

SEARCHING FOR NON-SENSE:
IDENTIFICATION OF
PACEMAKER NON-SENSE AND NON-CAPTURE FAILURES
USING MACHINE LEARNING TECHNIQUES

by

Michele Rae Bizub Malinowski

A Thesis submitted to the
Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of
the Requirements for
the Degree of
Master of Science in Electrical Engineering
Milwaukee, Wisconsin
May, 2003

Acknowledgement

Many thanks to all of you; you know who you are.

Table of Contents

<u>1</u>	<u>INTRODUCTION</u>	<u>1</u>
1.1	PROBLEM STATEMENT	1
1.1.1	MOTIVATION	2
1.1.2	REQUIREMENTS FOR THE ALGORITHM	2
1.1.3	DEFINITION OF FAILURES	5
1.2	OUTLINE	6
<u>2</u>	<u>BACKGROUND ON PACEMAKERS & CARDIAC ACTIVITY</u>	<u>7</u>
2.1	HEALTHY PATIENT	7
2.2	NORMAL PACED PATIENT	10
2.3	PACEMAKER TYPES	11
2.4	NON-SENSE FAILURE	12
2.5	NON-CAPTURE FAILURE	13
<u>3</u>	<u>HISTORICAL REVIEW</u>	<u>15</u>
3.1	TIME INTERVAL ANALYSIS	15
3.2	BIOMEDICAL SIGNAL ANALYSIS	17
3.3	CURRENT TECHNOLOGY	18
3.3.1	PUBLISHED RESEARCH	18
3.3.2	PATENT SEARCH	20

3.3.3	DATA AND PREPROCESSING	21
4	<u>METHODS</u>	23
4.1	DATA AND PREPROCESSING	23
4.2	TYPES OF FAILURES AND NON-FAILURES	24
4.3	FEATURES	26
4.4	RULE-BASED CLASSIFIER	29
4.5	K-NEAREST NEIGHBORS	30
4.6	THRESHOLD-BASED CLASSIFIER	32
4.7	STATISTICAL PATTERN RECOGNITION	34
5	<u>APPLICATION AND EXPERIMENTS</u>	37
5.1	CROSS-VALIDATION	37
5.2	K-NEAREST NEIGHBORS	38
5.3	THRESHOLD-BASED CLASSIFIER	40
5.4	HYBRID RULE-BASED AND BAYESIAN CLASSIFIER	44
5.4.1	RULE-BASED LEARNERS	47
5.4.2	FALSE FAILURES	48
5.4.3	MISSED FAILURES	50
5.4.4	NOTES ON IMPLEMENTATION	52
6	<u>CONCLUSIONS & FUTURE RESEARCH</u>	55
6.1	FUTURE RESEARCH	56

REFERENCES

1 Introduction

Between 200,000 and 300,000 patients worldwide have artificial cardiac pacemakers implanted on an annual basis; about 115,000 of these patients live in the United States [1]. These patients rely upon the pacemaker to maintain an active, independent life. Abnormal or unexpected function of pacemakers due to mechanical failure of the implantation, electrical failures of the battery and electrodes, or physiological failures to respond to the stimulus may cause harm to the patient. A method to detect two types of pacemaker failures, non-sense and non-capture, is proposed in this thesis.

1.1 *Problem Statement*

The goal of this research is to develop an automatic method for identifying pacemaker failures from time series data related to the patient's electrocardiogram (ECG) without prior knowledge of the type or model of the pacemaker. The application for the proposed algorithm is a patient monitoring system used in a hospital, transport, or emergency response environment.

Two types of pacemaker failures are investigated: non-sense (failure to detect a naturally occurring heartbeat) and non-capture (failure to stimulate the heart sufficiently to produce a paced heartbeat). A trained physician easily recognizes these failures, but manually searching and annotating thousands of heartbeats is a tedious task. It would be beneficial for a patient monitoring system to automatically detect these failures and alert a clinician, enabling him/her to review the electrocardiogram and determine whether adjustments to the pacemaker are required.

1.1.1 Motivation

Detection of non-sense and non-capture failures is desirable because these events precipitate from a malfunctioning electrical-physiological system involving the pacemaker and the patient's heart. If the patient does not exhibit symptoms of occasional non-capture, the condition may worsen over time. Additionally, a pacemaker failing to capture in a pacemaker dependent patient (one whose heart does not beat spontaneously) can lead to fatalities [2-4]. A pacemaker failing to sense may discharge at inappropriate times, causing fibrillation (an uncoordinated and ineffective heart rhythm), leading to further harm to the patient [1, 4]. Non-sense and non-capture failures are further discussed in Sections 2.4 and 2.5, respectively. Detection of non-sense and non-capture by the patient monitoring system will provide earlier notification to the clinician when a cardiologist or pacemaker-programming device is not available to diagnose the condition.

False alarms are a significant problem with patient monitoring systems. Clinicians tend to distrust systems that alarm at every unrecognized pattern on an ECG. This distrust may lead to alarm volume reductions or ignored alarms, potentially causing a clinician not to respond to an actual life-threatening event. The algorithm proposed here must recognize this requirement, and must carefully weigh false alarms against missed events.

1.1.2 Requirements for the Algorithm

One requirement for the proposed algorithm is the ability to implement this algorithm in existing patient monitoring products. Due to this requirement, the algorithm is limited in the signals it can use, as well as both time and space complexity. Existing patient monitors vary greatly in processor speed and memory size, from the equivalent of

a first generation personal computer (PC) (circa 1980) to the equivalent of a low-performance modern PC.

The patient monitoring system is assumed to already measure the cardiac data from surface electrodes on a patient. Before applying the various detection algorithms, the system performs data pre-processing and filtering. The proposed algorithm will use data from pacemaker discharge and heartbeat detection algorithms. A sequence of time-stamped markers is provided by the monitoring system, as shown in Figure 1.1.

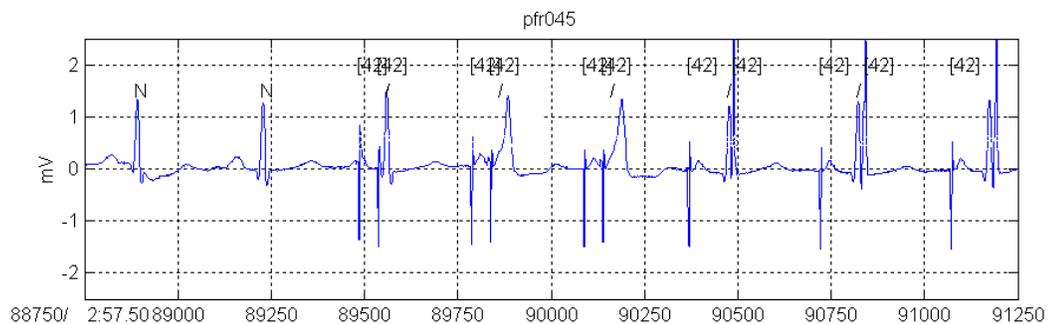


Figure 1.1 ECG Strip with annotations visible (5 seconds of data)

The annotations labeled on Figure 1.1 represent spontaneous heartbeats (N), pacemaker discharges ([42]), and paced heartbeats (/). The numbers along the x-axis represent the sample number in the data file (e.g. 88750 and 89000); and the time elapsed from the beginning of the recording (2:57.50). Along the y-axis is an amplitude scale in mV of the actual ECG signal. This particular sample was taken from patient “pfr045” (a designation the hospital assigned for the particular bed). Other ECG strips in this document have had the annotations removed to present the waveform more clearly. For simplicity, the algorithm assumes these annotations are correct, and does not use the directly measured ECG signal from the electrodes.

Implementation of this algorithm in a patient monitor requires prompt alarming within a reasonable period of time from the event. A common industry standard for time to alarm is compliance with *AAMI EC13:2002 – Cardiac monitors, heart rate meters, and alarms*. According to AAMI EC13, the limit for cardiac standstill alarms (heart stops beating), maximum elapsed time between the event occurring and alarm occurring is 10 seconds [5]. The 10-second alarm is considered acceptable, given the much longer response time for a clinician to reach the patient's bedside, evaluate the patient's condition, and respond to the medical crisis. Due to the pre-processing and other algorithms operating on the patient monitor, a reasonable time limit for this algorithm to process a failure and alert the clinician is less than one-half second using the hardware available in a patient monitoring system. This limit will allow a failure to be processed without interfering with other, more critical alarm processing that may occur simultaneously.

The processor used in the majority of patient monitoring systems is considerably slower than a standard PC of 2003 vintage. These systems do have microprocessors within them, but the variance of capabilities is broad. On the lowest end of the spectrum are monitors using 16-bit 68000-series processors operating around 20 MHz with 4 MB of RAM, and 4 MB of static memory. At the upper end of the spectrum are 32-bit PowerPC systems operating at 75 MHz with 64 MB of RAM, and 16 MB of static memory. Current patient monitoring systems utilize about 80% to 90% of the processing capabilities on a low-end system, and 25% to 50% on a high-end system. However, some of this is overhead for displaying data to a monitor and operating other features. Due to these hardware limitations, any additional algorithm should not strain the system

resources to the point of interfering with current functions. Although there is no specific conversion between a personal computer and an embedded microprocessor, these systems are similar to the capabilities available on an Intel 286 personal computer and an Intel 486 personal computer. Hence, the proposed algorithm must be able to run on the equivalent of an IBM 80486 DX2 system operating at 66 MHz with 8MB of memory. A benchmarking algorithm is used to estimate performance of the algorithm on the equivalent 80486 DX2, 66 MHz system, based upon measurements taken of computation time on the development platform. This benchmarking is further described in Section 5.4.4.

1.1.3 Definition of Failures

For the purposes of this research, the term *failure* refers to an error in the combined electrical-physiological system involving an artificial pacemaker and a human heart. Failures include a pacemaker behaving as expected, but the heart responding inadequately, or not at all. Failures also include pacemakers that function according to their design, but with an undesirable result. Finally, a failure can occur from a faulty pacemaker.

The term *data interval* will be used to describe a single set of measurements used by the algorithm and describes one point of the set. One data interval consists of two successive ventricular contractions, and all events that occur between the two.

For identification of errors in the algorithm, the following terms will be used:

Normal – a data interval of all events occurring between two QRS complexes (heartbeats) that has been correctly labeled normal.

True Failure – a data interval of all events occurring between two QRS complexes that has been correctly labeled non-sense or correctly labeled non-capture.

False Failure – a data interval of all events occurring between two QRS complexes that has been mistakenly labeled non-sense or non-capture.

Missed Failure – a data interval of all events occurring between two QRS complexes that has been mistakenly labeled normal, but is actually a non-sense or non-capture episode.

1.2 Outline

This thesis is divided into six chapters. Chapter 2 provides a brief introduction to the cardiac electrical conduction system, pacemaker function, and normal and abnormal activity of the heart. The background illustrates the specific failure modes this research will address.

Chapter 3 provides a historical review of the methods currently used to detect patterns within electrocardiograms followed by a discussion of the current technology including results from a patent search.

Chapter 4 describes the methods used in this research. Each technique is explained and discussed in the context of pattern detection in electrocardiogram data.

Chapter 5 presents the experimental procedures and results. The techniques presented in Chapter 4 are applied to the data, and results are provided. A brief description of clinical usefulness of the results is included, as well as a discussion of sensitivity versus specificity.

Chapter 6 reviews the thesis, results, and discusses future research.

2 Background on Pacemakers & Cardiac Activity

This section describes the electrical principles that govern cardiac activity and the electrical conduction system of the human heart. A discussion of pacemaker function is provided, as well as an explanation of each type of pacemaker failure addressed by this research. Electrocardiogram examples provided in this document are from different patients, and may not appear uniform due to normal inter-patient physiological differences.

2.1 Healthy Patient

An electrocardiogram (ECG or EKG) is a graphical record of the electrical activity of the heart. The electrical stimulus begins in the sino-atrial (SA) node, and travels through the atrial myocardium to the atrio-ventricular (AV) node. This initial impulse causes the deflection identified as the *P wave* and represents the electrical activation of the atria [1, 6].

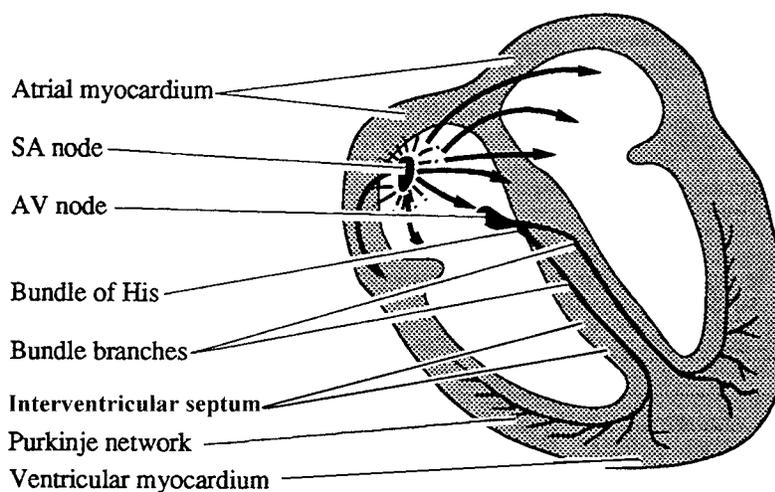


Figure 2.1 Electrical activity within the human heart [6]

The impulse reaches the AV node, and slowly travels through the node to create a delay between atrial and ventricular contraction. Upon leaving the AV node, the impulse travels quickly through the bundle of His, bundle branches, and Purkinje network. The Purkinje network located at the bottom of the heart muscle directs the impulse to the ventricular myocardium. Figure 2.1 shows the intra-cardiac conduction system.

