2013

ASSESSMENT AND PREDICTION OF CARDIOVASCULAR STATUS DURING CARDIAC ARREST THROUGH MACHINE LEARNING AND DYNAMICAL TIME-SERIES ANALYSIS

Sharad Shandilya
Virginia Commonwealth University

ASSESSMENT AND PREDICTION OF CARDIOVASCULAR STATUS DURING CARDIAC ARREST THROUGH MACHINE LEARNING AND DYNAMICAL TIME-SERIES ANALYSIS

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

by

Sharad Shandilya
M.S., Department of Computer Science (Virginia Comm. University, Virginia), 2010
B.S., Department of Biomedical Engineering (Virginia Comm. University, Virginia), 2006

Major Director: Kayvan Najarian
Associate Professor, Department of Computer Science

Virginia Commonwealth University
Richmond, Virginia
July, 2013

Dedicated to my guru

and my family

Table of Contents

Abstract

ASSESSMENT AND PREDICTION OF CARDIOVASCULAR STATUS DURING CARDIAC ARREST THROUGH MACHINE LEARNING AND DYNAMICAL TIME-SERIES ANALYSIS

by Sharad Shandilya

A thesis submitted in partial fulfillment of the requirements for the
degree of Doctor of Philosophy at Virginia Commonwealth University

Virginia Commonwealth University, 2013
Major Director: Kayvan Najarian, Associate Professor, Department of Computer Science

In this work, new methods of feature extraction, feature selection, stochastic data characterization/modeling, variance reduction and measures for parametric discrimination are proposed. These methods have implications for data mining, machine learning, and information theory.

A novel decision-support system is developed in order to guide intervention during cardiac arrest. The models are built upon knowledge extracted with signal-processing, non-linear dynamic and machine-learning methods. The proposed ECG characterization, combined with information extracted from PetCO2 signals, shows viability for decision-support in clinical settings. The approach, which focuses on integration of multiple features through machine learning techniques, suits well to inclusion of multiple physiologic signals.

Ventricular Fibrillation (VF) is a common presenting dysrhythmia in the setting of cardiac arrest whose main treatment is defibrillation through direct current countershock to achieve return of spontaneous circulation. However, often defibrillation is unsuccessful and may

even lead to the transition of VF to more nefarious rhythms such as asystole or pulseless electrical activity. Multiple methods have been proposed for predicting defibrillation success based on examination of the VF waveform. To date, however, no analytical technique has been widely accepted. For a given desired sensitivity, the proposed model provides a significantly higher accuracy and specificity as compared to the state-of-the-art. Notably, within the range of 80-90% of sensitivity, the method provides about 40% higher specificity. This means that when trained to have the same level of sensitivity, the model will yield far fewer false positives (unnecessary shocks).

Also introduced is a new model that predicts recurrence of arrest after a successful countershock is delivered. To date, no other work has sought to build such a model. I validate the method by reporting multiple performance metrics calculated on (blind) test sets.

1. Introduction

In this dissertation, I deal with some basic problems in predictive model development through machine learning and characterization of dynamical data. The problems are studied within the context of an applied project in emergency medicine. As such, all solutions are applied and tested in order to improve an impact-oriented decision-support system.

Specifically, the problems span the following areas: feature ranking, feature subset selection, parameter selection, learning with imbalanced classes, adaptive filtering, and modeling a dynamical stochastic system. The first four problems pose the bias-variance dilemma, which serves as a central theme.

The work presented here draws on ideas from three broad fields: Machine Learning, Non-Linear Dynamics, and Signal Processing. The reader is assumed to have a basic knowledge of concepts in these fields.

1.1 The Need for a Decision-Support System

Sudden cardiac death is a significant public health concern and a leading cause of death in many parts of the world [Lloyd-Jones 2010]. In the United States, cardiac arrest claims greater than 300,000 lives annually. Survival rates for out-of-hospital cardiac arrest remain dismal [Nichol 2008]

Otherwise robust and able to withstand many variations in physiologic state, once in fibrillation, the heart cannot spontaneously convert to a regular circulating rhythm with coordinated depolarization and repolarization. Ventricular Fibrillation (VF) is the initially encountered arrhythmia in 20-30% of cardiac arrest cases [Nadkarni 2006]. VF waveform is contributed by multiple reentrant circuits causing its pathophysiology to be extremely dynamic. Coronary artery perfusion provided by cardio-pulmonary resuscitation (CPR) prior to defibrillation has been shown to improve chances for Return of Spontaneous Circulation (ROSC) [Valenzuela 1997]. Repetitive unsuccessful shocks cause thermal injury to cardiac tissue, which deteriorates heart function upon survival [Strohmenger 2008], along with adding to the time lost. A victim's chances of survival worsen by 10% for every minute of VF that remains untreated [Valenzuela 1997].

Defibrillation is a procedure that delivers an electrical current that depolarizes a critical mass of the myocardium simultaneously. Defibrillation increases the possibility of the sino-atrial node regaining control of the rhythm. Coronary artery perfusion provided by cardio-pulmonary resuscitation (CPR) prior to defibrillation has been shown to improve chances for ROSC [Valenzuela 1997]. As victims enter the CPR phase of cardiac arrest, predicting defibrillation success may become paramount in preventing unnecessary interruptions to CPR [Weisfeldt 2002]. Repetitive unsuccessful shocks can reduce chest compression time and can cause injury to cardiac tissue, impacting heart function upon survival. Even worse, unsuccessful shocks can cause VF to deteriorate into asystole or pulseless electrical activity (PEA), which are more difficult to resuscitate [Strohmenger 2008].

Hence, increasing efficacy of countershocks is of principal importance. To achieve this, I develop an integrative decision-support model that guides the interventionist by learning from real-time information gained from the patient.

1.2 Contributions to Computer Science

I propose novel methods in the following sub-fields of computer science:

- Feature-Selection (section 7.3): The Wrapper-Based Method, as proposed, focuses on reduction in *variance*, while preserving minority class information. The method also focuses on selecting orthogonal/non-redundant features.
- Parameter-Tuning/Regularization (section 8.2): Two methods are proposed for boosting performance on non-homogenous datasets when varying multiple parameters: 1. a Wrapper-Based Method that searches for the 'best' model across training data and 2. the High-Platform Method that intentionally induces sub-optimality to exhaustive search in order to add *Bias.*
- Feature Extraction (sections 9.1.2, 9.1.3, 6.3): Second order features are extracted through an auto-recursive and a time-lapse (delta-state) method.
- Non-Linear Dynamics (section 6.3): A new method called 'QPD-PD' that is geared for classification is introduced.
- Information Theory (section 6.3): The measure *sKD* to compare discrete distributions is introduced. The measure *sep* is a non-traditional metric that is tested and validated in light of standard hypothesis testing. The 'Pole-Count' feature is introduced.

All of the methods proposed would find application across a broad spectrum of fields that deal with or utilize data. Especially, fields that exist within non-deterministic, non-stationary domains would benefit from the non-linear dynamical and feature extraction methods proposed. Other methods proposed have direct application in predictive/decision-support modeling. A few examples are financial services (behavioral modeling), financial markets, real-time bidding (internet traffic), gene selection, biomedical time-series analysis, forecasting, medical and non-medical goal-directed decision support systems, etc. Specifically, the prediction task may output the probability of stock-options being exercised in the future, or the value of the website visit with respect to a certain ad, or identify high-risk genes that are influence an individual's susceptibility to a disease, or short/long term patient outcomes based on current condition.

2. Background

2.1 Physiologic Background and Related Work

The ‘QRS’ complex within an ECG signal represents ventricular depolarization (the contraction that forces blood through the arteries and into the tissues), with Q and S representing minima while R representing a maximum in the ECG waveform. Lack of a clear QRS complex renders traditional methods of ECG analysis, which use physiologic correlates of the detected P, R and T waves, ineffective. The effect of acute ischemia on tissue excitability induces conversion of VF from type-1 coarse VF to type-2 smooth VF [Zaitsev 2000]. Type 1 VF has been correlated with the multiple-wavelet theory, while type 2 has been shown to be driven by a mother rotor [Weiss 2005]. This conversion *partially* conforms to rapidly attenuating chances of survival with increasing VF duration [Eilevstjonn 2007], and can be quantified by any measure that can account for both, a decrease in amplitude and a shift in spectral composition of the signal (such as the Fourier Transform).

