ Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 239–243, 1998.

[103] JH Kerr. Warning devices. *Br J Anaesth*, 57:696–708, 1985.

[104] IG Kestin, BR Miller, , and LH Lockart. Auditory alarms during anesthesia monitoring. *Anesthesiology*, 69:106–109, 1988.

[105] PH King and BE Smith. Computerized monitoring at Vanderbilt University status and future directions. *Int J Clin Mon Comp*, 8:117–120, 1991.

[106] NN Finer KJ Barrington and CA Ryan. Evaluation of pulse oximetry as a continous monitoring technique in the neonatal intensive care unit. *Crit Care Med*, 16:1147–1153, 1988.

[107] MA Klaas and EY Cheng. Early response to pulse oximetry alarms with telemetry. *J Clin Monit*, 10(3):178–180, May 1994.

[108] IS Kohane and IJ Haimowitz. Hypothesis-driven data abstraction with trend templates. In *Proceedings – the Annual Symposium on Computer Applications in Medical Care*, pages 444–448. American Medical Informatics Association, 1993.

[109] EMJ Koski, A Makivirta, T Sukuvaara, and A Kari. Frequency and reliability of alarms in the monitoring of cardiac postoperative patients. *Int J Clin Mon Comp*, 7:129–133, 1990.

[110] EMJ Koski, A Makivirta, T Sukuvaara, and A Kari. Development of an expert system for haemodynamic monitoring: computerized symbolization of on-line monitoring data. *Int J Clin Mon Comp*, 8:289–293, 1992.

[111] EMJ Koski, A Makivirta, T Sukuvaara, and A Kari. Clinicians' opinions on alarm limits and urgency of therapeutic responses. *Int J Clin Monit Comput*, 12(2):85–88, May 1995.

[112] EMJ Koski, T Sukuvaara, A Makivirta, and A Kari. A knowledge-based alarm system for monitoring cardiac operated patients–assessment of clinical performance. *Int J Clin Mon Comp*, 11:79–83, 1994.

[113] L Kukolich and R Lippmann. *LNKnet User's Guide*. MIT Lincoln Laboratory, 1995.

[114] G Laffel, R Luttman, and S Zimmerman. Using control charts to analyze serial patient-related data. *Quality Management in Health Care*, 3(1):70–77, 1994.

[115] CL Lake. American monitoring: standards and state of the art. *Infusionstherapie und Transfusionsmedizin*, 20(3):104–110, Jun 1993.

[116] K Lakshminarayan, SA Harp, R Goldman, and T Samad. Imputation of missing data using machine learning techniques. In *Proceedings Second International Conference on Knowledge Discovery and Data Mining. E Simoudis, J Han, U Fayyad, eds.*, pages 140–145, 1996.

[117] N Lavrac. Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine*, 16:3–23, 1999.

[118] ST Lawless. Crying wolf: False alarms in a pediatric intensive care unit. *Crit Care Med*, 22(6):981–985, 1994.

[119] WA Lea. *Speech recognition: past, present, and future*, pages 39–98. Prentice-Hall, Inc, 1980. In: Trends in Speech Recognition. WA Lea, ed.

[120] W Lee, SJ Stolfo, and KW Mok. Mining audit data to build intrusion detection models. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 66–72, 1998.

[121] W Lee, SJ Stolfo, and KW Mok. Mining in a data-flow environment: experience in network intrusion detection. In *Proceedings The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. S Chaudhuri and D Madigan, eds.*, pages 114–124, 1999.

[122] PR L'Estrange, AR Blowers, RG Carlyon, and SL Karlsson. A microcomputer system for physiological data collection and analysis. *Australian Dental Journal*, 38(5):400–405, 1993.

[123] A Makivirta, E Koski, A Kari, and T Sukuvaara. The median filter as a preprocessor for a patient monitor limit alarm system in intensive care. *Computer Methods and Programs in Biomedicine*, 34:139–144, 1991.

[124] A Makivirta and EMJ Koski. Alarm-inducing variability in cardiac postoperative data and the effects of prealarm delay. *J Clin Monit*, 10(3):153–162, May 1994.

[125] JF Martin. Closed-loop control of arterial pressure during cardiac surgery [letter]. *Journal of Clinical Monitoring*, 8(3):252–255, 1992.

[126] JP Mathews. How to use an automated vital signs monitor. *Nursing*, pages 60–64, February 1991.

[127] JWR McIntyre. Alarms in the operating room. *Canadian Journal of Anaesthesia*, 38(8):951–953, 1991.