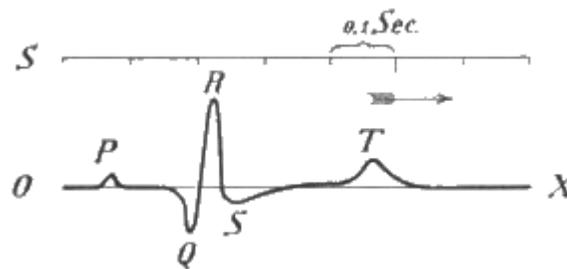


Figure 2.2 Normal ECG Complex [7]

The activation of the interventricular septum by the bundle branches causes the negative deflection identified as the *Q wave*. Next, the conduction through the ventricular myocardium causes the ventricles to contract and is represented by the largest deflection identified as the *R wave*. The interval between a P wave and an R wave is approximately 0.12-0.20 seconds [8]. The *S wave* represents the topmost areas of ventricular muscle stimulation, which are activated slightly later than the majority of the myocardium. Finally, the ventricles repolarize, generating a *T wave*. The interval between the Q wave and the T wave is heart rate dependant [8]. The group consisting of a P, Q, R, S, and T wave is referred to as an *ECG complex* and represents one full cycle of cardiac activity. The group of a Q, R, and S wave is referred to as a *QRS complex*, or simply a *QRS*, and represents the electrical activity associated with the ventricular contractions. A QRS complex is typically 0.06-0.10 seconds [8]. Figure 2.2 illustrates a

normal ECG complex (O and X identify the time axis only, not cardiac events, *S* is a time scale in divisions of 0.1 seconds).

The ECG is recorded through surface electrodes placed on the patient's skin. The placement of electrodes varies with the purpose of the ECG. Different electrode placements will provide different *views* (leads) of the electrical activity. Electrodes may be placed on the patient's chest, limbs, torso, back, or a combination of locations based upon the view desired. The most common configurations of electrodes allow the computation of three leads in the form of an equilateral triangle around the heart, known as Einthoven's triangle, illustrated in Figure 2.3, below.

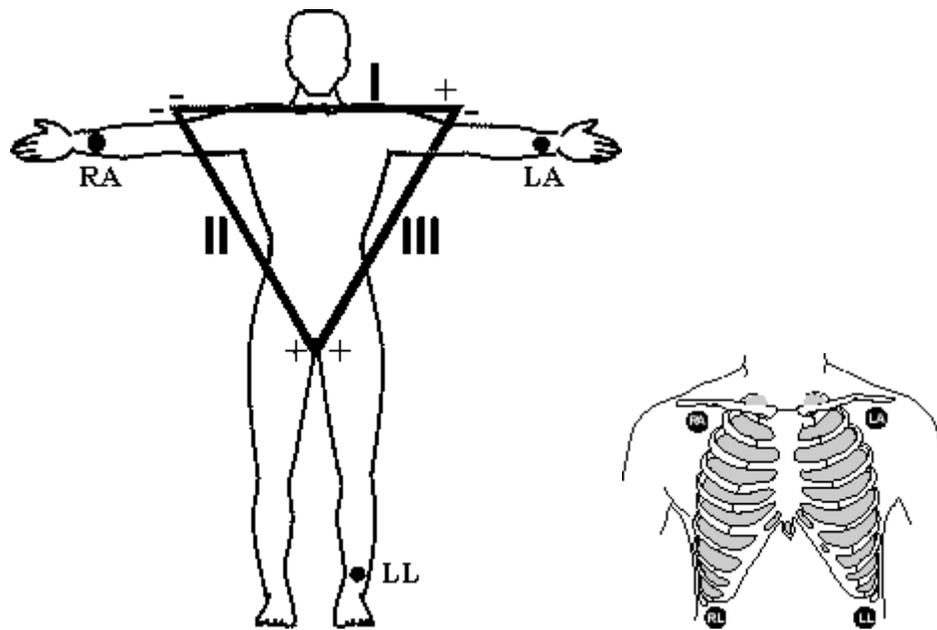


Figure 2.3 Einthoven's triangle and the limb lead locations[9, 10]

Lead I is oriented horizontally, right arm (-) to left arm (+). Lead II is oriented parallel to the interventricular septum, right arm (-) to left leg (+). Lead III is oriented from left arm (-) to left leg (+). Positive deflections on the ECG are the result of an impulse traveling towards the positive (+) electrode of a lead, while negative deflections

are caused by impulses traveling towards the negative (-) electrode. An impulse traveling perpendicular to the lead orientation produces no deflection of the ECG [4]. The sum of Lead I and Lead III is equal to Lead II. Unless otherwise noted, all ECG examples in this report are of Lead I, recorded from surface electrodes.

2.2 Normal Paced Patient

An ECG complex that is not triggered by an artificial pacemaker is called a Normal Sinus Rhythm, because the Sino-Atrial node generates the activity. The SA node is the natural pacemaker of the heart, generating the impulse that triggers cardiac activity. If the SA node fails, the AV node or other cells will adopt the role of primary pacemaker. The AV node, bundle of His, bundle branches, and Purkinje network conduct the impulse throughout the heart. If any of these fail, the electrical stimulus is lost or disrupted and the heart will not work as efficiently or effectively as it should.

If an artificial pacemaker is used to trigger the cardiac activity, the beat is considered a Paced Beat. An artificial pacemaker can be used to replace or augment a malfunctioning node or cardiac conduction system and artificially stimulate the heart. The artificial pacemaker is implanted with leads inserted into the heart muscle at locations suitable to compensate for the injury to the muscle. A ventricular pacemaker usually has the lead located at the apex of the right ventricle to generate ventricular contractions. An atrial pacemaker usually has the lead implanted where the SA node normally stimulates the right atrium [1, 3]. The pacemaker may be programmed to discharge for every beat or only discharge when the heart does not spontaneously beat.

In a normal paced patient, the pacemaker discharges and causes the atria and ventricles to contract in a prescribed fashion, mimicking the natural function of the

patient's heart. This results in blood circulation throughout the body. The ECG example in Figure 2.4 shows a patient with normal heart rhythm, and a single-chamber (ventricular) pacemaker spike (identified by the arrow), followed by the patient's QRS complex (the ventricular contraction).

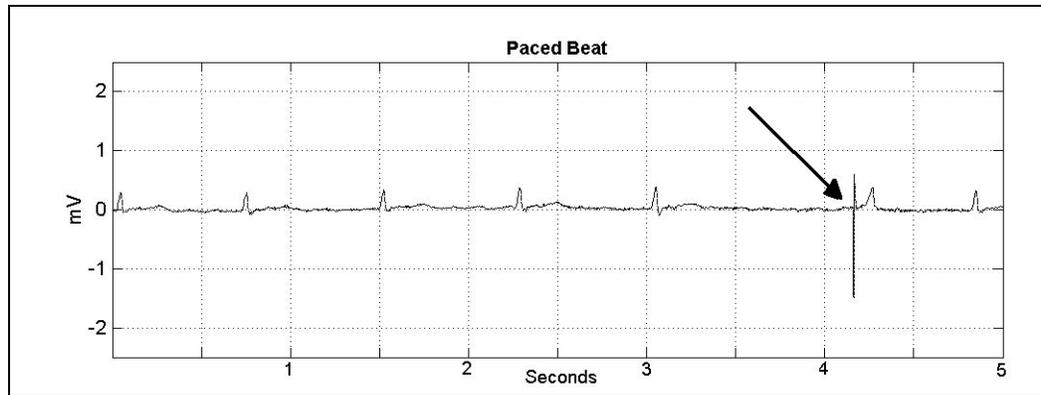


Figure 2.4 Normal ECG with a single paced beat

2.3 Pacemaker Types

This section provides a brief discussion of pacemaker types. While this describes several of the most common types, it is not an exhaustive list. Pacemakers described in this section include single and dual chamber pacing; single and dual chamber sensing; fixed rate; rate adaptive; and implantable cardioverter pacemakers. A pacemaker may have more than one of these qualities, for example a cardioverter pacemaker may be rate adaptive, dual pacing, and dual sensing.

A pacemaker with *single chamber pacing and sensing* has one electrode / lead implanted within either the atria or ventricle (usually on the right side of the heart). Single pacing allows the pacemaker to generate an electrical stimulus for either the atria

(in the event of an SA node malfunction) or the ventricles (in the event of an AV node, or bundle branch malfunction). Sensing pacemakers will detect if spontaneous electrical activity occurs within the chamber in which the lead is implanted, and inhibit pacemaker discharge if appropriate.

A dual chamber pacing and sensing pacemaker has two electrodes or leads implanted within the heart. Dual chamber pacing applies stimuli to both atria and ventricles. Dual sensing allows the pacemaker to determine if spontaneous electrical activity occurs within either the atria or ventricles.

A fixed rate pacemaker can be programmed to one fixed value by the cardiologist, but cannot change the rate itself. This value determines the rate of discharge, and subsequently, the patient's heart rate. Programming can occur by placing a programming device on the skin of the patient over the implanted pacemaker and sending the appropriate communication signals.

Rate adaptive pacemakers can vary their discharge rate based upon demand for increased circulation (respiration increases while pacemaker patient is running, causes the pacemaker to increase heart rate). The most common method for detecting this demand is an increase in respiration rate, but other methods exist.

Implantable *cardioverter pacemakers* have the ability to provide a defibrillating shock to the patient if they detect a potentially fatal arrhythmia, in addition to standard pacemaker functionality.

2.4 Non-Sense Failure

A pacemaker in non-sense mode fails to detect physiological cardiac activity within the heart, and discharges. This situation causes a "hiccup" reaction of the heart,

and discomfort to the patient. Further, a pacemaker discharging during the re-polarization of the heart (the T-wave) can initiate ventricular fibrillation [1, 4]. In the example below, the QRS complex is present, but the pacemaker still discharges, initiating a second QRS. The first QRS has very low amplitude and is identified by comparison to other beats earlier on the strip (not visible in this image). The arrow points to the apex of the first QRS.

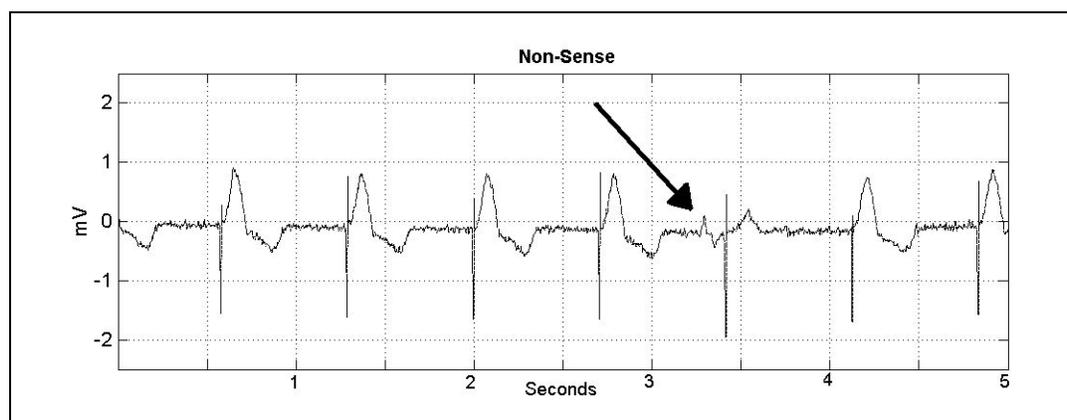


Figure 2.5 Patient with a normal QRS not sensed by the pacemaker

2.5 Non-Capture Failure

A pacemaker in non-capture mode discharges but fails to create a physiological response in the cardiac muscle. Thus, the pacemaker is working, but the patient is not receiving proper circulation. This case usually is corrected by increasing the amplitude of the pacer output. The worst-case scenario is fibrillation (non-synchronous beating of the heart) or asystole (no electrical activity of the heart), but the pacemaker continues to discharge as if nothing is wrong. With the combination pacemaker/cardioverter devices prescribed for some patients, this may delay a life-saving defibrillation. In Figure 2.6, the patient experiences an episode of non-capture. This patient has a dual-chamber

pacemaker (two pacemaker impulses are visible), but one pace does not produce any physiological response of the heart. After the first pace, a P-wave is visible; after the second, the QRS is absent, as noted in the figure.

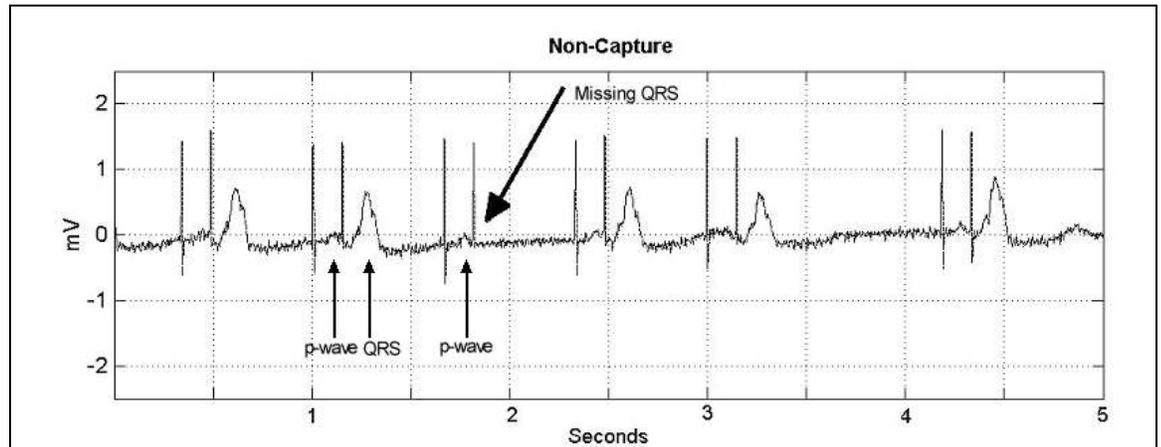


Figure 2.6 Patient with a dual-chamber pacemaker, exhibiting one episode of non-capture

3 Historical Review

This chapter discusses the current technology used to identify heart arrhythmias and pacemaker failures. Heart arrhythmia detection is included because the methods are similar to those used in this research, and little research has been published on non-sense and non-capture pacemaker failures. After discussions of time interval analysis and biomedical signal analysis techniques, a brief assessment of current technology is presented with both published research and patent information.