Gundersen and colleagues [Gundersen 2008] have shown that predictive features of the VF waveform suffer from random effects (p-values less than 10^{-3}). This was proved with a mixed-effects logistic regression model. Random effect-sizes, calculated as standard deviation of the ‘random’ term in the model, varied from 73% to 189% of the feature effect-sizes. Thus an additional objective of our work aims at countering the variance due to such effects. I hypothesized that other physiologic signals obtained during CPR, such as end-tidal carbon dioxide (PetCO2), can help build a more ‘complete’ model. PetCO2 monitoring allows for the measurement of exhaled carbon dioxide from a patient. The level of exhaled carbon dioxide has been positively correlated with the amount of blood flow produced by chest compressions during CPR (see *Discussion*).

2.2 Time-Series Methods

A time-series variable can be called a signal when the values that the variable takes are continuous. The sampling rate determines the granularity (discrete nature) of the measured values. Fourier Transform (FT) is a widely used technique for decomposing time-series into sinusoids. FT assumes that time-series are composed of sinusoids. FT also assumes that these sinusoids, when combined *linearly*, yield the real signal. Fast Fourier Transform [Cooley 1965] is a widely known algorithm that efficiently implements the Discrete Fourier Transform (DFT).

The wavelet transform [Akansu 1992] is another widely known and powerful technique for decomposing a signal into the scaled and dilated versions of a mother wavelet. While wavelet decomposition has proven to be more effective, clinical transition of such approaches has been precluded due to low specificities.

Other methods of time-series analyses involve calculating characteristics of the signal directly from the given time-series. These can include measures such as the median-slope [Joar 2007] or higher order measures such as ‘pole-count’ that I define in section 6.3.

2.3 Machine Learning

A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E* [Mitchell 1997]. For our purpose, the task is classification and performance is measured by a weighted misclassification rate. More specifically, for the applied *predictive* model, I classify a time-series segment into one of the possible future states.

To date, numerous machine learning techniques have been developed for inductive learning. Broad categories include rule-based algorithms, decision-trees and functions. The objective is to solve the *inverse problem*. The inverse problem refers to generalizing a concept or a model from given data or observations. Many possible solutions exist, although one can hope to find a 'good' solution by adopting a widely proven technique. Given a set of parameters (for a specific algorithm) that describe the relationship between the observed set of variables, we are faced with an optimization task to find the best combination of values for these parameters. In regression, such a combination would minimize residuals, while for classification, it would minimize generalization error.

Based on the information present (about a system) in given data and additional information that can be acquired by collecting more data, a global minimum of generalization error exists. To achieve this minimum or get close to it, the optimization of algorithm parameters (through methods such as expectation maximization or Newton's gradient descent [Mordecai 2003]) may not seek the global minimum of training error (described further later). Machine learning algorithms surmount complex optimization problems by utilizing heuristics in order to converge (to a solution) relatively quickly. Therefore, the final solution might achieve the minimum of generalization error for given methods and the assumptions underlying them, but may not achieve a global minimum for generalization error either. In the end, we want a solution that is 'good enough': uses a vast majority of the knowledge inherent in the data (small bias) and does not induct something that is not present in the data (small variance). Preceding ideas touch on two important concepts in machine learning: the *No Free Lunch Theorem* [Wolpert 1996] and the *Bias-Variance Dilemma* [Geman 1992].

For classification tasks, a *classifier* is a trained model that contains the mapping of an input *instance* to a class *label*. Based on the induction algorithm used, such a mapping exists in a hypothesis space that is explored during optimization and inherits certain assumptions about the data.

3. Brief Descriptions of Relevant Methods and Concepts

3.1 Definitions

An instance is a an observation or an example that results from an act of measurement or experimentation. An *instance* may also be referred to as a *sample* in statistical parlance. An instance may represent the state of a system at particular point of time or physical properties of an object (among many) or matter. It is the variation across instances that is of special interest to any scientist.

A single instance can be described by multiple attributes. The number of attributes measured or extracted is usually fixed for a set of instances. One may also refer to an attribute as a feature or a variable. Features can be categorical, ranked/ordinal or continuous.

One of these features may be considered the *response* or *dependent* variable. If the response variable is continuous, then the task of mapping the *independent* or *explanatory* variables to the dependent variable is called *regression*. If the response variable is *categorical* or *ranked*, then the task becomes that of *classification*. In this context, the independent variables may also be referred to as *predictor*s and the dependent variable may be called the class variable. Individual values of the class variable are often called *labels*.

3.2 Inductive Machine Learning Algorithms

A specific Machine Learning (ML) algorithm, such as logistic regression, explores a restricted hypothesis space to find an optimal mapping. Neural networks and decision trees [Breiman 1984] were some of the first algorithms in machine learning to gain widespread acceptance and implementation. In order for neural networks to learn a non-linear decision boundary, a multi-layer representation is necessary. The optimization algorithm for such a representation was first proposed by Bryson and Ho in 1969 [Bryson 1969], and successfully implemented in the late 1980s [Rumelhart 1986]. Since then, many algorithms have been invented. Practical ensemble methods, like bagging and boosting have proved to reduce generalization error for many ML algorithms.

The Bayes optimal classifier predicts the probability of membership of an instance to a class. This hypothetical classifier has the highest possible accuracy of any classifier for any dataset. Through an 'ensemble' perspective, the Bayes optimal can be defined as follows:

$$y = \mathrm{argmax}_{c_j \in C} \sum_{h_i \in H} P(c_j|h_i)P(T|h_i)P(h_i)$$

(3.1)

Here y is the prediction by the Bayes optimal, *C* is the set of classes (|C|=2 for the proposed model), *H* is the set of all possible hypotheses, *T* is the training set, and *y* is the predicted class [Mitchell 1997]. The resulting mapping lies in the ensemble space created by the hypothesis space *H*.

3.3 Bias-Variance Tradeoff

A central problem in machine learning or statistical learning is model selection. Models with higher complexity can explain a greater proportion of the variance in the data. This amount of 'explanation' is measured through the R-squared value in regression and as accuracy in classification. As complexity increases, so does the 'explained' quantity. However, as complexity increases, *variance* also increases. If *m*(*Xi*) gives the estimate of outcome variable for training data *Xi*, then *variance* can be defined as

$$E[\ (\ E[m(X)] - m(X_i)\ )^2\ ]$$

(3.2)

where X represents the entire training data from which subsets are drawn as X_i, and the function E[] represents expectation. As *variance* increases, *bias* decreases. If $P(Y|X_i)$ represents the true probability of outcomes given X_i, then *bias* is defined as

$$\{ E[m(X)] - Q(X_i) \}^2$$

(3.3)

where Q is the Bayes optimal classifier. Note that the expectation is calculated over all models built with all X_i.

If X is infinitely large, then a *consistent* induction algorithm is one which produces a model such that $E[m(X)] = Q(X)$. If an algorithm is consistent for all distributions of X and the outcome variable, then it also qualifies as *universally consistent.* Therefore, the bias term may arise from the properties of the data, in which case the data available is not enough to minimize the error produced due to bias, or it may arise from the choice of induction algorithm for the given data, in which case the algorithm may not be consistent.

The overall error of a model, given data *Xi*, is the sum of the above two quantities:

$$m(X_i) - Q(X_i) = E[\ (E[m(X)] - m(X_i))^2\ ] + (E[m(X)] - Q(X_i))^2$$

(3.4)

3.4 ANOVA and Kruskal-Wallis

Probabilistic hypothesis testing helps decide whether a given result is valid by establishing a 'statistically significant' difference between two or more set of variables. This significance can be arbitrarily set to any level and such a decision indicates that the hypothesis backing this result is true. Establishing statistical significance boils down to measuring the difference between a set of variables. The magnitude of this difference is set by the user as the significance level. This level represents the probability of a ‘significant’ result when there is no significance in reality. A few points to note:

- Significance level is set apriori
- Probability of false positive is not a direct but a second order measure (dependent on probability distribution for the test statistic) of the difference between the given set of variables.
- Even if significance at a certain level represents significance in reality, such a proof gives us a ‘yes’ or ‘no’ answer. For a regression or classification task, such a result does not give us a direct measure for the strength of a feature.