[128] JWR McIntyre and TM Nelson. Application of automated human voice delivery to warning devices in an intensive care unit: a laboratory study. *Int J Clin Mon Comp*, 6:255–262, 1989.

[129] JWR McIntyre and LM Stanford. *Ergonomics and anaesthesia: Auditory alarm signals in the operating room*, pages 81–86. Springer-Verlag, 1985. In: 'Anaesthesia: innovation in management', R Droh, W Erdmann and R Spintge, eds.

[130] CL Mee. Ventilator alarms. How to respond with confidence. *Nursing*, pages 61–64, July 1995.

[131] C Meredith and J Edworthy. Are there too many alarms in the intensive care unit? An overview of the problems. *Journal of Advanced Nursing*, 21:15–20, 1995.

[132] PhGH Metnitz, P Laback, C Popow, O Laback, K Lenz, and M Hiesmayr. Computer assisted data analysis in intensive care: the ICDEV project - development of a scientific database system for intensive care. *Int J Clin Mon Comp*, 12:147–159, 1995.

[133] C Meyer. Visions of tomorrow's ICU. *American Journal of Nursing*, pages 27–31, May 1993.

[134] Yves Meyer, editor. *Wavelets and applications*. Springer-Verlag, 1992.

[135] Yves Meyer. *Wavelets algorithms & applications*. Society for Industrial and Applied Mathematics, 1993. Translated and revised by RD Ryan.

[136] A Meyer-Falcke, R Rack, F Eichwede, and P-J Jansing. How noisy are anaesthesia and intensive care medicine? Quantification of the patients' stress. *European Journal of Anaesthesiology*, 11:407–411, 1994.

[137] D Michie, DJ Spiegelhalter, and CC Taylor, editors. *Machine learning, neural and statistical classification*. Ellis Horwood, 1994.

[138] S Miksch, A Seyfang, W Horn, and C Popow. *Abstracting steady qualitative descriptions over time from noisy, high-frequency data*, pages 281–290. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[139] TM Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[140] K Momtahan, R Hetu, and B Tansley. Audibility and identification of auditory alarms in the operating room and intensive care unit. *Ergonomics*, 36(10):1159–1176, 1993.

[141] JH Moore. Artificial intelligence programming with LabVIEW: genetic algorithms for instrumentation control and optimization. *Computer Methods and Programs in Biomedicine*, 47:73–79, 1995.

[142] CJ Morgan, J Takala, D DeBacker, T Sukuvaara, and A Kari. Definition and detection of alarms in critical care. *Computer Methods and Programs in Biomedicine*, 51(1–2):5–11, Oct 1996.

[143] K Morik, P Brockhausen, and T Joachims. Combining statistical learning with a knowledge-based approach – a case study in intensive care monitoring. In *Machine Learning Proceedings of the Sixteenth International Conference (ICML '99). I Bratko, S Dzeroski, eds.*, pages 268–277, 1999.

[144] P Mormino, P Palatini, A Di Marco, G Sperti, L Cordone, E Casiglia, AC Pessina, and C Dal Palu. Computer analysis of continuous direct blood pressure recording. *Clin Exp Hypertens*, 7(2–3):455–461, 1985.

[145] RM Nelson, HR Warner, RE Gardner, and JD Mortensen. Computer based monitoring of patients following cardiac surgery. *Isr J Med Sci*, 5(4):926–930, Jul 1969.

[146] VI Nenov, W Read, and D Mock. Compute applications in the intensive care unit. *Neurosurgery Clinics of North America*, 5(4):811–827, Oct 1994.

[147] MR Neuman. *Pulse oximetry: physical principles, technical realization and present limitations*, volume 220, pages 135–144. 1987. In: A Huch, R Huch and G Rooth, eds.

[148] J Nobel. Physiologic monitoring systems, acute care. *Pediatric Emergency Care*, 8(4):235–237, 1992.

[149] T Oates. Identifying distinctive subsequences in multivariate time series by clustering. In *Proceedings The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. S Chaudhuri and D Madigan, eds.*, pages 322–326, 1999.

[150] TM O'Carroll. Survey of alarms in an intensive therapy unit. *Anaesthesia*, 41:742–744, 1986.

[151] JA Orr and DR Westenskow. Evaluation of a breathing circuit alarm system based on neural networks [abstract]. *Anesthesiology*, 73(3A):A445, Sep 1990.