3.1 *Time Interval Analysis*

The ECG data is a representation of electrical cardiac events along a temporal scale. The heartbeat represents a set of P, Q, R, S, and T waves, each with significance to the condition and function of the heart. The most common measure of the heart function is the heart rate – the number of beats in one minute. This time interval provides a clinician with a readily available measure of how well the heart is performing. Further investigation of the ECG strip presents many time intervals to illustrate more specifically how the heart is functioning.

Time interval analysis techniques are used to analyze signals that contain structure. The structure in these signals may remain in a constant state until an event occurs, signaling a potential failure or abnormality. [11] Through measurement of specific intervals between states of an ECG, (e.g. the interval between two QRS complexes), changes in heart function can be identified (an increase or decrease in heart rate).

A benefit of choosing time intervals to represent the data allows the algorithm to define a level of similarity between a previously learned state and a new event. This similarity is essential to properly classify signals that do not repeat patterns exactly. Pacemakers failing to sense may show several different morphologies of the same failure mechanism, illustrated in

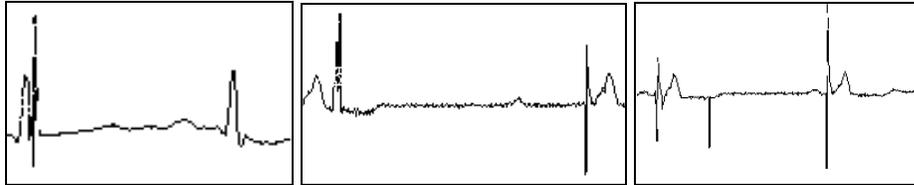


Figure 3.1 Three examples of non-sense morphologies

These morphologies are similar in some of their time-interval data, but rarely appear identically in the ECG strip. The time-interval properties allow a classification based upon the similarities of these cases of non-sense.

This research includes an investigation of threshold, statistical, and nearest neighbor searches for classification of ECG data intervals. Thresholding sets a defined limit for a time interval and dichotomizes the data based upon those that fall above or below the threshold. Statistical classification uses a model of the statistical distribution to determine which class is most likely. A nearest neighbor search identifies a particular data interval by those other intervals that have the closest measurements. Each of these techniques is described in detail in Section 4.

Time interval analysis provides a mechanism for the patient monitor to interpret what the clinician sees, and classify the data interval as normal or failure.

3.2 *Biomedical Signal Analysis*

Little research is published on non-sense and non-capture identification; therefore a similar field of research was reviewed for implementation ideas. The physiological similarity of the paced heart rhythm to a normal sinus rhythm leads to using ECG research as a starting point. A great deal of information in ECG-based arrhythmia detection and classification exists, some of which is suitable for non-sense and non-capture identification. Further, biomedical signals in general have similar time-based properties, and methods found in other biomedical research may be useful for detecting non-sense and non-capture failures in paced ECG rhythms.

The techniques presented in this research have been used in ECG signal classification successfully. Nearest neighbor searching, implemented using feature intervals of the ECG, has been used to identify arrhythmias and abnormalities. [12-14]. Bayesian statistical classifiers and threshold-based classifiers have been used to identify variations in QRS duration and beat classification [15, 16]. Other statistical tools have been suggested for interpreting ECG data and categorizing arrhythmias [17].

Several other techniques are being investigated by the research community, and may provide insights to problems encountered in this research. Several authors have investigated phase space reconstruction and chaotic methods for identification of arrhythmias and other biomedical signal patterns [18-21]. Zurro, et al. investigate frequency-based techniques to detect P waves within the ECG [22]. A tree-based technique for ECG classification, in which the degree of mismatch between two trees determines the normal and abnormal waveforms, is presented by Parthasarathy, et al [23].

3.3 Current Technology

Due to the lack of published reports on the detection of non-sense and non-capture, care must be taken to select appropriate sources for comparison. Few published papers investigate the problem of detecting non-sense and non-capture pacemaker failures; two are described in Section 3.3.1. Additionally, a search of patent records is discussed in Section 3.3.2 to provide further insight into current technology.

3.3.1 Published Research

One method to determine pacemaker function through threshold-based classifiers has been published by J. Bai and J. Lin [24]. The application for this method is a telemonitoring system for pacemaker patients in secluded areas, unable to travel to a hospital for routine pacemaker checkups. Data is recorded by a Holter ambulatory monitoring system and fed into the classifier algorithm for processing. A self-learning beat classifier is applied to the ECG signal, which defines the beats according to what type of pacemaker they represent (single vs. dual pacing, single vs. dual sensing, etc.), from a list of the most common pacemakers used in China. Once the pacemaker type is identified, the manufacturers' specifications are used to determine whether each beat is a pacemaker malfunction, cardiac response malfunction, or normal operation. Malfunctions were not limited to non-sense and non-capture episodes, but included several other types of cardiac disease. The results for correct classification of normal beats was 98.6% and for abnormal beats was 93.3%, with the majority of errors caused by noise on the ECG signal. Average processing time for the algorithm (beat classification, preprocessing and recognition) on an IBM PC 486 compatible computer was 50 ms per beat.

A second method for assessing pacemaker function in Holter ambulatory recordings is presented by S. Ghiringhelli, et al [25]. This approach analyzes the data after the entire recording has completed. The algorithm begins by creating a distribution of pacemaker discharge to QRS intervals, to determine whether a dual chamber or single chamber pacemaker is present. A second analysis of the data classifies the pacemaker discharges as atrial, ventricular, or possible malfunctions; and the paced QRS complexes as atrial, ventricular, or cardiac malfunction. The statistical distributions of the pacemaker discharge to QRS complex intervals are saved for analysis of the data. Pacemaker discharges that occur after a spontaneous QRS (non-sense failures) are handled separately. Features used by the classifier in this research included: sensed and paced heart chambers, pace-to-pace interval, R-to-pace interval, and pace-to-R interval. Results for this algorithm are reported as 92% correct classification for all cases combined.

Both of the methods identified in published research require knowledge of the types of pacemakers that are used for the experiment, determined either by machine learning based on a finite set of choices, or provided by an expert rule. The method presented by S. Ghiringhelli, et al, did learn the pacemaker type itself, but required multiple scans of the Holter data to determine if a failure was present. Considering the goal of this research is to determine a method that can easily be implemented in a patient monitoring system, it must work in real-time and without *a priori* knowledge of the pacemaker type or specifications.

3.3.2 Patent Search

Several patents discuss capture detection implemented within pacemakers through the use of time and amplitude thresholds. In U.S. Patent 6477422, V. Splett presents a capture detection algorithm to be implemented in a pacemaker that begins sensing for capture immediately following discharge of a pacemaker [26]. A physician or learning algorithm programs thresholds for minimum amplitude of cardiac response amplitude and time to respond. Events exceeding these thresholds are deemed non-capture. M. Gryzwa and Q. Zhu discuss a circuit for capture verification while eliminating false responses due to noise, residual polarization of electrodes, and artifacts on the sensed signal in U.S. Patent 6473649 [27]. M. Hemming, *et al.*, discusses the use of negative peak tracking and slope polarity changes to eliminate false capture detections, again using the pacemaker discharge as a primary reference point in U.S. Patent 5954756 [28].

Additionally, S. Marinello presents a patient monitoring system with a capture detection method in U.S. Patent 5771898 [29]. The method presented uses a logic network to determine the type of beat encountered and then applies thresholds to determine whether the heart is effectively captured by the pacemaker. This implementation detects overshoot and ringing caused by the pacemaker and appropriately inhibits QRS detection until these conditions subside. Single and dual chamber pacemakers are distinguished by another threshold of time between pacemaker discharges.

The patent search reveals the preferred method for detecting capture as compliance with a predefined or learned threshold. Only one patient monitor patent was found, but Marinello's work is similar in application to the problem presented by this

research. The main difference between this research and the patent is the use of timestamps for calculating the likelihood of non-capture or non-sense as opposed to the logic network and thresholds used by Marinello.

3.3.3 Data and Preprocessing

All commercially marketed ECG systems manipulate raw ECG data from the electrodes prior to applying it to the detection and identification functions. This preprocessing allows the system to filter noisy input, to normalize magnitudes for digitizing, or to improve the integrity of the ECG signal.

Generally, the first manipulation of the ECG signal within a medical device is amplification. A surface ECG normally can detect cardiac activity with amplitudes of 0.5 to 5.0 mV [5, 8], but recordings outside these ranges occur with electrode placement and conduction variances. Amplification allows the hardware to obtain finer granularity while converting the analog signal into a digital signal for the microprocessor to manipulate and decipher.

Modern patient monitors and ECG recording systems implement various filtering mechanisms on the raw data taken from the electrodes. Common filtering mechanisms include: 50Hz/60Hz notch filters for power frequency noise [30, 31]; high-pass filters to remove respiration interference in the 0 to 0.5 Hz range; and low-pass filters greater than 30 Hz to remove muscle tremors and other non-cardiac activity [32, 33]. Additionally, special filters may be implemented to remove specific interference such as an electrosurgical generator (e.g. Bovie knife) [34].

The ECG signal recorded from the electrodes is then processed for beat detection and classification. At this point, detection algorithms are implemented to identify QRS

complexes, arrhythmias, pacemaker discharges, defibrillator discharges, noise, physiological artifacts, and other signals of interest. The methods used for detection vary widely and may include both digital and analog sensing mechanisms [13, 33, 35-37].

Pacemaker discharges can be identified by the rapid increase in slew-rate of the signal. Often, a slew-rate limiting circuit is implemented in hardware to restrict the input signal from overloading the amplifiers. In some devices, this slew-rate limiting circuit is accompanied by a switch that turns off the amplifiers and introduces a pacemaker representation generated by the device itself. This representative pulse of a pacemaker is similar to the actual impulse, but not identical, and should not be used to identify if the output of a pacemaker is appropriately shaped.

The preprocessing techniques described here are commonly found in the medical device industry. These techniques are widely accepted and documented in product literature from several manufacturers of patient monitors (GE Medical Systems, Phillips, HP, etc.). The preprocessing methods discussed above are already implemented in the anticipated patient monitoring system application for the proposed algorithm.

4 Methods

This chapter describes the methods used within this research beginning with a discussion of the data used and associated preprocessing. Identification of failures and non-failures follows, with illustrations and descriptions of several possible data patterns. The identifying features of the dataset are then introduced and explained. The chapter ends with four sections discussing machine learning methods that are used in the research: rule-based learning, k-nearest neighbors searching, threshold-based classification, and statistical pattern recognition. This chapter provides the background for Chapter 5, Application and Experiments.

4.1 Data and Preprocessing

GE Medical Systems – *Information Technologies*, provided data for this study. Research performed at Universitaetsklinik Freiburg (Freiburg, Germany) collected ECGs from 34 pacemaker patients, with a total of 5785 R-to-R intervals. Pacemaker devices and modes included devices with single and dual chamber pacing; single and dual chamber sensing; fixed rate and rate adaptive.

Data collected for each patient included a 12-lead ECG recorded from surface ECG electrodes through a GE Medical Systems –*Information Technologies* CardioSys Exercise Testing System V3.01, with the patients at rest. The hardware detects the pacemaker pulses by the high slew rate and replaces them with generated marker pulses representative of the actual pulse. The purpose of this replacement is to shut off the sensitive preamplifiers in the ECG circuitry during a potentially harmful slew rate input, and reduce recovery time to baseline.

All data was stored in the Massachusetts Institute of Technology-Beth Israel Hospital (MIT-BIH) Database format for ECG data [38]. The CardioSys Exercise Testing System software classifies the beats and annotates the patient's ECG file. The data contains annotations for pacemaker pulses and ECG annotations. Detected pulses and annotations were manually checked and corrected (if necessary) by a medical doctor.

For this research, the data was processed into individual data intervals with labels of *Normal*, *Non-Sense*, or *Non-Capture*. Each data interval contains information about all events occurring between two QRS complexes. Labeling the data in this manner provides future extension to a real-time algorithm, which allows for a new data interval to be classified upon identification of the next QRS. Non-sense and non-capture modes were manually labeled, as the CardioSys either labeled these as "unknown" beats or had no event to label. In the 34-patient data set, 13 cases of non-sense and 20 cases of non-capture were identified and labeled, with 5752 normal data intervals.

4.2 Types of Failures and non-Failures

This section expresses the complexity of the problem by illustrating many types of possible patterns within the data. Examples with no pacemaker discharges and those with more than two pacemaker discharges in one data interval are omitted because an expert rule classifies these conditions, described in detail in Section 4.4, Rule-Based . The data intervals are divided into two categories based upon the number of pacemaker discharges present within the R-to-R interval.

Tables Table 4.1 Table 4.2 show a representation of a data interval in annotated form (R = QRS complex; p = pacemaker discharge) with a section of ECG recording. The description of what this particular case represents is provided, as well as the correct

label for this data interval. These tables do not represent an exhaustive list of all possible data intervals, but are meant to show a sample of the general cases.

The first category is data with one pacemaker discharge present. These data intervals occur with an atrial or ventricular pacemaker in normal and failure events.

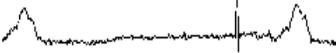
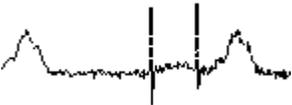
Representation of ECG	Description	Label
<p>R p R</p> 	Single-chamber pacemaker (ventricular)	Normal
<p>R p R</p> 	Single-chamber pacemaker (atrial)	Normal
<p>R p R</p> 	Single-chamber pacemaker (non-capture) followed by spontaneous QRS	Non-capture
<p>Rp R</p> 	Single-chamber with an episode of non-sense followed by spontaneous QRS	Non-sense

Table 4.1: Potential rhythms while Pace Count = 1

The second category of data occurs when two pacemaker discharges are present. Only one example of this category is considered a normal event, with the remainder of possibilities representing various failures that may occur.