By choosing to work with the test statistic itself instead of probabilities, I can eliminate the need to assume a distribution for the test statistic. However, the data must still follow a certain distribution in order to get predictable results from the chosen statistic.

In order to calculate the probability, standard hypothesis tests assume that the data and the test statistic follow certain distributions or classes of distributions. ANOVA [Scheffe 1959] is a parametric method that assumes a normal distribution for the data and an F distribution for the test statistic, $\mathrm{F_t}$.

$$\mathrm{F_t} = \frac{\sum_{\mathrm{c}} \mathrm{n_c}(\underline{\mathrm{x}}_{\mathrm{c}} - \underline{\mathrm{x}})/(\mathrm{C}-1)}{\sum_{\mathrm{ci}} \mathrm{n_c}(\underline{x}_{\mathrm{ci}} - \underline{\mathrm{x}}_{\mathrm{c}})/(\mathrm{N}-\mathrm{C})} \tag{3.5}$$

Here C is the number of groups, c_i represents instance from each group, n_c is the number of instances in group c, $\underline{\mathrm{x}}$ represents the overall mean of all samples, $\underline{\mathrm{x}}_{\mathrm{c}}$is the mean of group c, $\underline{x}_{ci}$ is a sample from group c, and N is the total number of samples. Non-parametric methods do not assume that the data follows a certain distribution. Such methods may still assume that data from different groups or classes follow identically shaped distributions. Like parametric methods, the test statistic is assumed to follow a certain distribution. Kruskal-Wallis [Kruskal 1952] is a non-parametric test whose test statistic must follow a chi-squared distribution. In practice, non-parametric methods provide a more robust way of testing for significance, but do so at the cost of losing data resolution. For example, a continuous variable may be converted to ranks, thereby becoming more discrete.

3.5 Kullback-Liebler Divergence

In information theory, the difference between two given distributions can be measured by the f-divergence. Specifically, the divergence of distribution N from distribution M is

$$D(M||N) = \int f(\frac{dM}{dN})\, dN$$

(3.6)

Kullback-Liebler is a special case of f-divergence where the function *f(x)* has been replaced *x*ln(*x*). For discrete distributions:

$$KL(M||N) = \sum_{i} \ln(\frac{M(i)}{N(i)})\, M(i)$$

(3.7)

KL *divergence* cannot be called a *distance* because it does not satisfy the third condition (known as 'symmetry') from the following conditions.

1 $d(p, q) \geq 0$
2 $d(p, q) = 0$ if and only if $p = q$
3 $d(p, q) = d(q, p)$
4 $d(p, q) \leq d(p, q) + d(p, q)$

KL divergence is an asymmetric measure that is biased towards the reference distribution. It measures the extra bits required to code samples drawn from the reference distribution when a code based on the given distribution (instead of the reference distribution) is used. Unlike a test statistic, divergence and entropy measures may not serve to summarize the data.

3.6 Best-First Search

Exhaustive search evaluates all possible subsets of features in order to find the subset that results in best performance, given by a criteria C_{fs}, of the given classifier. For m features, this results in $O(2^{m})$ combinations. As *m* increases, exhaustive search becomes computationally infeasible and also leads to overfitting by selection of spuriously 'best' subsets [Kohavi 1997]. An alternative is Best-First search, which evaluates $\sum_{i=1}^{m} i$ subsets. Best-First search in forward direction starts by evaluating all *m* features individually. The best feature f_{first} is chosen for the next step. Step two involves evaluating all 2-feature combinations formed by $\{f_{\text{first}}, f_{\text{i}}\}$, where f_{i} is drawn from all remaining features after step one. Step three forms 3-feature combinations with $\{f_{\text{first}}, \text{f}_{\text{two}}, f_{\text{i}}\}$ and so on. Best-First search in backwards direction eliminates features, instead of adding, and starts with the set of all features. At each step, the feature whose elimination leads to the smallest decrease (or equivalently, the biggest increase) in C_{fs}, is discarded. This reduces computational time while searching for a local optimum that may not overfit the data.

3.7 Feature Selection

Many methods have been devised to measure feature strength. Such methods can be divided into two broad categories: heuristic-based methods and wrapper-based methods. Heuristic methods utilize a predefined measure of feature strength with respect to the class variable. An example is *info_gain-ratio,* defined as follows.

$$\text{InfoGain}(\mathit{Class}, \mathit{Attribute}) = \text{H}(\mathit{Class}) - \text{H}(\mathit{Class} \mid \mathit{Attribute})$$

(3.8)

Wrapper based methods utilize an induction algorithm to create a model. Then, according to the performance of the model, the features are either ranked (through some measure of contribution to the model) or ‘best’ subsets are found.

In theory, the task of feature selection can be categorized under the task of parameter optimization for an ML algorithm. For most induction algorithms certain parameters are not tuned/optimized automatically. While the weights assigned to each feature are necessarily optimized when building, for instance, a logistic regression model or a neural network model, other constant parameters such as the number of hidden neurons, learning rate, misclassifications allowed, etc. remain untouched in an out-of-the-box implementation. During additive logistic regression, as the weights assigned to some features may approach *zero*, it would result in automatic feature selection. As such, feature selection can be seen as the task of optimizing a utility vector U that selects/discards each of m features.

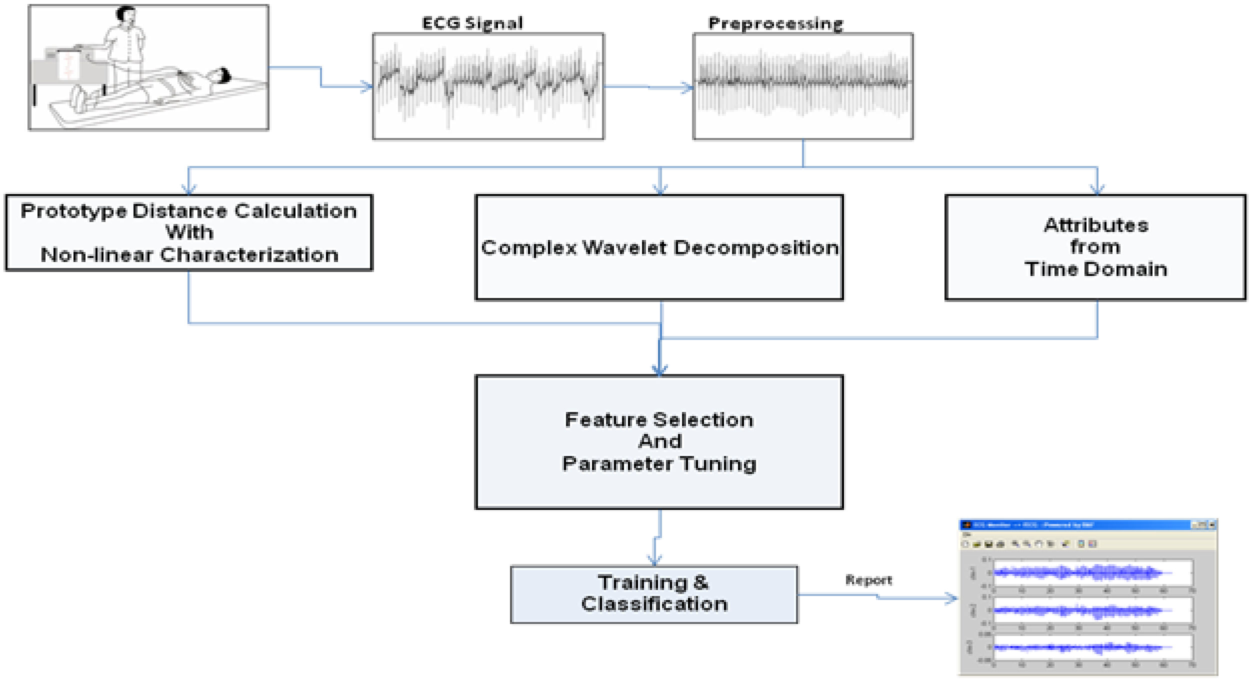
$$U = \{u_{f1},...,u_{fm}\}, \quad \text{where } u_{fi} \subset \{0,1\}$$

(3.9)

Since the wrapper approach involves building numerous models/mappings, only the fastest induction algorithms can be used in wrappers. Simple decision trees, logistic regression, naive bayes are a few examples. The fastest implementations of SVMs are known to be still too slow for use in wrappers for feature selection. However, the combination of linear SVMs and feature ranking has been used successfully for this purpose [Guyon 2002].