[152] JA Orr and DR Westenskow. A breathing circuit alarm system based on neural networks. *Journal of Clinical Monitoring*, 10(2):101–109, 1994.

[153] E Osuna, R Freund, and F Girosi. Training support vector machines: an application to face detection. In *Proceedings of CVPR '97*, 1997.

[154] B Padmanabhan and A Tuzhilin. Pattern discovery in temporal databases: a temporal logic approach. In *Proceedings Second International Conference on Knowledge Discovery and Data Mining. E Simoudis, J Han, U Fayyad, eds.*, pages 351–354, 1996.

[155] P Pan and C James. Effects of default alarm limit settings on alarm distribution in telemetric pulse oximetry network in ward setting [abstract]. *Anesthesiology*, 75:A405, 1991.

[156] D Pappert, R Rossaint, K Lewandowski, R Kuhlen, H Gerlach, and KJ Falke. Preliminary evaluation of a new continuous intra-arterial blood gas monitoring device. *Acta Anaesthesiologica Scandinavica. Supplementum*, 107:67–70, 1995.

[157] P Perner, TB Belikova, and NI Yashunskaya. *Knowledge acquisition by symbolic decision tree induction for interpretation of digital images in radiology*, pages 208–219. Springer-Verlag, 1996. In: Advances in Structural and Syntactical Pattern Recognition. LNCS 1121. P Perner, P Wang, and A Rosenfeld, eds.

[158] E Pesonen, M Eskelinen, and M Juhola. Treatment of missing data values in a neural network based decision support system for acute abdominal pain. *Artificial Intelligence in Medicine*, 13:139–146, 1998.

[159] HV Pipberger, FW Stallmann, K Yano, and HW Draper. Digital computer analysis of the normal and abnormal electrocardiogram. *Progress in Cardiovascular Diseases*, 5(4):378–392, January 1963.

[160] PE Plsek. Tutorial: introduction to control charts. *Quality Management in Health Care*, 1(1):65–74, 1992.

[161] JX Polaschek, GW Rutledge, SK Andersen, and LM Fagan. Using belief networks to interpret qualitative data in the ICU. *Respiratory Care*, 38(1):60–71, 1993.

[162] M Pontil and A Verri. Properties of support vector machines. A.I. Memo/C.B.C.L. Paper 1612/152, Massachusetts Institute of Technology, Artificial Intelligence Laboratory, and Center for Biological and Computational Learning, 1997.

[163] DJ Price. The use of computers in neurosurgical intensive care. *Balliere's Clin Anaesthesiology*, 1:533–556, 1987.

[164] JR Quinlan. Comparing connectionist and symbolic learning methods. Published source unknown. Paper received from author.

[165] JR Quinlan. Unknown attribute values in induction. In *Proceedings of the Sixth International Workshop on Machine Learning. AM Segre, ed.*, pages 164–168, 1989.

[166] JR Quinlan. *C4.5 Programs for machine learning*. Morgan Kaufmann Publishers, 1993.

[167] IJ Rampil. *Intelligent detection of artifact*, pages 175–190. Butterworths, Boston, 1987. In: 'The automated anesthesia record and alarm systems,' JS Gravenstein, RS Newbower, AK Ream and NT Smith, eds.

[168] RB Rao, S Rickard, and F Coetzee. Time series forecasting from high-dimensional data with multiple adaptive layers. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 319–323, 1998.

[169] E Rehse, M Kramer, HJ Stahl, and K Muller. Problems of equipment failure and signal disturbances in computer aided monitoring. *Acta Anaesthesiologica Belgica*, 23 Suppl:239–240, 1975.

[170] AT Rheineck-Leyssius and CJ Kalkman. Influence of pulse oximeter lower alarm limit on the incidence of hypoxaemia in the recovery room. *British Journal of Anaesthesia*, 79:460–464, 1997.

[171] JD Ridges, WJ Sanders, and DC Harrison. The on-line analysis of atrial pressures using a digital computer. *Comput Biomed Res*, 6(2):196–208, Apr 1973.

[172] RM Farrell. *A neural network based anesthesia alarm system.* PhD thesis, The University of Utah, December 1995.

[173] RA Rosenbaum, BJ Levine, and TA Sweeney. Another false alarm? apnea monitor activation in a neonatal intensive care unit graduate. *J Emerg Med*, 15(6):855–858, Nov 1997.

[174] S Rosset, U Murad, E Neumann, Y Idan, and G Pinkas. Discovery of fraud rules for telecommunications – challenges and solutions. In *Proceedings The Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. S Chaudhuri and D Madigan, eds.*, pages 409–413, 1999.