Representation of ECG	Description	Label
<p>R p p R</p> 	Dual-chamber pacemaker	Normal

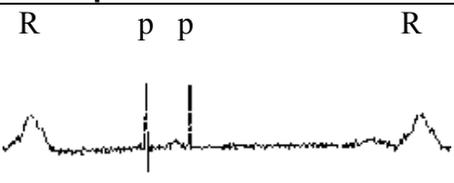
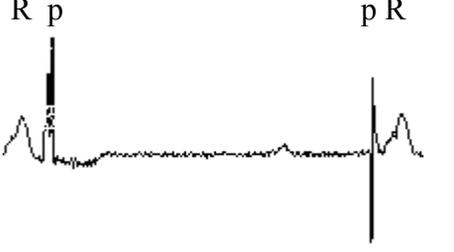
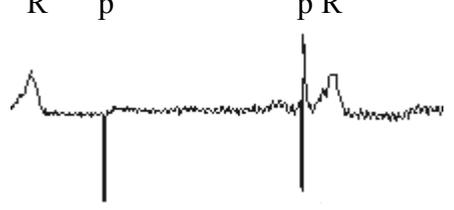
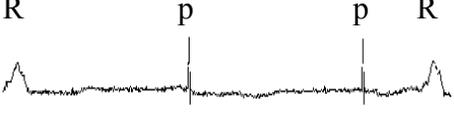
Representation of ECG	Description	Label
	Dual-chamber pacemaker with an episode of non-capture followed by spontaneous QRS	Non-capture
	Single-chamber pacemaker with an episode of non-sense followed by a normal paced beat	Non-sense
	Single-chamber pacemaker with an episode of non-capture followed by a normal paced beat	Non-capture
	Single-chamber pacemaker with two episodes of non-capture followed by a spontaneous QRS	Non-capture

Table 4.2: Potential rhythms while Pace Count = 2

Inspection of the tables above provides some pattern to the failure and normal data intervals. Non-sense failures have pacemaker activity occurring shortly after a QRS complex. Non-capture failures have no QRS complex occurring within the expected time following a pacemaker discharge. The classifiers discussed in sections 4.4 through 4.7 will exploit these patterns to separate non-sense and non-capture failures from normal data intervals.

4.3 Features

The first task for implementing any pattern recognizer is to identify and extract descriptive information, *features*, from the data set. For ECG analysis, the features

frequently involve time interval measurements based upon the ECG complex, including the R-to-R interval, Pacemaker-to-R interval, R-to-Pacemaker interval, Pacemaker-to-Pacemaker interval, and cached previous values of these features [13, 19, 22, 24, 25, 36]. All of these choices exhibit *a priori* knowledge of the intra-cardiac conduction system, and a clinical knowledge of the pacemaker function.

Some of the literature suggests transforming these features using the Karhunen-Loeve Transform [11, 39-41] or other dimension-reducing techniques [37, 42, 43]. While these transforms might improve computational efficiency in some problems, they distort the underlying physiological meanings of the features [13], which may prove helpful when assigning the data to classes. Features chosen for this research are listed in Table 4.3, below.

Feature	Description
Pace Count	Number of pacemaker discharges between two QRS complexes
R-to-R interval	Time between two ventricular contractions of the heart, equal to the patient's pulse divided by 60 seconds.
R-to-Pace	Time between a QRS complex and the next pacemaker discharge
Pace-to-Pace	Time between two pacemaker discharges

Table 4.3: Feature labels and descriptions

These features are chosen because they provide specific information relative to the proper functioning of the electrical and physiological aspects of the cardiac cycle. Each feature represents a particular measurement within the cardiac cycle.

The first feature, Pace Count, provides an uncomplicated identifier for normal data and examples of several non-capture beats in a row. Pace Count = 0 is always assumed normal, because no pacemaker discharge occurred within the selected interval.

A Pace Count > 2 will always be labeled non-capture because no acceptable rhythm has more than two pacemaker discharges within the selected interval, and at least one of the marked discharges has not received a physiological response. Additionally, this feature provides information for segregating the rest of the algorithm: Pace Count = 1 can be treated differently from Pace Count = 2

The second feature, R-to-R interval, provides a measure for the inter-beat period of the heart. An excessively long R-to-R interval may identify a heart not responding properly to a pacemaker discharge, or a lack of stimulation to the heart as illustrated in Figure 4.1.

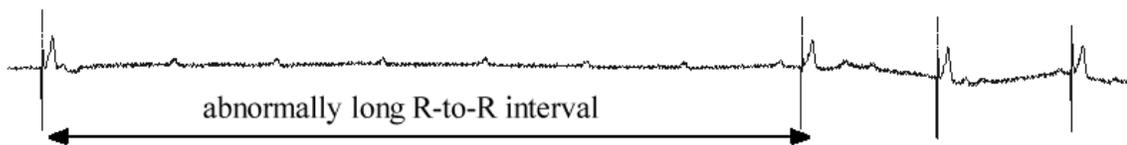


Figure 4.1 Long R-to-R interval due to lack of stimulation

The next feature, R-to-Pace interval, is the time between a QRS complex and the following pacemaker discharge. This measures the period of time the pacemaker allows for repolarization of the myocardium prior to the next discharge. In the event of a non-sense failure, the pacemaker will usually discharge too quickly after the QRS for proper repolarization.

The final feature, Pace-to-Pace interval, is the time between two pacemaker discharges occurring between successive QRS complexes. The Pace-to-Pace interval represents the time between atrial and ventricular discharges in a dual-chamber pacemaker. This interval is similar to the P-wave to QRS complex interval in the heartbeat, and is typically on the order of 0.12-0.20 seconds [8]. Other lengths of this interval represent various failure conditions previously described in Table 4.2.

The R-to-R / Pace-to-Pace Ratio used by this research is a convenient way to characterize the relationship between the R-to-R interval and Pace-to-Pace interval. The ratio is taken by dividing the R-to-R interval by the Pace-to-Pace interval. This ratio provides information on whether the time between pacemaker discharges is appropriate for the length of time between heartbeats, and is useful for both non-capture and non-sense detection.

4.4 Rule-Based Classifier

A desirable classifier implementation for this research is one based upon the relationships between data intervals and the normal and failure data sets [40, 44]. The use of two rules allows the classifier to label the data intervals with fewer calculations. The rules can be implemented easily using if-then statements, eliminate further computation on these data intervals, and reduce the potential for misclassifications.

RULE 1: If pace count = 0, then classify as normal beat. This rule allows for exclusion of spontaneous natural beats that have no pacemaker stimulation, like those in Figure 4.2, below, which are triggered by the SA node. These data intervals cannot have an episode of non-sense or non-capture, and should be identified early to eliminate erroneously labeling these as failures.

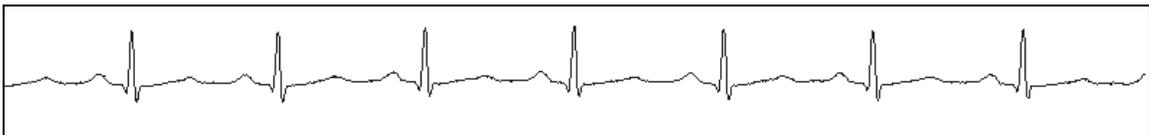


Figure 4.2 Normal Sinus Rhythm, with no paced beats (5 seconds of data)

RULE 2: If pace count > 2, then classify as non-capture failure. This rule detects the case of more than two pacemaker impulses occurring between two QRS

complexes as illustrated in Figure 4.3, below. There is no acceptable heart rhythm that uses more than two pacemaker impulses, therefore at least one of the impulses did not receive a physiological response and is a non-capture failure.

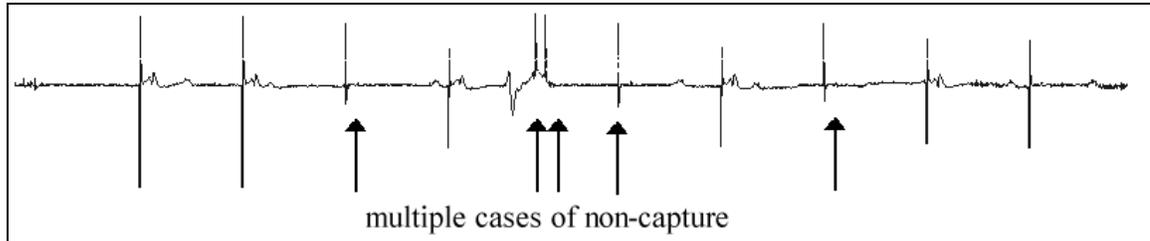


Figure 4.3 Multiple non-capture failures resulting in Pace Count > 2 (5 seconds of data)

4.5 *K-Nearest Neighbors*

Due to the lack of literature published on detection of non-sense and non-capture failures in ECG data, it is necessary to provide a baseline for comparing this research. The baseline chosen is a nearest-neighbor search. The nearest-neighbor search provides a method to identify the closest training points to a test point. Those neighbors then vote upon the label for the test point [11, 40, 45]. A diagram of a 3-nearest neighbor search is provided in Figure 4.4

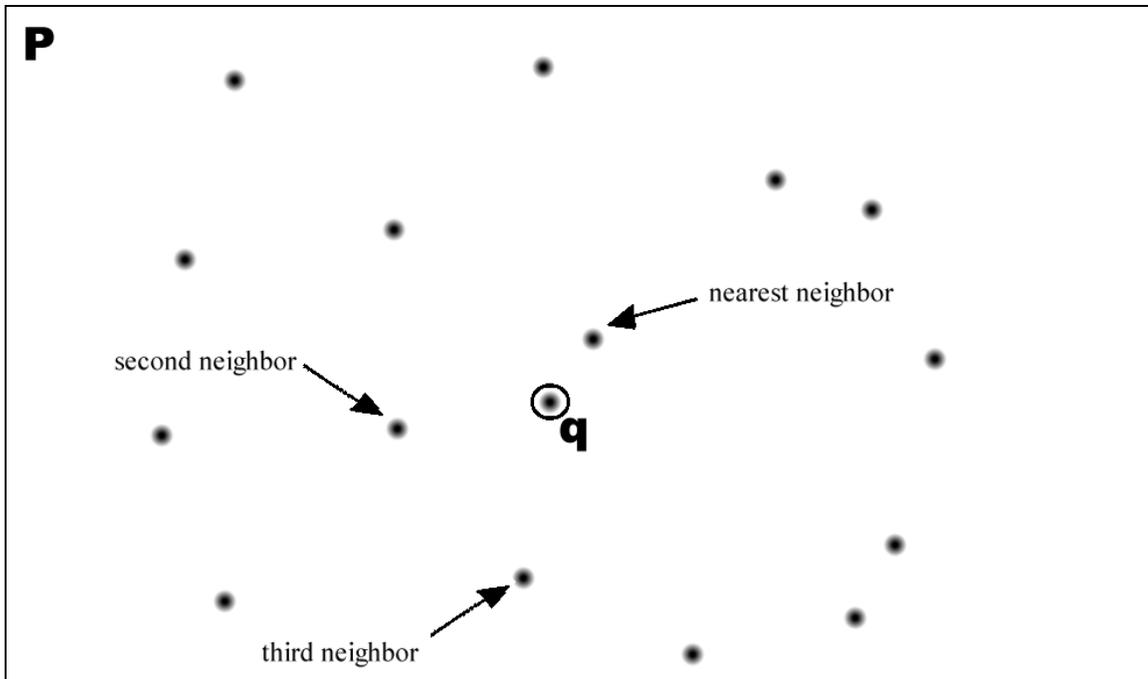


Figure 4.4 3-Nearest Neighbors

The nearest-neighbor search begins by identifying a training set P of data points and a test point q . The nearest neighbor to q is the point within P that has the minimum distance to q . Self-matches are excluded if q is also a member of P . For three nearest neighbors, the three closest points to q within P are selected, excluding self-matches [11, 40]. The distance between neighbors may be calculated with several different methods, this research uses Euclidean distance.

Once the three nearest neighbors have been identified, a vote takes place to determine the label of the test point. Each neighbor's label is counted as one vote, and the majority vote wins. These numbers were chosen because they produce the best results experimentally.

The example provided above in Figure 4.4 shows a two-dimensional nearest-neighbors search. This method can be easily expanded to additional dimensions, and is

used with all features in this research. TStool version 1.11 nearest neighbor search functions (nn_prepare.m and nn_search.m) in Matlab are used in this research [46].

4.6 Threshold-Based Classifier

A second classifier based upon predetermined limits is implemented for comparison to the algorithm developed by this research. The threshold-based classifier is used to illustrate any gain in accuracy resulting from a calculated probability in the proposed algorithm as opposed to a fixed limit in the threshold-based classifier. The structure of the threshold-based classifier is given in Figure 4.5.