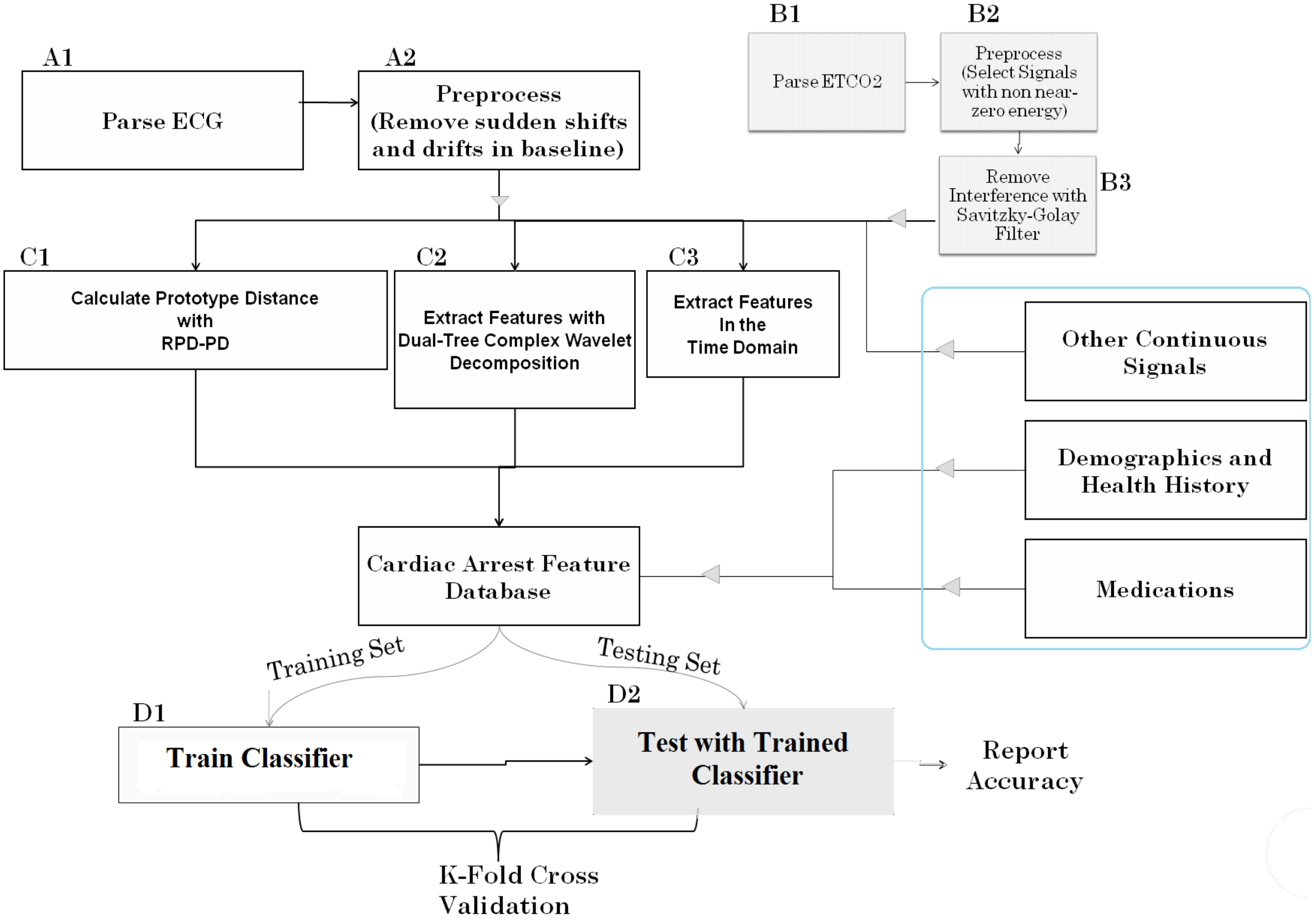
4. Overview of the Decision-Support System

Time-series features were devised in order to distinguish pre-defibrillation VF signals yielding a successful defibrillation from those that did not. The methods for extracting these features are described in Chapter 5. I have developed a novel non-linear method, the Quasi-Period Density Prototype Distance (QPD-PD), with stochastic quasi-periods derived from time-delay embedding. This method focuses on distributions of pseudo-periodicities while accounting for stochasticity in the signal. Parameter selection and feature calculation for the QPD-PD model are geared toward classification. Supervised feature selection (Chapter 7) was performed to identify the most discriminative features. Selection was performed in a nested fashion so as to maintain blindness to the test folds. Simultaneous 10-fold cross-validation was used to evaluate the model. Matlab® software was utilized for all signal-processing needs. Figure 4.1 titled “Overview of Methodology” illustrates the high-level steps of our methodology.


ECG Signal
Preprocessing
Prototype Distance Calculation With Non-linear Characterization
Complex Wavelet Decomposition
Attributes from Time Domain
Feature Selection And Parameter Tuning
Training & Classification
Report

4.1 Overview of Methodology

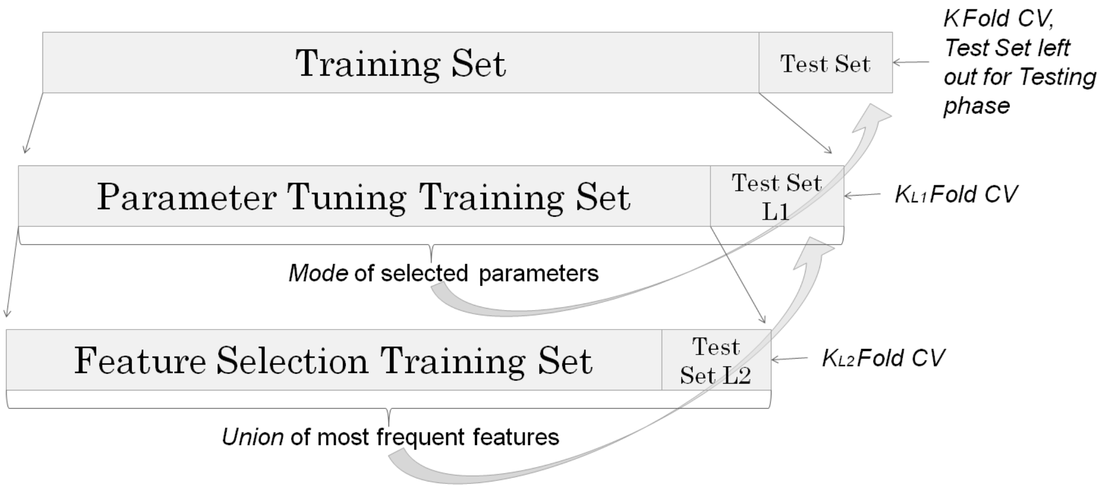
Time-series and complex wavelet features were also extracted from the PetCO2 signal using the same methodology as for ECG signals. The system below was used to develop two separate ML models. The first model predicts the outcome of a delivered countershock as either "successful" or "unsuccessful", while the second model predicts the reccurrence of cardiac arrest after a successful shock has been delivered.


B1
B2
Parse ETCO2
Preprocess (Select Signals with non near-zero energy)
A1
A2
Parse ECG
Preprocess (Remove sudden shifts and drifts in baseline)
B3
Remove Interference with Savitzky-Golay Filter
C1
C2
C3
Calculate Prototype Distance with RPD-PD
Extract Features with Dual-Tree Complex Wavelet Decomposition
Extract Features In the Time Domain
Other Continuous Signals
Demographics and Health History
Cardiac Arrest Feature Database
Medications
Training Set
Testing Set
D1
D2
Train Classifier
Test with Trained Classifier
Report Accuracy
K-Fold Cross Validation

4.2 Overview of the System

Classification

Feature selection, performed with cross-validation on the whole dataset, creates a positive bias in accuracies by indirectly using information from the test set. As such, feature selection must be performed within the training set that is generated for each run of *k*-fold cross-validation. However, using the entire training set leads to over-fitting within the training set, which creates a negative bias in accuracies when the test fold is passed through the model [Kohavi 1997]. To prevent this, and to also select parameters for the learning algorithm in a nested fashion, I employ a twice-nested version of cross-validation.