[175] EB Rudy and A Grenvik. Future of critical care. *American Journal of Critical Care*, 1:33–37, 1992.

[176] WB Runciman, RK Webb, L Barker, and M Currie. The Australian incident monitoring study. The pulse oximeter: applications and limitations–an analysis of 2000 incident reports. *Anaesth Intens Care*, 21:543–550, 1993.

[177] GW Rutledge, SK Andersen, JX Polaschek, and LM Fagan. A belief network model for interpretation of ICU data. Report KSL-90-49, Stanford University, Knowledge Systems Laboratory, 1990.

[178] RM Sailors and TD East. A model-based simulator for testing rule-based decision support systems for mechanical ventilation of ARDS patients [abstract]. In *Proceedings - the Annual Symposium on Computer Applications in Medical Care*, page 1007. American Medical Informatics Association, 1994.

[179] M Sanklecha. The pulse oximeter [letter]. *Indian Journal of Pediatrics*, 60(3):469–470, May-Jun 1993.

[180] CA Sara and HJ Wark. Disconnection: an appraisal. *Anesthesia and Intensive Care*, 14:448–452, 1986.

[181] LM Schnapp and NH Cohen. Pulse oximetry. Uses and abuses. *Chest*, 98:1244–1250, 1990.

[182] PJ Schreiber and J Schreiber. Structured alarm systems for the operating room. *J Clin Monit*, 5:201–204, 1989.

[183] PE Scuderi, DL Bowton, RL Anderson, and DS Prough. Pulse oximetry: Would further technical alterations improve patient outcome? *Anesth Analg*, 74:177–180, 1992.

[184] AV Sebald, M Quinn, NT Smith, A Karimi, G Schnurer, and S Isaka. Engineering implications of closed-loop control during cardiac surgery. *J Clin Monit*, 6:241–248, 1990.

[185] JW Severinghaus and JF Kelleher. Recent developments in pulse oximetry. *Anesthesiology*, 76:1018–1038, 1992.

[186] MM Shabot, M LoBue, BJ Leyerle, and SB Dubin. Inferencing strategies for automated ALERTS on critically abnormal laboratory and blood gas data. In *Symposium on Computer Applications in Medical Care*, pages 54–57. American Medical Informatics Association, 1989.

[187] BA Shapiro. Quality improvement standards for intensive care unit monitors: we must be informed and involved. *Crit Care Med*, 20(12):1629–1630, Dec 1992.

[188] LC Sheppard. Computer control of blood and drug infusions in patients following cardiac surgery. *J Biomed Eng*, 2(2):83–84, Apr 1980.

[189] LC Sheppard and JW Kirklin. Cardiac surgical intensive care computer system. *Fed Proc*, 33(12):2326–2328, Dec 1974.

[190] H Shubin and MH Weil. Efficient monitoring with a digital computer of cardiovascular function in seriously ill patients. *Ann Intern Med*, 65(3):453–460, Sep 1966.

[191] B Sierra, N Serrano, P Larranaga, EJ Plasencia, I Inza, JJ Jimenez, JM De la Rosa, and ML Mora. *Machine learning inspired approaches to combine standard medical measures at an intensive care unit*, pages 366–371. Springer-Verlag, 1999. In: AIMDM'99. LNAI 1620. W Horn et al., eds.

[192] RL Simpson. Automating the ICU: facing the realities. *Nursing Management*, 23(3):24, 26, 1992.

[193] DF Sittig and M Factor. Physiologic trend detection and artifact rejection: a parallel implementation of a multi-state kalman filtering algorithm. In *Symposium on Computer Applications in Medical Care*, pages 569–574. American Medical Informatics Association, 1989.

[194] BE Smith. Universities and the clinical monitoring industry: feckless independents or fruitful partners? *Int J Clin Mon Comp*, 7:249–258, 1990.

[195] KDC Stoodley, DR Walker, AD Crew, and JS Marshall. Problems in the development of a computerized ward monitoring system for a paediatric intensive care unit. *Int J Clin Mon Comp*, 8:281–287, 1992.

[196] T Sukuvaara, EMJ Koski, A Makivirta, and A Kari. A knowledge-based alarm system for monitoring cardiac operated patients–technical construction and evaluation. *Int J Clin Monit Comput*, 10:117–126, 1993.