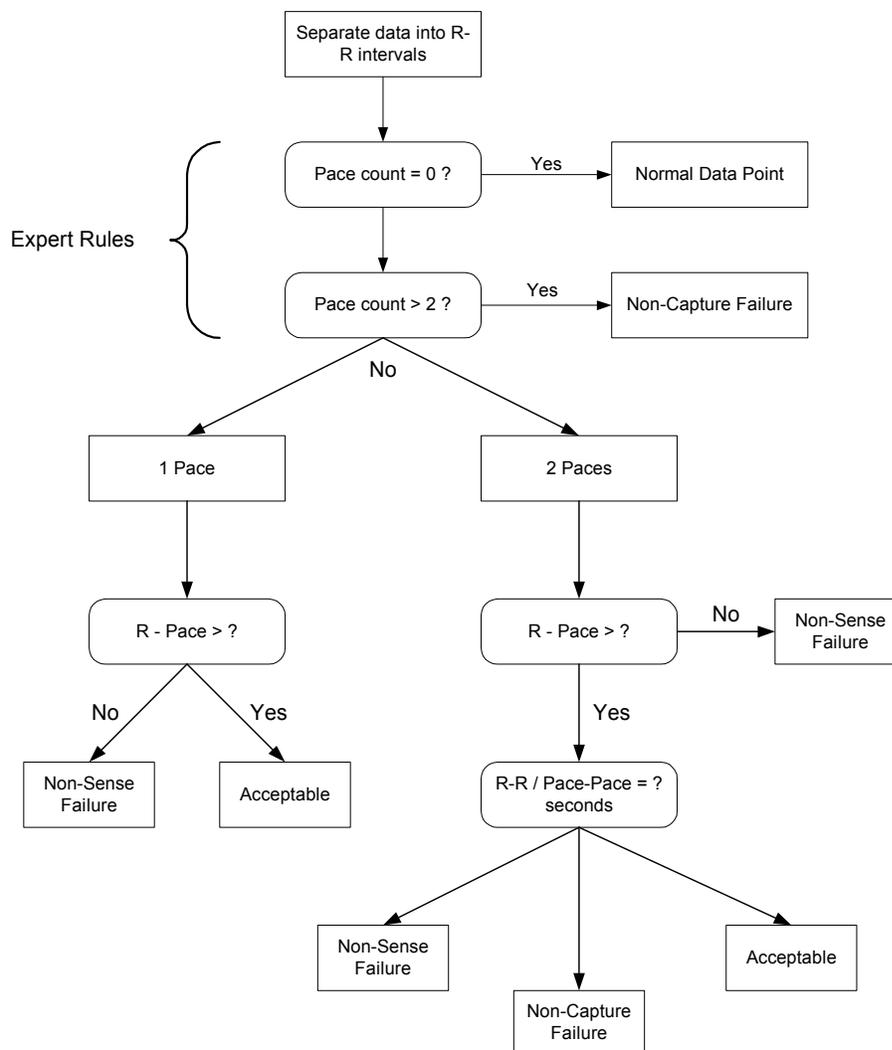


Figure 4.5 Threshold-Based Classifier block diagram

The Threshold-Based classifier begins with the two expert rules discussed in Section 4.4. The Pace Count = 0 and Pace Count > 2 data intervals are immediately classified as normal or non-capture, respectively. Then, the data intervals are sorted into two classifications based upon the Pace Count = 1 or = 2. These data intervals are subjected to threshold tests and classified appropriately.

The simplest implementation of an algorithm in a patient monitor would compare the measured data interval with fixed thresholds programmed into the monitor. If this

implementation proves to have similar results to a statistically based classifier, the benefits gained by fixed thresholds would include less variable storage space and computation time. Justification for the threshold values is discussed with the application in Section 5.3.

4.7 Statistical Pattern Recognition

A Bayesian classifier chooses the most likely class to a given data interval based upon its features. After a model is selected for the statistical distribution of each feature relative to the different classes, posterior probabilities are calculated. The class with the greatest probability is chosen for the data interval. This approach allows us to classify data intervals while minimizing the probability of error [40].

Expert rules already described segregate the data intervals prior to the statistical classifiers. These rules identify two distinct cases for the statistical classifiers: Pace Count = 1 and Pace Count = 2. The structure of the statistical classifier is shown in Figure 4.6.

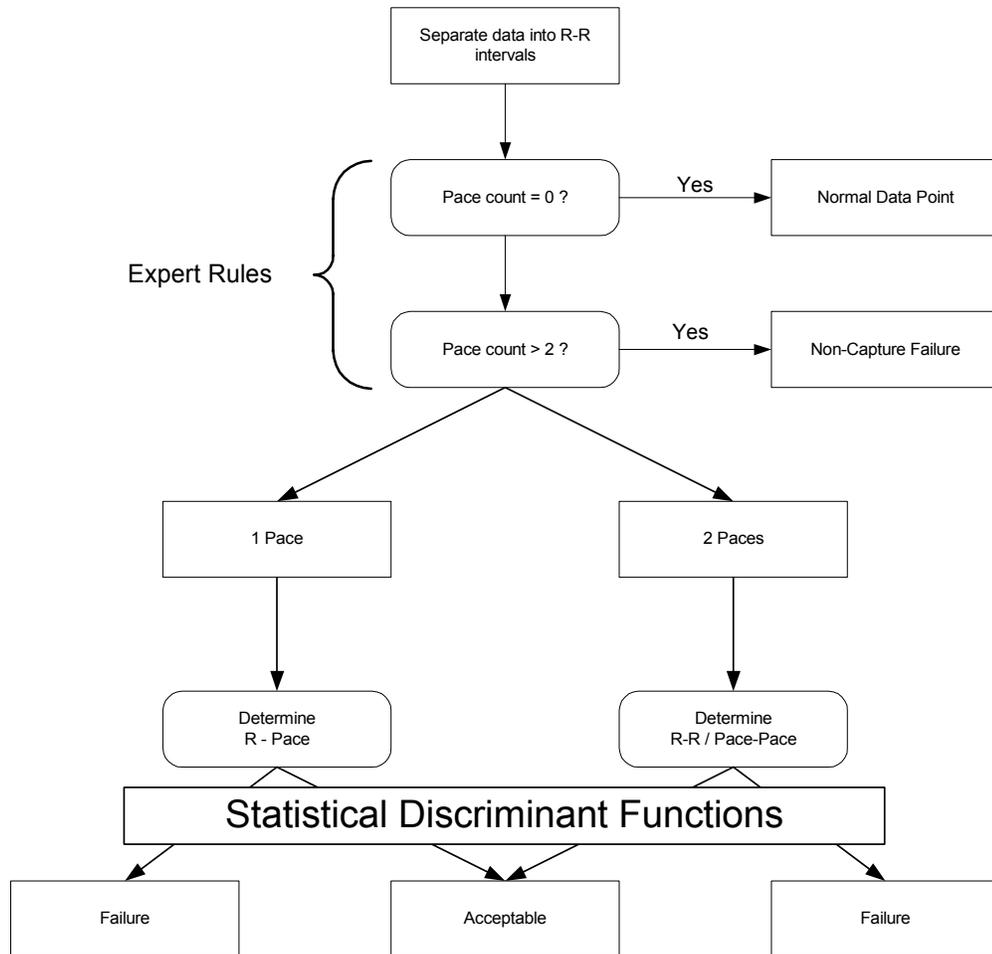


Figure 4.6 Bayesian and Rule-Based Classifier block diagram

Two stages are used in the statistical classifier: the first determines if the point is normal or a failure; the second labels the failure points as non-sense or non-capture. This two-step approach spawns from the primary goal of identifying failures and the secondary goal of distinguishing between failure types. An example of the calculation is shown below to classify data interval j using the R-to-R/Pace-to-Pace ratio:

First, calculate the prior distribution, $P(\omega_i)$, and likelihoods $p(x|\omega_i)$, of class ω_i , given the particular value of the R-to-R/Pace-to-Pace ratio, x_j . Next, compute the posterior distribution by multiplying the prior and likelihood for each class. Finally, use

the decision rule below to determine which of two classes has the greater posterior probability, and is the appropriate label for the data interval[40]:

decide ω_1 if $p(x|\omega_1)P(\omega_1) \geq p(x|\omega_2)P(\omega_2)$
decide ω_2 otherwise

5 Application and Experiments

The experiment setup consisted of a Matlab function that sorted the data into bins for ten-fold cross-validation to measure confidence in the results. Each method was implemented using the same set of bins for the data to produce comparable results.

The results are compared through sensitivity and specificity values. Sensitivity is the probability of a specific failure classification among data intervals with that failure condition actually present. Specificity is the probability of a normal classification among data intervals with no failures actually present.

$$\text{Sensitivity} = \frac{\text{Correctly labeled failures}}{\text{Total number of failures present in data set}}$$

$$\text{Specificity} = \frac{\text{Correctly labeled normal data intervals}}{\text{Total number of normal data intervals present in data set}}$$

Sensitivity and specificity are reported on a scale from 0% to 100%, where 100% is a perfect classifier.

Additionally, computation time is used to compare the degree to which classifiers sacrifice speed for accuracy. Only the Threshold-Based, Rule-Based, and Hybrid classifiers are compared by computation time due to their structural similarities. The k-Nearest Neighbors classifier labels all data intervals simultaneously within the algorithm, and has no easy method to determine the time for one classification.

5.1 Cross-Validation

Cross-validation is used in the following experiments to estimate the error of the classifiers. The set of data intervals is randomly assigned into m disjoint subsets (folds)

of approximately n/m data intervals each, as illustrated in Figure 5.1. This is done in a statistically balanced manner so that each fold has approximately the same number of normal, non-sense, and non-capture data intervals.

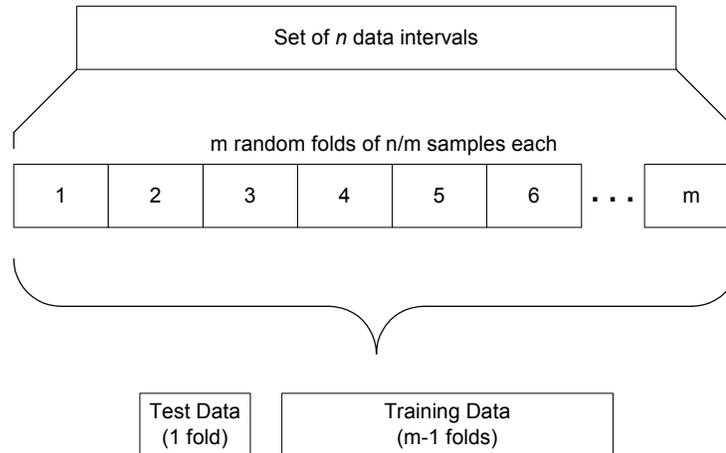


Figure 5.1 m-Fold Cross-Validation

The classifier is applied m times, selecting a different fold as test data for each iteration. The remaining $m-1$ folds are used as the training set [40]. In the following experiments, m was set to ten, yielding a 10-fold cross-validation. Results for each classifier are represented as the mean and standard deviation of results from each iteration of the 10-fold cross-validation.

5.2 *K-Nearest Neighbors*

A 3-nearest neighbor search is performed for comparison to the proposed algorithm. The k-nn classifier is based upon the TSTool program [46], using a Euclidian distance measurement, the default cluster threshold of 64, and all four features (pace count, R-to-R interval, R-to-Pace interval, and Pace-to-Pace interval). A simple majority vote is taken to determine the label for the data interval. A three-neighbor search was chosen experimentally, no significant improvement results were observed with 5 or 7

neighbors. The feature space in Figure 5.2 illustrates separation between the three classes of data intervals, and the outliers that interfere with the classifiers.

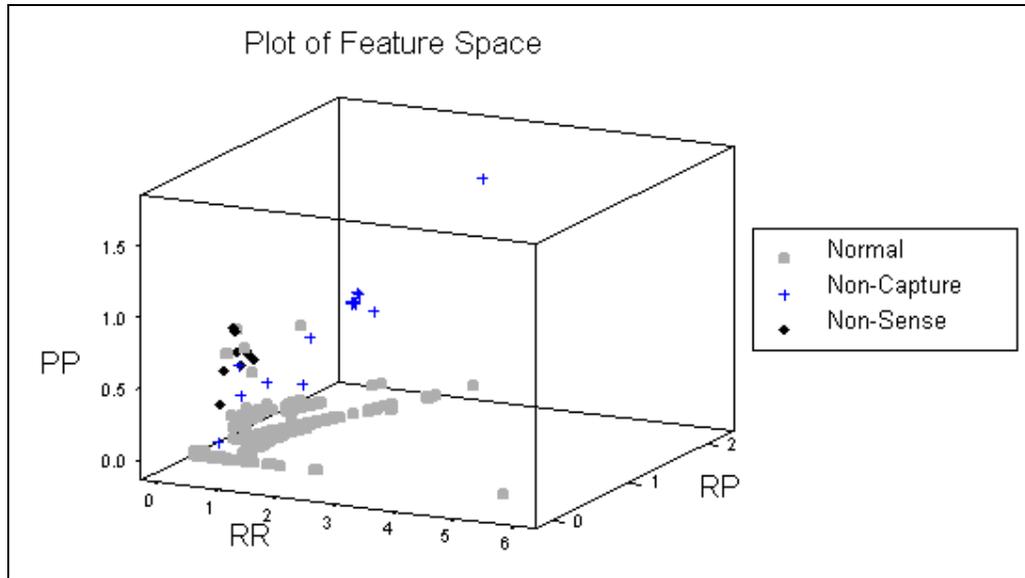


Figure 5.2 Feature Space of All Data Intervals

Results of the 3-nearest neighbor search are given in Table 5.1, below. A time of computation is not valuable in comparison with the other classifiers, because the method of calculation requires searching the entire set of data for neighbors, therefore the time calculation has been omitted.

	Mean	Standard Deviation
% Correct	99.7407	3.237
% False Failure (normal points misclassified as failures)	0.22472	0.14231
% Missed Failure (failure points misclassified as normal)	0.034572	0.072885
% Sensitivity	93.94	0.15077
% Specificity	99.88	0.32203

Table 5.1: 3-Nearest Neighbor Results

5.3 Threshold-Based Classifier

The threshold-based classifier limits are based statistically upon the entire set of data intervals. Labeled data is imported into Minitab [47], and the software draws statistical box plots from the data for analysis. The box plot is drawn with lines extending vertically to the lowest value and highest value in the set of data. The top and bottom edges of the box, and the line within the box represent the 25%, 50% and 75% of the data. Asterisks (*) beyond the box plot have been determined by the program to be statistical outliers of the data. Note that these box plots include all data intervals; the plots are not separated by Pace Count.

The distributions of the R-Pace interval are shown in Figure 5.3. The distributions overlap between normal and failure data, however, a physiological limit can be defined in this case to minimize incorrect labels.