Training Set
Test Set
KFold CV, Test Set left out for Testing phase
Parameter Tuning Training Set
Test Set L1
KL1Fold CV
Mode of selected parameters
Feature Selection Training Set
Test Set L2
KL2Fold CV
Union of most frequent features

4.3 Framework for Wrapper Based Selection. Twice-nested cross-validation setup. Parameter tuning is performed at Level 1 (L1), where an optimal feature subset has already been selected by cross-validation at Level 2 (L2). $k = k_{L1} = k_{L2} = 10$ folds; same for all levels.

The final model was selected using techniques proposed in Chapters 6 and 7. The final classifier was trained using the Logitboost algorithm.

5. Data Processing

5.1 The Data

The study was approved by Virginia Commonwealth University Institutional Review Board. De-identified cardiac arrest data, for a total of 153 out-of-hospital subjects from a data bank, was provided by the Richmond Ambulance Authority (Richmond, VA) and Zoll Medical Corp. (Chelmsford, MA). Prior to computational analysis, shocks were manually classified as either successful or unsuccessful based on the post-defibrillation ECG segments and data from the pre-hospital care record. Successful defibrillation was defined as a period of greater than 15 seconds with narrow QRS complexes under 150 beats per minute with confirmatory evidence from the medical record or ECG that a return of spontaneous circulation (ROSC) has occurred. Such evidence included lack of CPR resumption over the next minute, mention of ROSC in record, and/or rapid elevation in PetCO2 levels. While others have utilized alternative definitions that incorporate longer periods of ROSC and specific blood pressures, I chose this definition because a shorter timeframe is more clinically relevant in light of a renewed emphasis on minimizing "hands-off" time during the CPR duty cycle as well as the ever evolving treatment paradigms of cardiac arrest. [Berg et al 2010] The short pause allows for ROSC determination and rapid return to CPR if defibrillation was unsuccessful. A total of 358 countershocks were deemed usable for analysis (218 unsuccessful and 140 successful).

Where available, PetCO2 data obtained from capnography was also parsed from the subjects' records. PetCO2 values for a total of 48 pre-defibrillation signal-segments (28 unsuccessful and 20 successful) were used to extract features that could be valuable in predicting the success of a defibrillation in terminating VF, leading to ROSC. Prediction of defiibrillation success is the aim of this study.

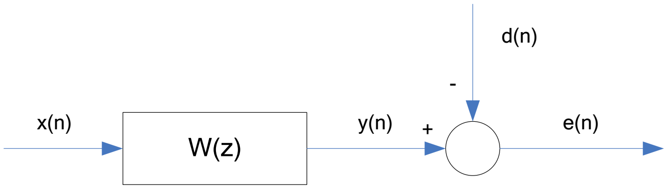
As an additional objective, another level of prediction capability was developed for the system. For a successful shock, an interventionist's strategy can be further guided by predicting occurrence of re-arrest. A total of 104 successful countershocks were labeled as "re-arrest" or "no re-arrest". If a successful shock was followed by any countershock(s) or the post-shock rhythm presented the same evidence as that for an unsuccessful shock (described in the previous paragraph), the countershock was marked as "rearrest". Otherwise, it was labeled as "no re-arrest".

5.2 Motivation

Time-series can contain noise from multiple sources. Here, noise is data that arises from sources that are irrelevant for our purpose. Note that noise may not present itself as ‘outliers’. In general, the word ‘outlier’ is not synonymous with ‘noise’ as an outlier may represent actual data or quantity of interest. For continuous time-series, the case of overlapping noise and data is the norm. In frequency domain, the data can be visualized as a frequency spectrum. Here noise may present itself as a specific range of frequencies that do not constitute our signal of interest. However, there is a possibility that the amplitude of these frequencies may be contributed by both, the noise and the signal. In such a case, frequency-domain transformation may not suffice as a filtering method. Moreover, the signal may not be transformable into the frequency domain due to its inherent properties, such as stochasticity and/or non-stationarity. Other filtering methods, such as wavelet-based filtering, may assume static morphologic properties.

5.3 Adaptive Filters

Filters can also be ‘adaptive’. Kalman filter is possibly the best known robust filter from this class. The theory of Kalman filter follows from that of Kolmogorov-Wiener (KW) filter, which is a type of causal filter but is not adaptive. In order to implement the KW filter, the autocorrelation needs to be calculated from the original signal. The data can be stochastic in nature, but also has to be stationary. KW filter linearly estimates the filtered signal from the original one and minimizes the mean-squared-error.



where *d*(*n*) is the desired signal, *x*(*n*) is the input signal, *W*(*z*) is the Z-transform of the filter coefficients *W*. *e*(*n*) is the error signal calculated by subtracting the output signal from the desired one. Minimization of the expectation of *e*(*n*) serves as the objective. The output of the filter is given by $y(n) = X^T W$.

Setting the derivative of the error signal yields the optimal weights *W*. For practicality, the optimal solution can be found in terms of cross-correlation and auto-correlation as follows.

$$W_{op} = R^{-1}P$$

where $R = E[X(n)X^T(n)]$

and $P = E[X(n)d(n)]$

(5.1)

Kalman filter, on the other hand, is recursive and adaptive. An adaptive filter is able to adjust its transfer function based on some criteria and changing properties of the noise and/or the system. In Kalman’s case, this adaptation is also based on its own prior output. As such, the data is assumed to be arising from a dynamical system and is *not* assumed to be stationary. Noise is assumed to be normally distributed and centered at 0 (filter performs optimally if the noise also has a finite variance and is uncorrelated over time/frequency). Kalman filter minimizes the MSE for the estimate.

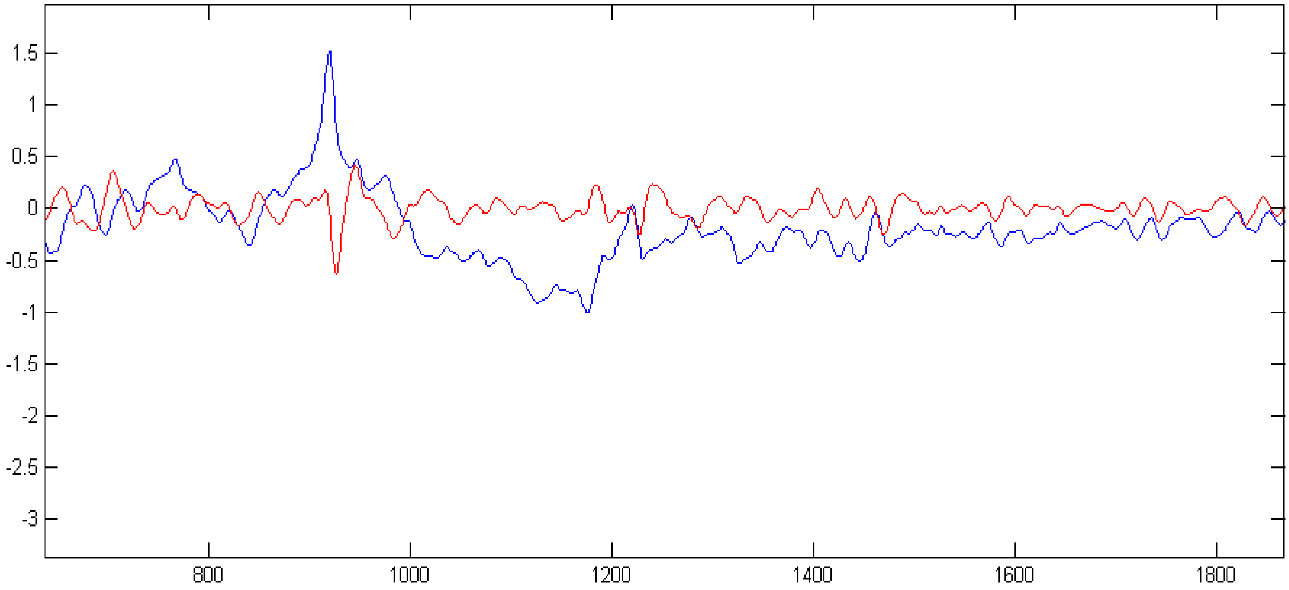
The transfer function of such a filter can be seen as modeling the data, since a future state of the system is being predicted based on a previous state. As such, parameters of such a model can be seen as characteristics of the system that may be predictive of its future state. Since our task does not require a prediction of the post-shock signal itself but a prediction of longer-term post-shock state of the system, I can pose our problem as a classification task. I can then use the parameters of the filter "model" as predictive features that can discriminate between the given classes. I adopt this approach and explain it further in Chapter 7 Section 7.2. For the purpose of removing noise and drift from the data, I take the following approach.