[197] Y Sun, I Kohane, and AR Stark. Fuzzy logic assisted control of inspired oxygen in ventilated newborn infants. In *Proceedings – the Annual Symposium on Computer Applications in Medical Care*, pages 757–761. American Medical Informatics Association, 1994.

[198] NL Szaflarski. Emerging technology in critical care: continuous intra-arterial blood gas monitoring [review]. *American Journal of Critical Care*, 5:55–65, 1996.

[199] P Szolovits. *Knowledge-based systems*, pages 317–370. MIT Press, 1991. In: 'Research Directions in Computer Science: An MIT Perspective,' AR Meyer, JV Guttag, RL Rivest and P Szolovits, eds.

[200] R Thomas. A PC-based monitoring and alarm system. *Am Biotechnol Lab*, 10:38–39, Oct 1992.

[201] A Timcenko and DL Reich. Real-time expert system for advising anesthesiologists in the cardiac operating room [abstract]. In *Proceedings – the Annual Symposium on Computer Applications in Medical Care*, page 1005. American Medical Informatics Association, 1994.

[202] KM Ting. The characterisation of predictive accuracy and decision combination. In *Machine Learning Proceedings of the Thirteenth International Conference (ICML '96). L Saitta, ed.*, pages 498–506, 1996.

[203] JH Tinker, DL Dull, RA Caplan, RJ Ward, and FW Cheney. Role of monitoring devices in prevention of anesthetic mishaps: A closed claims analysis. *Anesthesiology*, 71:541–546, 1989.

[204] M Topf and E Dillon. Noise-induced stress as a predictor of burn-out in critical care nurses. *Heart Lung*, 17:567–574, 1988.

[205] CL Tsien. Reducing false alarms in the intensive care unit: a systematic comparison of four algorithms. In *Proceedings 1997 AMIA Annual Fall Symposium*, page 894, 1997.

[206] CL Tsien and J Fackler. Poor prognosis for existing monitors in the intensive care unit. *Crit Care Med*, 25(4):614–619, 1997.

[207] CL Tsien, HS Fraser, and IS Kohane. *LRTree: a hybrid technique for classifying myocardial infarction data containing unknown attribute values*, pages 409–411. Springer-Verlag, 1998. In: Research and Development in Knowledge Discovery and Data Mining. X Wu, R Kotagiri, KB Korb, eds.

[208] CL Tsien, HSF Fraser, WJ Long, and RL Kennedy. Using classification tree and logistic regression methods to diagnose myocardial infarction. In *Proceedings of the Ninth World Congress on Medical Informatics*, pages 493–497, 1998.

[209] S Tsumoto and H Tanaka. Automated discovery of medical expert system rules from clinical databases based on rough sets. In *Proceedings Second International Conference on Knowledge Discovery and Data Mining. E Simoudis, J Han, U Fayyad, eds.*, pages 63–69, 1996.

[210] S Uckun. Model-based reasoning in biomedicine. *Critical reviews in biomedical engineering*, 19(4):261–292, 1992.

[211] S Uckun. Intelligent systems in patient monitoring and therapy management. *Int J Clin Mon Comp*, 11:241–253, 1994.

[212] C Ulbricht, G Dorffner, and A Lee. Neural networks for recognizing patterns in cardiotocograms. *Artificial Intelligence in Medicine*, 12:271–284, 1998.

[213] AR Visram, RDM Jones, MG Irwin, and J Bacon-Shone. Use of two oximeters to investigate a method of movement artefact rejection using photoplethysmographic signals. *British Journal of Anaesthesia*, 72:388–392, 1994.

[214] HR Warner. Experiences with computer-based patient monitoring. *Anesth Analg*, 47(5):453–462, Sep 1968.

[215] HR Warner, RM Gardner, and AF Toronto. Computer-based monitoring of cardiovascular functions in postoperative patients. *Circulation*, 37(4 Suppl):II68–II74, Apr 1968.

[216] RK Webb. Medical decision making and decision anaylsis. *Anaesthesia and intensive care*, 16(1):107–109, 1988.

[217] DE Weese-Mayer and JM Silvestri. Documented monitoring: an alarming turn of events. *Clinics in Perinatology*, 19(4):891–906, December 1992.

[218] GM Weiss and H Hirsh. Learning to predict rare events in event sequences. In *Proceedings The Fourth International Conference on Knowledge Discovery and Data Mining. R Agrawal, P Stolorz, and G Piatetsky-Shapiro, eds.*, pages 359–363, 1998.