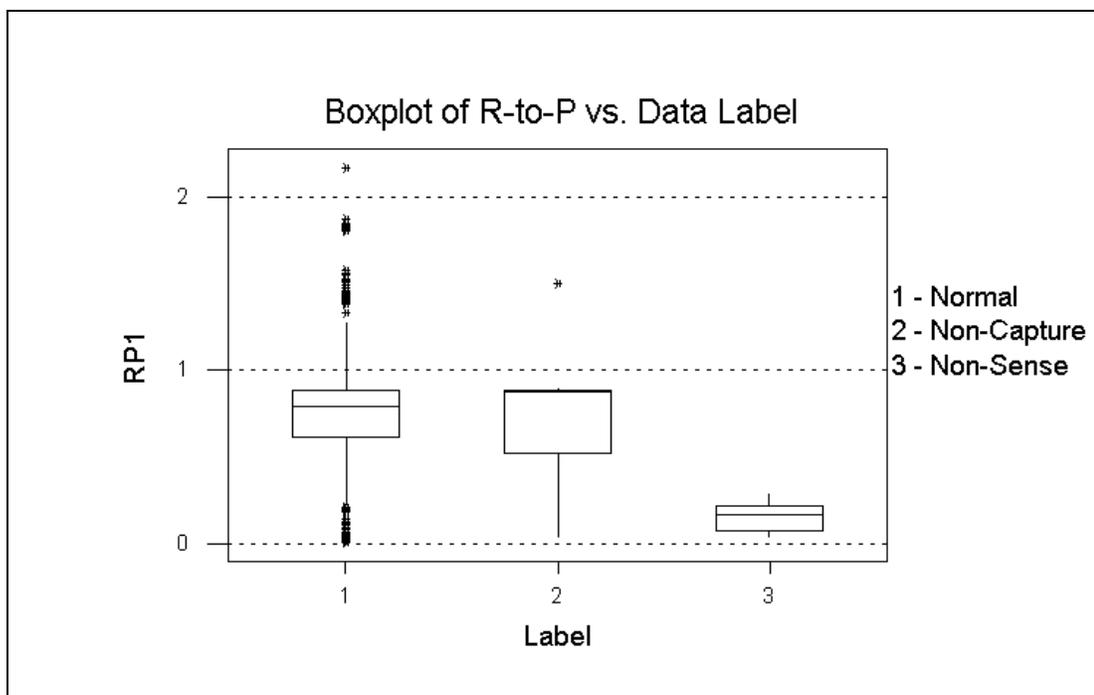


Figure 5.3 Box plot of R-to-Pace interval sorted by data labels

The threshold for R-Pace interval is based upon physiological thresholds for ST-segment length. If a pacemaker were to occur prior to the T wave (the conclusion of the cardiac cycle), it would most certainly be a failure. In general, the ST-segment shortens with increasing heart rate, but the ST-segment is considered to be normal when it is 0.503 seconds or shorter [4]. A hard threshold of 0.503 seconds is implemented for the R-Pace interval, anything less than this is classified as a non-sense failure. The ST-segment length is dependant upon the patient's heart rate, and this threshold is a general rule. A few outliers do fall below this threshold, and will be labeled incorrectly by the classifier. Future investigation should be made for an adaptive R-to-Pace interval limit based upon the R-to-R interval.

The threshold for R-to-R/Pace-to-Pace ratio is based upon statistical distributions from the data set. The distributions were plotted and the limits of 9 for the upper threshold of normal data intervals, 3 for the lower threshold of normal data intervals, and 1.8 for the threshold between the two error classes were chosen. This feature exhibits significant separation between the two failure modes, and a minimal error separation between the failures and normal data, as illustrated in Figure 5.4. Again, a few outliers will be labeled incorrectly.

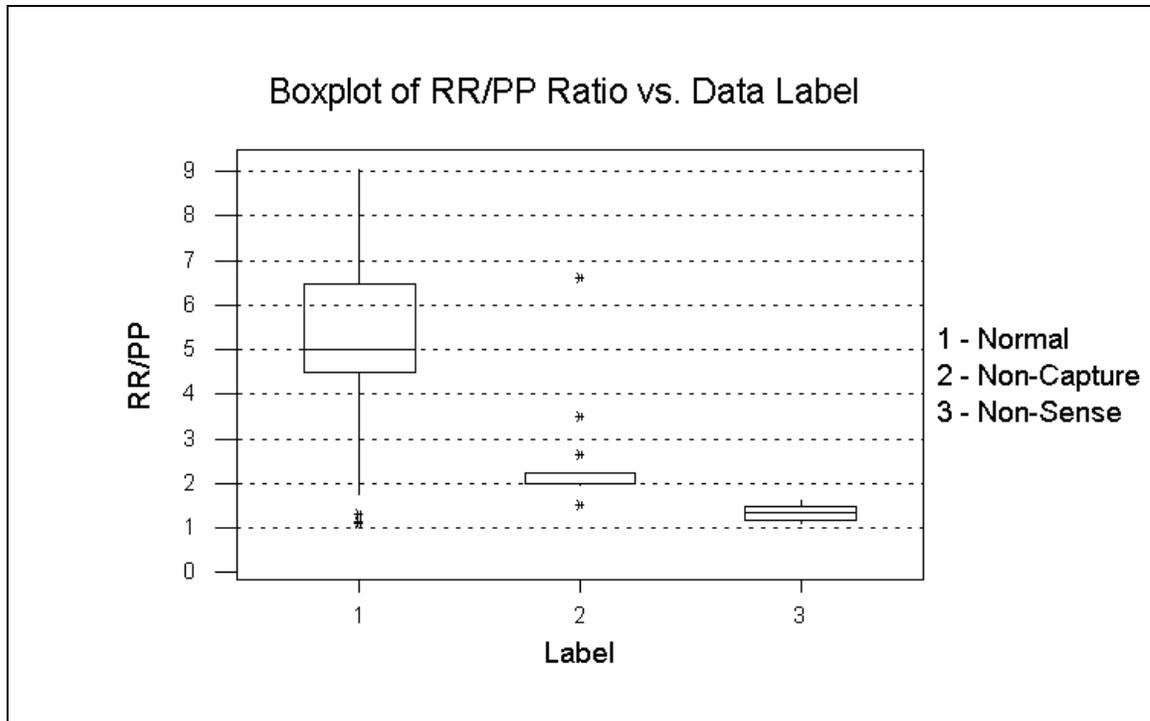


Figure 5.4 Boxplot of R-to-R/Pace-to-Pace ratio sorted by data labels

The block diagram of the Threshold-Based Classifier is shown again, in Figure 5.5, with the appropriate limits included.

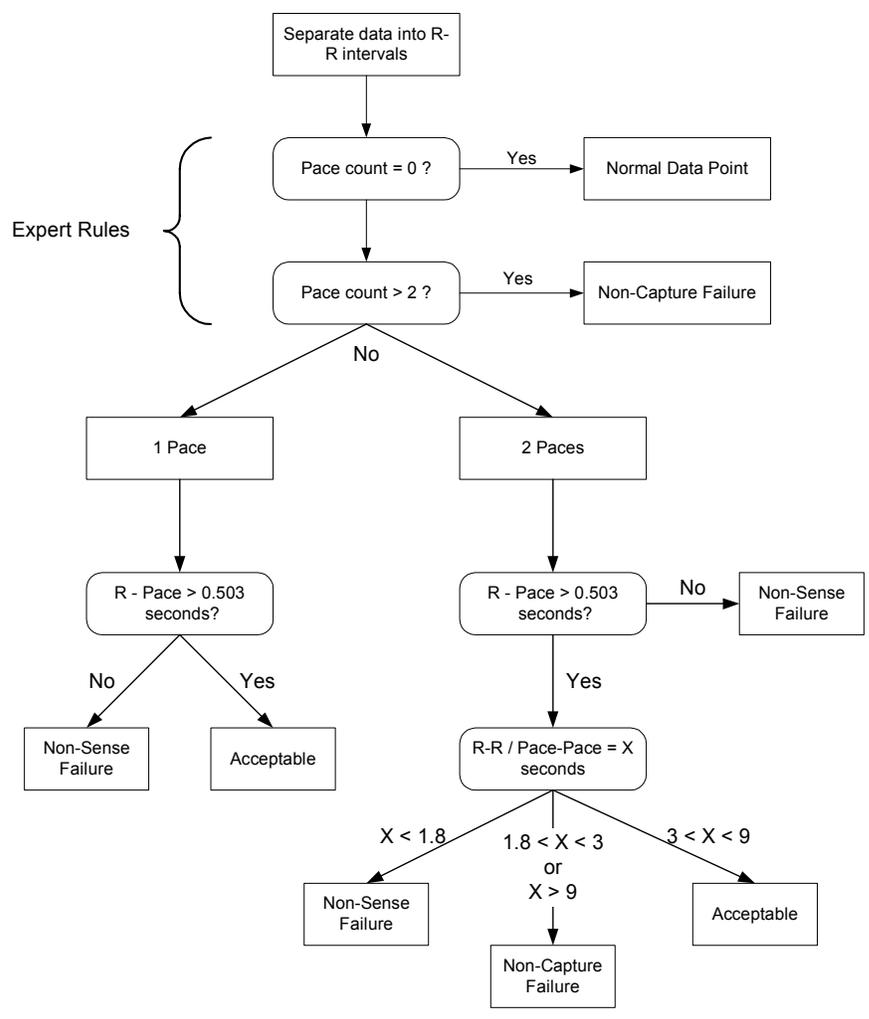


Figure 5.5 Threshold-Based Classifier block diagram

The results of the threshold-based classifier with R-to-Pace threshold of less than 0.503; and RR/Pace-to-Pace ratio thresholds of 9, 3, and 1.8; are presented in Table 5.2, below.

	Mean	Standard Deviation
% Correct	95.5913	2.9096
% False Failure (normal points misclassified as failures)	4.2358	0.85577
% Missed Failure (failure points misclassified as normal)	0.17289	0.0
% Sensitivity	69.70	0.81208
% Specificity	95.74	0.86066
Run Time (ms) (one data interval) (on development system)	0.025903	0.016459

Table 5.2: Threshold-Based Results

The threshold-based classifier is much easier to implement in a patient monitor, but is not as effective as the 3-nearest neighbor search. The accuracy (% Correct) of the classifier dropped from 99.7% of data intervals labeled correctly in the 3-nearest neighbor search to 95.6% in the threshold-based classifier. Sensitivity and specificity also dropped, from 94% to 70% and 99% to 96%, respectively. These results will be compared to the Hybrid classifier discussed in the following section.

5.4 Hybrid Rule-Based and Bayesian Classifier

The final classifier used in this research is a combination of expert rules and statistical pattern recognition. This Hybrid Rule-Based and Bayesian Classifier is the proposed algorithm for non-sense and non-capture classification. This approach allows the identification and rapid classification of easily separable cases while allowing more ambiguous cases to be determined by learned discriminant functions.

Again, the expert rules are implemented. A data interval with Pace Count = 0 is considered a normal data interval as the ECG is spontaneous and not artificially paced

and any data intervals with Pace Count > 2 are immediately identified as non-capture failures. This data interval may represent multiple failures in succession, or a single failure. Then, the data interval is separated into one of two categories: Pace count = 2 or Pace count = 1. Each of these categories has a discriminant function learned from training data.

For the case Pace count = 1, three possible conditions exist: A normal QRS complex initiated by a single-chamber pacemaker (either atrial or ventricular); a spontaneous QRS not sensed by a single-chamber pacemaker followed by another spontaneous QRS; or a spontaneous QRS complex following an episode of non-capture from a single-chamber pacemaker. The R-to-Pace interval is used to separate the non-sense and non-capture failures from the normal data in this category.

For the case Pace count = 2, several possibilities exist. The single normal case is a dual-chamber pacemaker operating properly to trigger a normal paced QRS complex. Failures include a single episode of non-capture by a dual-chamber pacemaker; single episode of non-capture by a single chamber pacemaker followed by a normal paced QRS; two episodes of non-capture by a single chamber pacemaker followed by a spontaneous QRS; and a combination of non-sense and non-capture by a single-chamber pacemaker followed by a normal paced QRS. All of these failures can be identified by an abnormal ratio of R-R interval / Pace-Pace interval.

A two-step approach is implemented to classify the data intervals. Initially, the classifier establishes whether the data interval is normal or a failure. If it is a failure, another classifier determines whether it is non-sense or non-capture. The category with the highest probability is assigned the label for the data interval. A different

implementation of this would assign a cost associated with mislabeling one failure type as the other, and a higher cost associated with mislabeling a failure as normal.

For simplicity, Gaussian models are used for all of the features. Normality tests of the data show generally Gaussian trends, with some slight deviation, which is a path for future investigation. The training data is used to determine the mean, μ_i , and standard deviation, σ_i , of each feature within each class, ω_i ; and the prior distributions for each class $P(\omega_i)$:

$$\text{Mean:} \quad \mu_i = \frac{1}{n} \sum_{k=1}^n x_k$$

$$\text{Standard deviation:} \quad \sigma_i = \sqrt{\frac{1}{n} \sum_{k=1}^n (x_k - \mu_i)^2}$$

$$\text{Prior probability:} \quad P(\omega_i) = \frac{\text{occurrences of class } \omega_i}{\text{total number of training data intervals}}$$

Based upon these measurements, the likelihood and posterior probabilities are calculated:

$$\text{Likelihood:} \quad p(x|\omega_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left[\frac{1}{2}\left(\frac{x-\mu_i}{\sigma_i}\right)^2\right]}$$

$$\text{Posterior probability:} \quad P(\omega_i|x) = P(\omega_i)p(x|\omega_i)$$

The posterior probabilities are compared and the class with the greatest probability is selected for the data interval [40, 48].

5.4.1 Rule-based learners

Results with and without expert rules of Pace = 0, Pace Count > 2 were taken to illustrate the contribution of the rules. Due to the structure of the algorithm, the data still was sorted for Pace = 1, but all other cases were treated with the Pace = 2 algorithm.