5.4 Proposed Method

Methods described herein correspond to block A2 in Figure 4.2 titled "Overview of the System". Some signals exhibited high frequency noise, which was attenuated by application of the Savitzky-Golay low-pass (smoothing) filter [Savitzky 1939]. High-frequency attenuation was achieved by fitting a moving window, of width *k* data points, to a $p \leq k$-1 degree polynomial by the least-squares method. For a constant *p*, *k* is set to be relatively small when only "slight" smoothing is needed; thereby making the difference between *p* and *k* to be relatively small as well. Simple averaging filters were avoided so as to better preserve the high-frequency content.

Next, sudden baseline jumps caused by interference were removed. The signal was successively 'smoothed' by repetitive application of Savitzky-Golay filter until only the jumps and drifts remained. The resulting signal was then subtracted from the already 'low-passed' signal obtained from the preceding step, yielding the cleaned signal. Filtering steps can be summarized as follows:

- Step 1: Reduce high frequency noise using Savitzky-Golay low-pass (smoothing) filter
- Step 2: De-Trending
  - Step 2a: Successively smooth the signal until only baseline variations and drifts, caused by noise and interference, remain.
  - Step 2b: Subtract the new signal (from step 2) from the signal (from step 1)



5.1 Filtering. Blue: Original signal with a sudden jump around sample 900 and then a drift till sample 1200. Red: Filtered signal displaying physiologic morphology around sample 900 and no drift till sample 1200. y-axis::mV, x-axis::samples.

5.5 Summary

Frequency-domain dependent filtering methods were precluded due to the presence of all frequencies in a baseline jump and the non-stationary nature of data. Traditional high/low pass filters (such as Butterworth) cannot be employed due to spectral overlap. The baseline jump and drift removal is similar to a high-pass filter. In contrast, the signal is not resolved into frequencies. Resulting signals were reviewed by the cardiac care specialist, Dr. Michael Kurz, to confirm physiologic propriety. Preceding plot shows the final signal versus the original one.

While there are myriad existing solutions for removing noise, I believe that a custom solution that is based on close examination of the data is a foolproof way to preserve the data’s integrity while discarding what is irrelevant.

6. Information Driven Stochastic Dynamical Modeling

6.1 Motivation

FT, as utilized by others [Ristagno 2008], performs a linear transformation of a function space such that the original signal (function) is decomposed into multiple sinusoids that are globally averaged. In other words, the Fourier basis is not localized in space/time. Characterizing a short-term/non-stationary, pathological signal requires the assumptions of linearity and periodicity to be relaxed. Limitations of a Fourier based analysis have also been discussed in other studies [Watson 2004], [Kantz 1999].

Wavelet decomposition yields better time-frequency resolution. It uses a mother wavelet, which is a prototype bandpass function, and a scaling function (discussed further in Chapter 7 section a1) to represent the signal. Due to time/space localization of the wavelet coefficients, attenuation or removal of certain coefficients does not lead to global effects. As such, I can perform a non-linear decomposition/reconstruction of the signal through careful selection of detail coefficients. A traditional linear approach would entail selection of coefficients from s-r levels where $r<s$. For instance, I can choose to select only those coefficients that are above a certain threshold level. The wavelet basis has been shown to have attractive properties [Nowak 1998] and wavelet decomposition is widely accepted as a powerful method for filtering, compression, and reconstruction. Because of the time-space localization aspect, wavelets are specifically chosen for non-stationary signals. However, studying properties on dynamical (possibly chaotic) and stochastic data requires techniques from the ‘non-linear dynamics’ domain.

6.2 Theory

Non-linear analysis time-series analysis helps in bridging the gap between deterministic chaos theory and the observed “randomness” of a system. Methods of non-linear time-series analysis arise from the theory of deterministic dynamical systems. The ‘embedding’ theorem [Takens 1981] [Sauer 1991] can be used to construct the phase space from a single variable. Dimensions of the phase space *p* correspond to multiples of the delay *τ*.

$$p_n = [p_n, p_{n-\tau}, \ldots p_{n-(m-1)\tau}]$$

(6.1)

Here, the value of each dimension (from equation 6.1) at time *t* corresponds to the value of the signal at times: $t = i\ dt$, $t = (i+\tau)\ dt, \ldots, t = \{i+(m\text{-}1)\tau\}\ dt$. Here *dt* is the time between each sample, i.e. $(Sampling\ Rate)^{-1}$. For a fixed *m*,

1) *τ* has to be large enough so that the information at *i*+*τ* is significantly different from the information at *i*. Once a proper *τ* is chosen, it will give us enough information to construct the phase space.
2) On the other hand, the system may appear not to have any memory if *τ* is chosen to be too large.

Depending on the actual amount of information (about the system) present in the signal segment (which may partly be a function of the length of the segment), the ‘loss of memory’ is also a characteristic of chaotic systems, where a small change in initial conditions produces a large divergence in trajectory in the phase space. It is important to note that the effect of incomplete information about a complex dynamic system (such as that of a patient in cardiac arrest) may produce properties that are similar to that of a chaotic system. In both cases, the system will appear to lose the memory of its initial state and will therefore become unpredictable in time.

The Lyapunov exponent quantifies the rate of divergence of two trajectories in the phase space. If the initial separation of two trajectories is given by ΔS_0, they diverge according to the rule

$$|\Delta S(t)| = e^{\lambda T} \cdot |\Delta S_0|$$

(6.2)

For a discrete time system, where S_0 is the starting point of the orbit, and $S(t+1)$ is a function $S(t)$, the lyapunov exponent can be expressed as

$$\lambda = \lim_{n \to \infty} \left(\frac{1}{n} \sum_{i=0}^{n-1} \ln \left| \frac{dx_{n+1}}{dx_n} \right| \right)$$

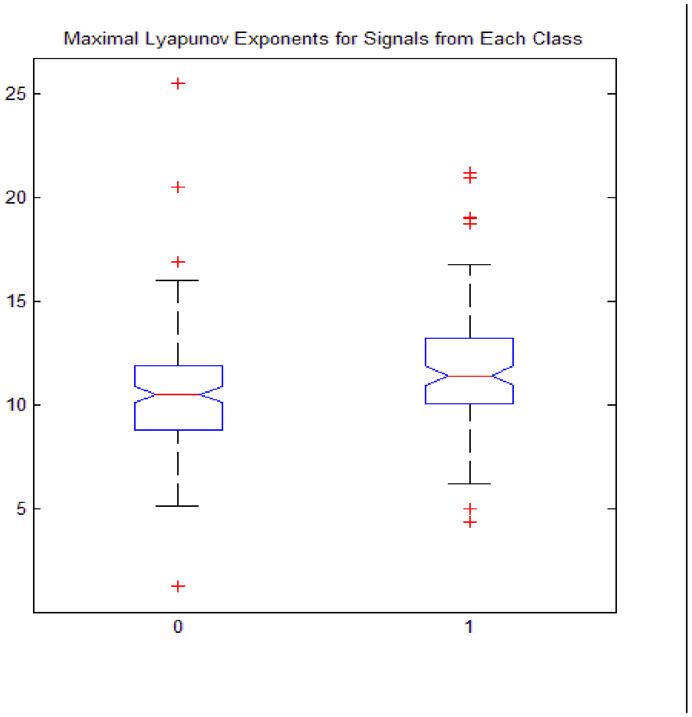


Figure 6.1. The Lyapunov Exponent of VF. Boxplots representing distribution of the Lyapunov exponent (y-axis) calculated for all signals. x-axis: "0" signifies "unsuccessful" class, while "1" signifies "successful" class.

A positive Lyapunov exponent indicates that the underlying system is chaotic. Topological mixing is a necessary property of a chaotic system [Vellekoop 1994], but proving this property is not necessary for our proposed model. The quasi-period plots (Figure 6.2), can represent deterministic, non-dynamical, dynamical, chaotic, as well as stochastic properties of a system.