[219] Sholom M. Weiss and Nitin Indurkhya. *Predictive Data Mining A Practical Guide.* Morgan Kaufmann Publishers, Inc., 1998.

[220] SM Weiss and I Kapouleas. An empirical comparison of pattern recognition, neural nets, and machine learning classification methods. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.

[221] DR Westenskow, JA Orr, FH Simon, H-J Bender, and H Frankenberger. Intelligent alarms reduce anesthesiologist's response time to critical faults. *Anesthesiology*, 77:1074–1079, 1992.

[222] AP White. *Probabilistic induction by dynamic path generation in virtual trees*, pages 35–46. Cambridge University Press, 1987. In: Research and Development in Expert Systems III. MA Bramer, ed.

[223] L Wiklund, B Hok, K Stahl, and A Jordeby-Jonsson. Postanesthesia monitoring revisited: frequency of true and false alarms from different monitoring devices. *J Clin Anesth*, 6:182–188, May/Jun 1994.

[224] JL Willems, C Abreu-Lima, P Arnaud, JH van Bemmel, C Brohet, R Degani, B Denis, J Gehring, I Graham, G van Herpen, H Machado, PW Macfarlane, J Michaelis, SD Moulopoulos, P Rubel, and C Zywietz. The diagnostic performance of computer programs for the interpretation of electrocardiograms. *N Engl J Med*, 325:1767–1773, 1991.

[225] S Wilson. Conscious sedation and pulse oximetry: False alarms? *Pediatr Dent*, 12:228–232, 1990.

[226] M Wolf, M Keel, K von Siebenthal, and H-U Bucher. Improved monitoring of preterm infants by fuzzy logic. *Technol Health Care*, 4(2):193–201, Aug 1996.

[227] H Yien, S Hseu, L Lee, TBJ Kuo, T Lee, and SHH Chan. Spectral analysis of systemic arterial pressure and heart rate signals as a prognostic tool for the prediction of patient outcome in the intensive care unit. *Crit Care Med*, 25(2):258–266, 1997.

[228] WH Young, RM Gardner, TD East, and K Turner. Computerized ventilator data selection: artifact rejection and data reduction. *International Journal of Clinical Monitoring and Computing*, 7:1–12, 1997.

[229] W Zhang. A Region-based learning approach to discovering temporal structures in data. In *Machine Learning Proceedings of the Sixteenth International Conference (ICML '99). I Bratko, S Dzeroski, eds.*, pages 484–492, 1999.

[230] Y Zhang and CL Tsien. A data collection and data processing system to support real-time trials of intelligent alarm algorithms. In *Proceedings of the American Medical Informatics Association Fall Symposium*, 1998.

[231] H-J Zimmermann. *Fuzzy sets in pattern recognition*, pages 383–391. Springer-Verlag, 1987. In: Pattern Recognition Theory and Applications. PA Devijver and J Kittler, eds.

[232] MA Zissman. Automatic language identification of telephone speech. *Lincoln Laboratory Journal*, 8(2):115–144, Fall 1995.

[233] B Zupan, N Lavrac, and E Keravnou. Data mining techniques and applications in medicine. *Artificial Intelligence in Medicine*, 16:1–2, 1999.

About the Author

Christine L. Tsien was born in Minneapolis, Minnesota. After graduation in 1987 from Mounds View High School in Arden Hills, Minnesota, she attended the Massachusetts Institute of Technology where she majored in Computer Science and Engineering, with a humanities concentration in Russian language. After completing her Bachelor of Science degree in 1991, she worked on her master's thesis project at Hewlett Packard Laboratories in Palo Alto, California, earning her Master of Science degree in 1993. At that time, she joined the MIT Laboratory for Computer Science Clinical Decision Making Group ('MEDG'), where she was first introduced to the field of artificial intelligence in medicine.

In 1994, she began her medical studies at Harvard Medical School in the Division of Health Sciences and Technology. For the next six years, she divided her time amongst medical coursework, research in ICU monitoring, and clinical clerkships. During that time, she also published five peer-reviewed first-author articles, seven abstracts, and four essays, as well as competed in over 20 amateur ballroom dance competitions around the country and in England. She also supervised several MIT undergraduate research students, and served as a resident advisor for an MIT undergraduate living group.

After finishing her M.D. in June 2001, she plans to train in emergency medicine. Her long term goals are to both care for patients and continue research in the area of medical artificial intelligence. She is a member of the Massachusetts Medical Society, American Medical Association, American Medical Informatics Association, and American Association for Artificial Intelligence.