	Mean		Standard Deviation	
	Without Rules	With Rules	Without Rules	With Rules
% Correct	83.1634	98.5998	3.3055	3.229
% False Failure (normal points misclassified as failures)	16.6638	1.2273	1.8852	1.2016
% Missed Failure (failure points misclassified as normal)	0.17286	0.17286	0.0	0.0
% Sensitivity	60.61	87.88	0.18496	0.11846
% Specificity	83.29	98.66	0.80508	0.86066
Run Time (ms) (one data interval) (on development system)	0.19471	0.19248	0.042629	0.052483

Table 5.3: Hybrid Classifier Results with and without Rules

The results support the use of rules in the classifier. The greatest improvement is in the accuracy of the classifier. The false failures without rules are approximately 16.7% of all data intervals. This improves to less than 1.5% with the rules applied. The improvement of false failures brings the specificity to 98.7% from 83.3%; while the sensitivity improves to 87.9% from 60.6% due to a reduction of missed failures. A slight time improvement is present with the rules, however this is insignificant when compared to the standard deviation of the time of computation.

Comparing the results of the Threshold-Based classifier with those of the Hybrid classifier supports the learned statistical rules over specific thresholds. The false failures

dropped from 4.2% to 1.2% with the Hybrid classifier, improving the correctly labeled data from 95.6% to 98.6% with the Hybrid classifier. The specificity remained the same, but the sensitivity improved from 0.96 to 0.99 with the Hybrid classifier. These results show the machine-learning algorithm performs superior to the specific thresholds. A comparison between the 3-nearest neighbor search and the statistical classifier with the best performance, the Hybrid classifier, is provided in Section 6.

5.4.2 False Failures

The Hybrid classifier incorrectly labeled a few data intervals as failures. These intervals are actually normal, but due to some abnormality in the patient's ECG, they are misclassified as a failure. This section does not illustrate all false failures that occurred, but provides representative examples.

The first of these false failures is from a patient with a dual-chamber pacemaker that exhibits an abnormally long Pace-to-Pace interval during one beat. There is no immediate cause of this abnormality apparent in the ECG strip, Figure 5.6, below.

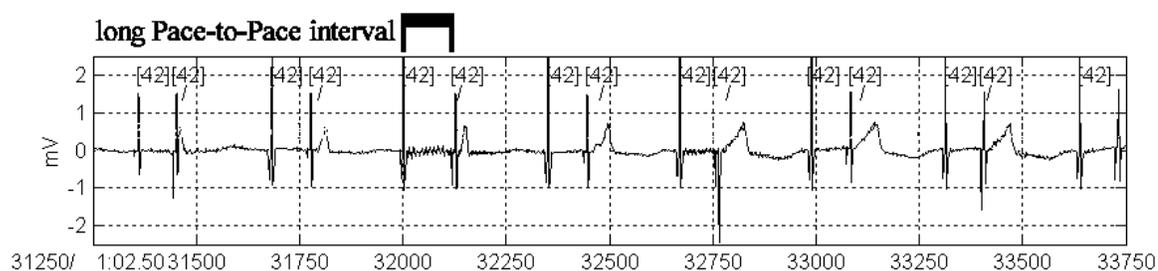


Figure 5.6 False Failure due to abnormally long Pace-to-Pace interval

The statistical classifier interprets this abnormally long Pace-to-Pace interval as a non-capture episode because the first pacemaker discharge appears to lack a physiological response.

The next false failure is an instance where a QRS was labeled earlier in the ECG than it actually occurs. The paced QRS marker (/) appears before the pacemaker discharge marker ([42]) in the annotation files, however the pacemaker discharge occurs prior to the QRS in the ECG strip, shown in Figure 5.7.

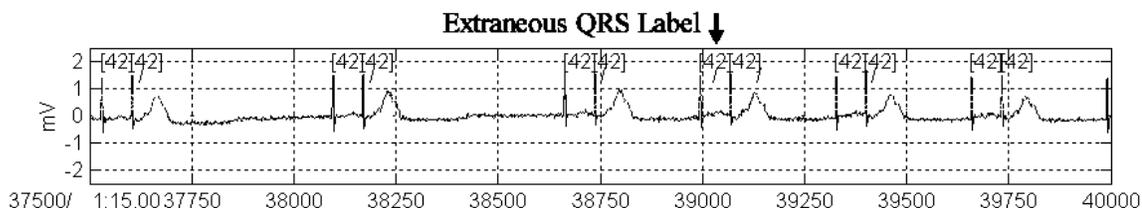


Figure 5.7 False Failure due to early QRS label in data

This results in a non-sense label for the data interval, which is an appropriate label given the error in the annotation file. This classifier is susceptible to similar errors when the timestamp of the pacemaker discharge is delayed beyond the QRS.

A third type of false failure occurs when the patient's heart rate increases to the point where each successive beat becomes misinterpreted by the classifier as a non-sense episode. One example of this is shown in Figure 5.8.

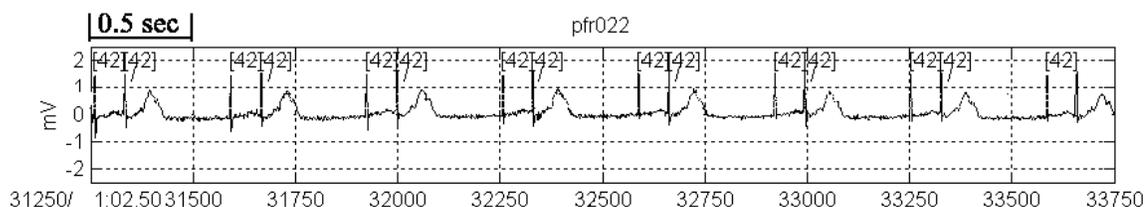


Figure 5.8 False Failure, rapid heart rate classified as non-sense

This patient has a heart rate of 96 beats per minute overall, however some of the beats occur slightly faster, which are labeled non-sense by the statistical classifier. In this patients' ECG, approximately every third beat is misclassified for several seconds during this period of rapid heart rate. An alteration to the classifier that accounts for the present

heart rate may improve these false failure classifications. One example of this alteration would include a manipulation of the R-to-Pace limit based upon the heart rate using a corrected time limit for the patient's Q wave to T wave interval, shown below.

$$QT_{CORRECTED} = \frac{QT_{ESTIMATED}}{\sqrt{(R-to-R\ Interval)}} \quad [8]$$

A final type of false failure is where a mistake occurred in the data labels. Figure 5.9 shows what appears to be a QRS labeled early in the ECG strip, but is actually an episode of non-sense as evident by the malformed QRS complex (appears inverted), following the pacemaker discharge. This point is labeled normal, but the classifier correctly identified it as a non-sense episode.

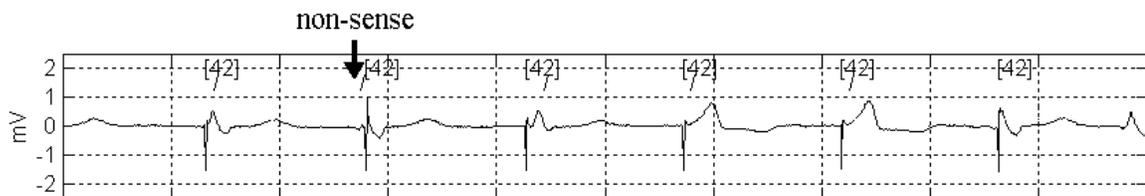


Figure 5.9 False Failure due mislabeled data

5.4.3 Missed Failures

This section discusses two common types of missed failures, explains what caused the failures, and suggests action to modify the classifier to prevent these from occurring.

The first type of missed failure is the case of a single-chamber pacemaker with a non-capture episode followed by a paced beat. In some instances of this situation, the ratio between R-to-R interval and Pace-to-Pace interval is similar to that of a normal beat triggered by a dual-chamber pacemaker, as illustrated in Figure 5.10.

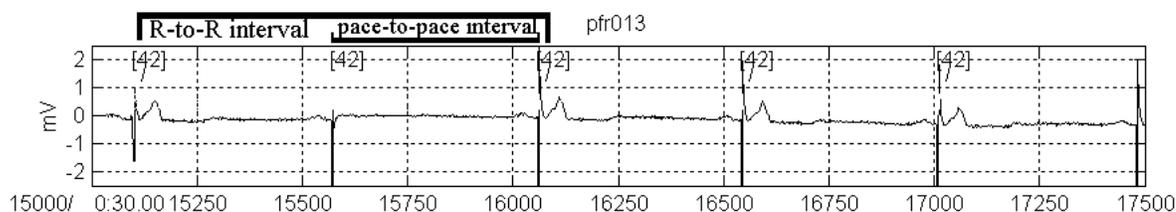


Figure 5.10 Missed Failure, non-capture

This particular failure has a borderline R-to-R/Pace-to-Pace ratio of 1.96. The threshold-based classifier correctly labels this point non-capture, however the statistical classifier does not. This particular case would show an R-to-R interval 1.92 seconds, equivalent to a heart rate of 31.25 beats per minute, and on the edge of the distribution of normal R-to-R intervals.

A second type of missed failure involves the R-to-Pace interval. The interval for the non-sense case in Figure 5.11 is considerably longer than other cases, and may be confused as a non-capture failure by the statistical classifier. If the R-to-R/Pace-to-Pace ratio is similar to the previous case, the point may be labeled normal by the classifier.

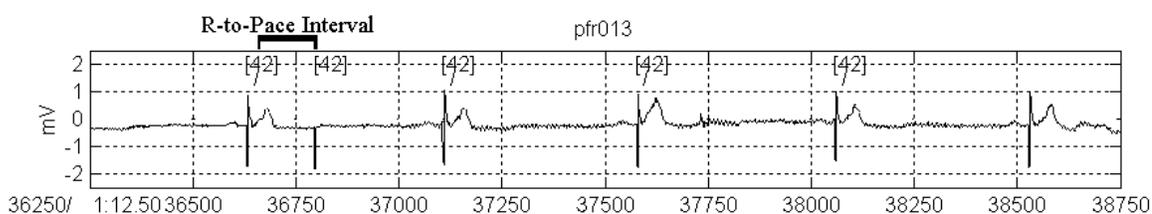


Figure 5.11 Missed Failure, non-sense

Both of these types of missed failures are related to the R-to-R/Pace-to-Pace ratio used as a feature by the statistical classifier. A future solution for this problem may be to separate the R-to-R interval and the Pace-to-Pace interval as individual features for statistical classification in addition to the ratio between the two.

5.4.4 Notes on Implementation

This section briefly discusses the time and computational complexity for the Hybrid Rule-Based and Bayesian Classifier. The memory space required for variable storage and the algorithm itself are given, and the computation time for classification of one data interval on the intended platform is estimated through a benchmarking program.

A widely accepted standard for comparing measurements of computing time for algorithms is *asymptotic notation*. This notation is calculated based upon the structure of an algorithm and is a convenient method to compare multiple algorithms that have vastly different compositions [49, 50].

In the worst-case path through the Hybrid Classifier, the following calculations will be encountered:

Operation	Quantity
Addition, multiplication	36
Exponential Calculations (e^x) x is constant	4
If-Then-Else Operations	8

These operations are encapsulated by a *for-do* loop that runs the length of the fold used for test data. This creates a classifier of complexity $O(n)$ to classify n individual data intervals in the fold. In the implementation, this algorithm will be applied to a single data interval as the interval is measured, setting $n = 1$, for a constant time complexity of $O(1)$ [49].

The spatial complexity of this algorithm is also of order $O(n)$, with $n = 1$ in the planned application. Successive iterations reuse the same variable space, eliminating the

first-order complexity. Considering the strict limitations of the patient monitoring system, it is useful to identify the specific storage space required by the variables, provided in Table 5.4, and the static storage space required for constants, provided in Table 5.5.

Variable	Space needed
Timestamp for QRS (2 needed)	2 x 32-bit word
Timestamp for Pacemaker Discharge (2 needed)	2 x 32-bit word
Pace Count	1 byte
R-to-R interval	32-bit word
Pace-to-Pace interval	32-bit word
R-to-Pace interval	32-bit word
R-to-R / Pace-to-Pace ratio	32-bit word
Posterior probabilities (normal, failure, non-sense, non-capture)	4 x 32-bit word
Data interval class label	1 byte
Total variable storage needed in RAM:	14 x 32-bit words

Table 5.4: Variable storage space required for implementation

Variable	Space needed
Prior probabilities	4 x 32-bit word
First-order statistics for all categories	32 x 32-bit word
Total storage needed in static memory:	36 x 32-bit words

Table 5.5: Constant storage space required for implementation

The program storage requirement for this algorithm is 8.34 kB of space for the uncompiled Matlab code. This is reasonably considered the upper limit of storage requirements, as the application would be compiled into machine code.

A benchmarking algorithm is used to compare the processing capabilities of the development system and the intended patient monitoring system platform. The

Whetstone Benchmark measures the execution speed of fixed- and floating-point instructions; if-then-else statements; sine, cosine, square root and exponential calculations; and assignment statements [51]. This benchmark was chosen for the similarities between the measurements and the computations required by the proposed algorithm. A program written in Visual Basic for Excel using the Whetstone Benchmark compares the PC used for algorithm development to an 80486 DX2 running at 66 MHz [52].

CPU	Whetstone instructions per second	Floating Point operations	Floating Point operations	Floating Point operations	Sin/Cos operations	Exponential, square root operations	Fixed point operations	If-then-else operations	Assignment operations
	MWIPS	MFLOPS 1	MFLOPS 2	MFLOPS 3	COS MOPS	EXP MOPS	FIXPT MOPS	IF MOPS	EQUAL MOPS
80486DX2 66.7 MHz	2.9537749	0.5180816	0.4756098	0.5120944	0.3141133	0.1446065	0.76103	0.3033406	0.2513271
Development System	51.985	10.100	8.960	10.553	7.260	3.499	9.273	4.605	4.034
Performance Increase:	17.59964	19.4943	18.8395	20.6082	23.1139	24.1974	12.185	15.1798	16.0503

Table 5.6: Whetstone Benchmark Comparison

Results in Table 5.6 show the development PC runs between 15 and 25 times faster on the benchmarking computations, with a Whetstone rating 17.6 times faster in MWIPS. This corresponds to the test algorithm running about $(17.6) \times (0.19248 \text{ ms}) = 3.387648 \text{ ms}$ on an 80486 DX2 / 66MHz equivalent system, well below the acceptable limit of one-half second.