6.3 Proposed Method

Methods described herein correspond to block C1 in Figure 4.2 titled "Overview of the System. I begin by projecting our data *x*(*t*) onto a state space *p*(*t*). Time delay embedding is used to project the data series into multiple dimensions of a phase space. Each dimension of the phase-space itself represents a time-delay. Note that $m{\cdot}\Delta T{\cdot}\tau$ gives us the span of each point in the phase space onto the time-domain signal.

The False Nearest Neighbor method [Kennel 1992] has been used successfully for selection of a proper value of *m*. FNN seeks to find the degrees of freedom, represented by *m*, that are inherent in the signal and sufficient to minimize divergence of neighboring points in phase space as time evolves. The goal, again, is to construct the phase space that aptly models the deterministic nature of the signal. As such, this approach may not be suited to data that have a large stochastic component. The concept of finding an appropriate degrees of freedom, however, remains highly relevant. In a low-energy (than non-pathologic ECG) VF signal, the degrees of freedom may be smaller as the regular pacemakers and the complex beats associated with them are lost (explored later).

The concept of recurrence [Kohavi 1997] can be interpreted as measuring the level of aperiodicity in the data.

$$p(t) \subset hypersphere(p(t+\delta t), r)$$

(6.3)

Here, the data projected onto a state-space is *p(t)*, *r* is the radius of a hypersphere defined around a state *p*(*n*) (where *n* is a specific value of *t*). Following the data in state space for a given *n*, *δt* is the recurrence time at which data falls within the sphere, once again, after having left it.

Quasi-Periodicity/Recurrence

Periodicity is a special case of recurrence when *r*=0 and all 'states' exhibit the same δt. Alternatively, through a perspective of harmonic motion, recurrence can be seen as quasi-periodicity. As a special case, if all the frequencies wi corresponding to the harmonic components have zero amplitude, then the signal can be called *aperiodic*. It is important to reiterate here that Fourier analysis yields a definite result only in the *periodic* case.

Autocorrelation and mutual information have been suggested [Kantz 1999] for selecting a proper combination of dimension m, time delay τ, and radius r. However, our objective is to separate the two classes, 'successful' and 'unsuccessful', as far as possible based on a given distance metric and data without losing generalization power. Neither class presents apparently periodic signals. As such, the novel parameter selection regime, as proposed here, finds a 'structure' in the signal, defined by dimensions *m* and time delay *τ*. This structure would differ significantly in its quasi-periodicities for the two classes, where the quasi-periodicities are conditional upon the pre-selected value of *r*. Therefore, *r* can also be tuned for our classification purpose as well. *r* is usually chosen to be a small value if at least one of the classes presents deterministic data. A relatively larger tuned value can be seen as yielding stochastic quasi-period densities (QPD) for both classes.

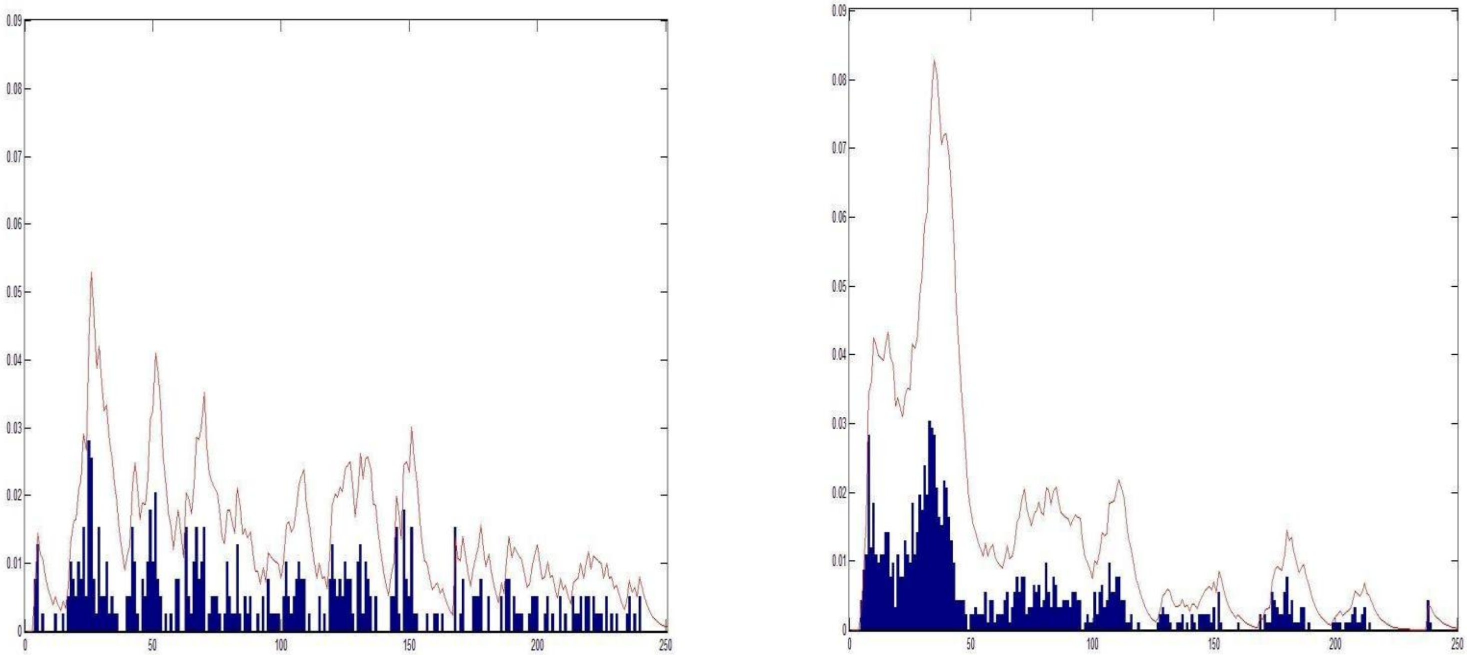
Proper parameter selection is essential in rendering this method useful. Four post-defibrillation signals that exhibited regular sustaining sinus rhythms, with narrow complexes, were selected as successful prototypes. Four defibrillations that induced minimal change in the ECG or were immediately followed by smooth VF, with no conversion, were selected as unsuccessful prototypes. It should be noted that selection of pre-shock signals is 'blind' in the sense that only post-defibrillation segments are considered during selection.

For 10-fold cross validation and a dataset with n instances, each training set would contain *n*-(*n*/10) samples, thus leaving out the test set. A range of possible values was defined for each parameter. Quasi-period density was then calculated for each combination of parameter values and each signal in the training-set (TS) and prototype-set (PS). I define the metric *KD* (Equation 6.4) to calculate the pairwise distances from each TS density to all PS densities.

$$KD = \sum_{i=1}^{T} (1 + D_i^c).(D_i^c - D_i^S)^2$$

(6.4)

Here, *s* stands for a given signal while *c* can stand for any of the other signals; ${D^c}_i$ and ${D^s}_i$ are the density values at a certain period *i*. *KD*, being inspired by the Kullback-Leibler distance, is biased towards the characteristics of *c*, but unlike KL, can also serve to measure the distance between two discrete distributions. Given classes A and B, a density from class A is subdivided into non-overlapping windows or ranges, which are compared (by *KD*) with respective windows of other densities (Figure 6.2 titled ‘Quasi-Period Density’). Therefore, our optimization is performed over a total of four variables, *m*, *τ*, *r*, and *window*. A description follows.



6.2 Quasi-Period Density Function. QPD for a successful shock (left) and QPD for an unsuccessful shock (right). Blue bars represent the normalized amplitude (y-axis) for each pseudo period (x-axis). Red line represents QPD convolved with the exponential function. If most of the Quasi-Periods are clustered within a small subset of values, as is the case above (right), the convolution helps quantify that fact.