The time, space, and processing needs of the proposed algorithm outlined in this section are within the requirements outlined in Section 1.1.2.

6 Conclusions & Future Research

The two best performing classifiers proposed in this research are the 3-Nearest Neighbors Search and the Hybrid Rule-Based and Bayesian Classifier. The greatest difference is in sensitivity, where the nearest-neighbor method shows 94% to the Hybrid sensitivity of 88%. This is due to the increase of missed failures. Slight improvements in specificity and false failure quantity are also seen with the nearest neighbor search. These superiorities do not entirely justify the use of a nearest-neighbor search in the patient monitoring application. Table 6.1 below shows the confusion matrices of the results from each classifier.

3-nn Search		Actual	
		Failure	Normal
Classified	Failure	31	7
	Normal	2	5745

Threshold		Actual	
		Failure	Normal
Classified	Failure	23	245
	Normal	10	5507

Hybrid with rules		Actual	
		Failure	Normal
Classified	Failure	29	77
	Normal	4	5675

Bayesian without rules		Actual	
		Failure	Normal
Classified	Failure	20	961
	Normal	13	4791

Table 6.1 Confusion matrices of results

The main drawbacks with the 3-nearest neighbor search are the amount of data that must be stored by the algorithm and the computation time to search through the data set for the neighbors. Storage for this relatively small set of data intervals takes nearly 380 kB of space, almost 50 times larger than the amount of space required for the entire

application in Matlab. In devices that have only 8 MB of program storage space, this sort of internal database is unreasonable. Additionally, the time complexity to search through a stored population of n data intervals of dimension d , and find k neighbors would be $O(kd \log(n))$, using an optimal approximate nearest neighbors algorithm [53] as compared to the Hybrid Classifier with a time complexity of $O(1)$.

6.1 Future Research

This research shows that while the Hybrid Rule-Based and Bayesian Classifier is useful for detecting non-sense and non-capture, a more accurate, but resource- method of nearest neighbor search exists. Further investigation into the false failures and missed failures has identified some shortcomings of the algorithm and paths for future improvement. Future enhancements to the algorithm will include utilization of the R-to-R interval and Pace-to-Pace intervals separately as well as the ratio between the two; investigation and correction of mislabeled data; additional ECG recordings that remain unlabeled at this point; and implementation techniques that have been presented by other research.

Recent advancements in pacemaker technology include *biventricular* and *dual atrial* pacemakers, with electrodes implanted into both ventricles and the both atria. These pacemakers independently stimulate all four chambers, causing the potential of two atrial and two ventricular pacemaker discharges. Depending upon delays to surface electrodes and the programming of the pacemaker, these pacemakers may display three or four discharges on the ECG while operating normally [54-56]. This algorithm must be adapted to appropriately diagnose these newer pacemakers and accommodate changes in annotation systems designed to identify four-chamber pacemakers.

References

- [1] K. A. Ellenbogen, *Cardiac Pacing*. Cambridge, MA: Blackwell Scientific Publishing, 1992.
- [2] P. Lucas, A. Tholen, and G. van Oort, "An intelligent system for pacemaker reprogramming," *Artificial Intelligence in Medicine*, vol. 17, pp. 249-269, 1999.
- [3] H. J. L. Marriott, *Practical Electrocardiography*, 7th ed. Baltimore, MD: Williams & Wilkins, 1983.
- [4] S. S. Barold, *Modern Cardiac Pacing*. Mount Kisco, New York: Futura Publishing Company, 1985.
- [5] "AAMI EC13," in *Cardiac monitors, heart rate meters, and alarms*. Arlington, VA: Association for the Advancement of Medical Instrumentation, 2002.
- [6] "Basic ECG Self-Study Program," Marquette Electronics, Inc., Milwaukee, Wisconsin 1992.
- [7] W. Einthoven, "The Different Forms of the Human Electrocardiogram and their Signification," *Lancet*, vol. 1, pp. 853-861, 1912.
- [8] F. G. Yanowitz, "ECG Learning Center," 2001.
- [9] D. Lemkin and G. Plotnick, "Introduction to ECGs."
- [10] D. Wagner, "ECG Chest Leads." Milwaukee: GE Medical Systems - Information Technologies, 2003.
- [11] H. Kantz and T. Schreiber, *Nonlinear time series analysis*. Cambridge: Cambridge University Press, 1997.
- [12] H. A. Guvenir, "Detection of Abnormal ECG Recordings using Feature Intervals," *Proceedings of the Tenth Turkish Symposium on Artificial Intelligence and Neural Networks*, pp. 265-274, 2001.
- [13] M. Akay, E. J. Ciaccio, and S. M. Dunn, "Biosignal Pattern Recognition and Interpretation Systems: Part 4 of 4: Review of Applications," *IEEE Engineering in Medicine and Biology Magazine*, vol. 13, pp. 269-273, 1994.
- [14] H. A. Guvenir, B. Acar, G. Demiroz, and A. Cekin, "A supervised machine learning algorithm for arrhythmia analysis," *Computers in Cardiology*, vol. 24, pp. 433-436, 1997.
- [15] M. Hoher, D. E. Ritscher, S. Bauer, M. Kodler, H. A. Kestler, and V. Hombach, "Beat-to-Beat Variability of QRS Duration," *Computers in Cardiology*, vol. 24, pp. 613-616, 1997.
- [16] B.-U. Köhler, C. Hennig, and R. Orglmeister, "The principles of software QRS detection: Reviewing and comparing algorithms for detecting this important ECG waveform," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, pp. 42-57, 2002.
- [17] E. Tatara and A. Cinar, "Interpreting ECG data by integrating statistical and artificial intelligence tools: Combining knowledge-based systems with statistical methods for more robust intelligent patient monitoring," *IEEE Engineering in Medicine and Biology Magazine*, vol. 21, pp. 36-41, 2002.
- [18] F. M. Roberts, R. J. Povinelli, and K. M. Ropella, "Identification of ECG Arrhythmias using Phase Space Reconstruction," presented at Principles and

- Practice of Knowledge Discovery in Databases (PKDD'01), Freiburg, Germany, 2001.
- [19] M. E. Cohen and D. L. Hudson, "New Chaotic Methods for Biomedical Signal Analysis," *IEEE EMBS Information Technology Applications in Biomedicine*, pp. 117-122, 2000.
 - [20] D. T. Kaplan and R. J. Cohen, "Is Fibrillation Chaos," *Circulation Research*, vol. 67, pp. 886-892, 1990.
 - [21] D. M. Tumey, P. E. Morton, D. F. Ingle, C. W. Downey, and J. H. Schnurer, "Neural Network Classification of EEG using Chaotic Preprocessing and Phase Space Reconstruction," presented at 1991 IEEE Seventeenth Annual Northeast Bioengineering Conference, 1991.
 - [22] V. R. Zurro, A. L. Stelle, and J. Nadal, "Detection of Atrial Persistent Rhythm Based on P-Wave Recognition and RR Interval Variability," *Computers in Cardiology*, pp. 185-188, 1995.
 - [23] N. Parthasarathy, M. Ahmadi, and M. Shridhar, "Pattern Based ECG Diagnosis," *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pp. 585-588, 1991.
 - [24] J. Bai and J. Lin, "A Pacemaker Working Status Telemonitoring Algorithm," *IEEE Trans. on Information Technology in Biomedicine*, vol. 3, pp. 197-204, 1999.
 - [25] S. Ghiringhelli, E. Petrucci, A. Vigano, L. Mainardi, M. Bertinelli, and R. Canziani, "Assessment of Pacemaker Function in Ambulatory ECG Recordings Using Real Time Analysis," *IEEE Computers in Cardiology*, pp. 37-40, 1995.
 - [26] V. E. Splett, "Method and apparatus for capture detection," in *United States Patent*. United States of America: Medtronic, Inc., 2002.
 - [27] M. Gryzwa and Q. Zhu, "Rate management during automatic capture verification," in *United States Patent*. United States of America: Cardiac Pacemakers, Inc., 2002.
 - [28] M. T. Hemming, B. C. Peck, B. A. Blow, and S. M. Morrison, "Microprocessor capture detection circuit and method," in *United States Patent*. United States of America: Medtronic, Inc., 1999.
 - [29] S. A. Marinello, "Method and apparatus for accurate counting of paced heartbeats," in *U.S. Patent*. United States of America: Siemens Medical Systems, Inc., 1998.
 - [30] S.-C. Pei and C.-C. Tseng, "Elimination of AC interference in electrocardiogram using IIR notch filter with transient suppression," *IEEE Trans. on Biomedical Engineering*, vol. 42, pp. 1128-1132, 1995.
 - [31] J. C. Huhta and J. G. Webster, "60-Hz interference in electrocardiogram," *IEEE Trans. on Biomedical Engineering*, vol. 20, pp. 91-101, 1973.
 - [32] W. Dai, Z. Yang, S. L. Lim, O. Mikhailova, and J. Chee, "Processing and analysis of ECG signal using nonorthogonal wavelet transform," *Proceedings of the Annual Conference on Engineering in Medicine and Biology*, vol. 20, pp. 139-142, 1998.
 - [33] H. Jadvar, J. M. Jenkins, R. E. Stewart, M. Schwaiger, and R. C. Arzbaecher, "Computer Analysis of the Electrocardiogram During Esophageal Pacing Cardiac Stress," *IEEE Trans. on Biomedical Engineering*, vol. 38, pp. 1089-1099, 1991.

- [34] M. Yelderman, B. Widrow, J. Cioffi, E. Hesler, and J. Leddy, "ECG enhancement by adaptive cancellation of electrosurgical interference," *IEEE Trans. on Biomedical Engineering*, pp. 392-398, 1983.
- [35] S. R. Quint, J. A. Messenheimer, M. B. Tennison, and H. T. Nagle, "Assessing Autonomic Activity from the EKG Related to Seizure Onset Detection and Localization," presented at Proc. of Second Annual IEEE Symposium on Computer Based Medical Systems, 1989.
- [36] M. Akay, E. J. Ciaccio, and S. M. Dunn, "Biosignal Pattern Recognition and Interpretation Systems. 2. Methods for Feature Extraction and Selection," *IEEE Engineering in Medicine and Biology Magazine*, vol. 12, pp. 106-113, 1993.
- [37] M. Akay, E. J. Ciaccio, and S. M. Dunn, "Biosignal Pattern Recognition and Interpretation Systems: 3. Methods of Classification," *IEEE Engineering in Medicine and Biology Magazine*, vol. 13, pp. 129-135, 1994.
- [38] G. B. Moody, "MIT-BIH Database," Harvard-MIT Division of Health Sciences and Technology, 2002.
- [39] H. D. I. Abarbanel, *Analysis of observed chaotic data*. New York: Springer, 1996.
- [40] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York, New York: John Wiley & Sons, Inc., 2001.
- [41] M. Stridh, L. Sornmo, C. J. Meurling, and S. B. Olsson, "Time-frequency characterization of atrial arrhythmias using principal decomposition," *Proceedings of the Annual Conference on Engineering in Medicine and Biology*, vol. 2, pp. 1808-1811, 2001.
- [42] L. Vladutu, S. Papadimithou, S. Mavroudi, and A. Bezerianos, "Ischemia detection using supervised learning for hierarchical neural networks based on kohonen-maps," *Proceedings of the Annual Conference on Engineering in Medicine and Biology*, vol. 2, pp. 1688-1691, 2001.
- [43] Y. H. Nam, Y. Y. Kim, H. T. Kim, H. D. Ko, and K. S. Park, "Automatic detection of nausea using bio-signals during immersion in a virtual reality environment," *Proceedings of the Annual Conference on Engineering in Medicine and Biology*, vol. 2, pp. 2013-2015, 2001.
- [44] M. Akay, E. J. Ciaccio, and S. M. Dunn, "Biosignal Pattern Recognition and Interpretation Systems," *IEEE Engineering in Medicine and Biology Magazine*, vol. 12, pp. 89-95, 1993.
- [45] J. L. Bentley, B. W. Weide, and A. C. Yao, "Optimal Expected-Time Algorithms for Closest Point Problems," *ACM Transactions on Mathematical Software*, vol. 6, pp. 563-580, 1980.
- [46] C. Merkwirth, U. Parlitz, I. Wedekind, and W. Lauterborn, "TS Tools," 2001.
- [47] "Minitab 12.2 for Windows," 1998.
- [48] A. Papoulis, *Probability, random variables, and stochastic processes*, 3rd ed. New York: McGraw-Hill, 1991.
- [49] G. Brassard and P. Bratley, *Fundamentals of Algorithmics*. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1996.
- [50] U. Manber, *Introduction to Algorithms: A Creative Approach*. Reading, MA: Addison-Wesley, 1989.
- [51] H. J. Curnow and B. A. Wichmann, "A Synthetic Benchmark," *Computer Journal*, vol. 19, pp. 43-47, 1976.

- [52] R. Longbottom, "Visual Basic Whetstone Benchmark Via Excel," vol. 2003, 1997.
- [53] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM*, vol. 45, pp. 891-923, 1998.
- [54] A. Bakels, R. Leinders, and C. de Roos, "Four-chamber pacing system for optimizing cardiac output and determining heart condition," in *U.S. Patent*. United States of America: Medtronic, Inc., 2001.
- [55] A. P. Kramer, J. E. Stahmann, R. H. Wentkowski, K. L. Baker, J. W. Hartley, and D. B. Krig, "System providing ventricular pacing and biventricular coordination," in *U.S. Patent*. United States of America: Cardiac Pacemakers, Inc., 2001.
- [56] M. Brignole, "A novel use of cardiac pacing to improve cardiac function, quality of life and (hopefully) survival in patients with heart failure and permanent atrial fibrillation," *European Heart Journal*, vol. 23, pp. 1732-1736, 2002.