Classes are maximally separated by maximizing the quantity *sep* (equation 6.5) as follows.

$$\arg_{m,\tau,r,win} \max\,(sep)$$

Sep represents closeness of all TS signals to PS signals in their own class (and remoteness from the opposite class), while also accounting for differential variation in within-class distances for the two classes. I deem this normalization necessary, as data in one class may be more homogenous than data in the other.

$$sep = \sum_{i}^{L} \frac{(\overline{KD_i^B} - \overline{KD_i^W})}{\max(\sqrt{\frac{1}{C^B}\sum_{j=1}^{C^B}(KD_i^j - \overline{KD_i^B})^2}, \sqrt{\frac{1}{C^W}\sum_{j=1}^{C^W}(KD_i^j - \overline{KD_i^W})^2})}$$

(6.5)

Here, L is total number of TS instances/defibrillations. For a given i, KD^B and KD^W are means of between-class and within-class distances, respectively, to instances in PS. C^B and C^W are total number of PS instances in the opposite class and i's own class, respectively.

Each input signal from the test set is then compared to each prototype in both classes. The following distance is calculated as two features, sKD_B and sKD_W, to be used in the predictive model for a signal s.

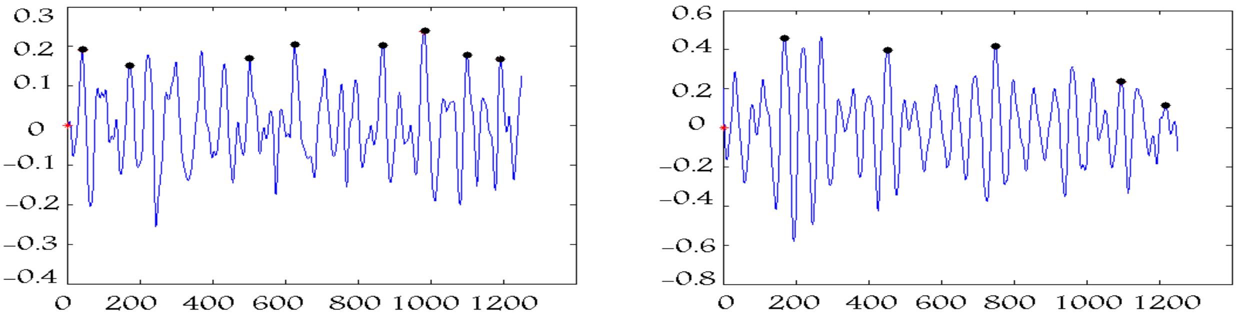
$$sKD_{B,W} = \frac{1}{Q}\sum_{p=1}^{Q}\{\sum_{i=1}^{T}(1+D_i^p)\bullet(D_i^p - D_i^S)^2\}\bullet \mathrm{sgn}(D^p - D^S)$$

(6.6)

Here, Q is total number of signals in PS for a given class, T is longest period in the chosen window, D^P and D^S are vectors representing densities of the prototype and s, respectively, and *sgn* is the sign or signum function. The average *sKD* for each class serves as an attribute of a given signal.

The Pole-Count Feature

Methods described herein correspond to block C3 in Figure 4.2 titled "Overview of the System. Time-series features are based on a priori reasoning that ROSC yielding VF waveforms exhibit more activity, having properties of the coarser VF, as described above. An illustration of the Pole Count feature (Figure 6.3 titled “Polecount Attribute”) depicts the variations in fibrillation activity of the heart along the lead II axis (sampled at 250Hz) [Shandilya 2011], and may at least partially represent the extent of homogeneity in VF across classes.



6.3 Polecount Attribute. Number of peaks signified by dots, quantifies variation in the pre-shock waveforms leading to an unsuccessful shock (left) and a successful shock (right). X-axis: samples, Y-axis: mV

A dynamically adjusting threshold is used to find a minimum number of maxima, V_{mx}, in the signal. Pole-Count feature is then calculated as the number of maxima that satisfy the following condition.

$$V_{mx}^{i} > V_{mx}^{i-1} + 1.2 \times \sqrt{\frac{1}{N}\sum_{j=1}^{N}(V_{mx}^{j} - \overline{V_{mx}})^{2}}$$

(6.7)

Here, V_{mx} is the vector of all maxima and N is the length of this vector. Next, signal attributes/features are derived from the complex wavelet domain.

6.4 Summary

Theories from non-linear dynamics yield powerful techniques to characterize, decompose, and transform continuous time-series having auto-recursive properties. As a result, assumptions of linearity, stationarity, and even determinism can be relaxed. The transformation achieved through the proposed method is into the (quasi) frequency domain.

Figure 6.2 titled "Quasi-Period Density" shows the quasi-period spectrum for a typical case from each class. Convolving each spectrum with the exponential function/kernel creates the probability density function and helps quantify the difference between the two spectra. If most of the amplitude is clustered in neighboring Quasi-Periods, as is the case above, the convolution results in a higher peak for that region.

Concepts from theory of dynamical-systems also yield other methods that can be implemented without time-delay embedding. Such a method for feature extraction (as in Chapter 7 Section A2) would find more similarity with control theory and adaptive filters.

7. A Broad Feature-Selection Framework

This chapter delves into the broad problem of feature selection. I take a variance-reduction perspective to tackle the problem. Multiple objectives are explicitly stated within the 'Motivation' section 7.2.

7.1. Theory

The simplest decision function can be formulated as a weighted sum of individual feature values.

$$f(X) = W.X + b$$

(7.1)

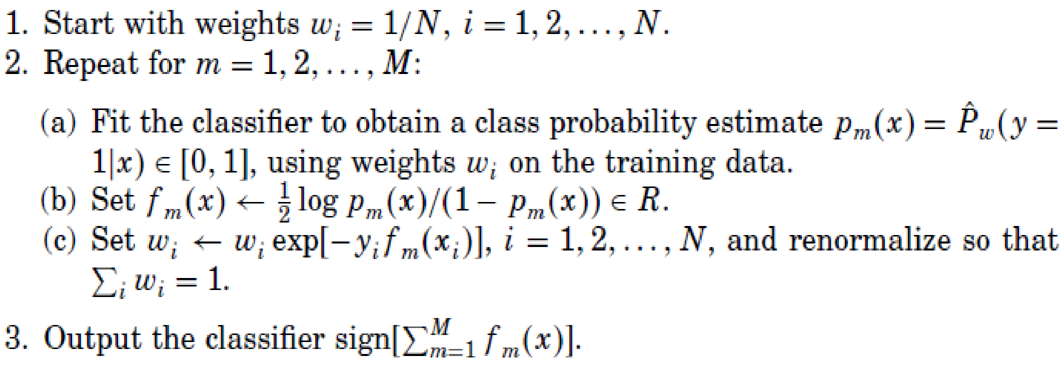
Here, upper case letters represent vectors and *b* is the bias term. This forms the basis for Fisher's Linear Discriminants (also known as LDA). LDA constructs a linear decision boundary and assumes that the features are normally distributed with respect to each class label, ie. the probability P($X|yi$) where *yi* is a specific value of the outcome variable arises from a normal distribution. While logistic regression is functionally equivalent to LDA, the process of optimization of coefficients for logistic regression allows for non-normal distribution of features. For a binary class variable, a logistic regression model can be expressed as

$$\mathrm{P}(yi|Xi) = e^{B.Xi} / (1+\mathrm{e}^{B.Xi})$$

(7.2)

The decision boundary for a logistic regression model is linear. In other words, the decision boundary created by logistic regression within n-dimensional feature space can be represented in *n-1* dimensions. Note that each feature in this space does not have to be linearly separable. In other words, for binary classification problems, the feature does not have to have binomial distribution with each peak corresponding to each class. However, it does make intuitive sense that such features would make for strong contributors to a linear decision boundary. Objective: I extend this idea to find maximum information content through appropriate heuristics and statistical tests (next chapter).

‘Bagging’ and ‘Boosting’ methods have made crucial contributions in enhancing the performance of induction algorithms in the past couple of decades. Bagging [Brieman 1996] uses multiple models built from a base classifier and assigns the class predicted by a majority of the classifiers as the final predicted class for a given instance. Boosting differs from bagging in a significant way. Boosting can involve multiple iterations of a base classifier that actually interact with each other. Also, the final prediction does not have to be based on a majority vote. The individual outputs of iterations can be less granular by consideration of real-valued classifier output. Adaboost is a classic example that serves as a powerful boosting method. Adaboost can be seen as training a weak classifier ‘more’. Between each iteration, a weight is assigned/modified for each instance according to a loss function (steps b and c below). The resulting effect is increased weights for instances that were misclassified during this iteration.



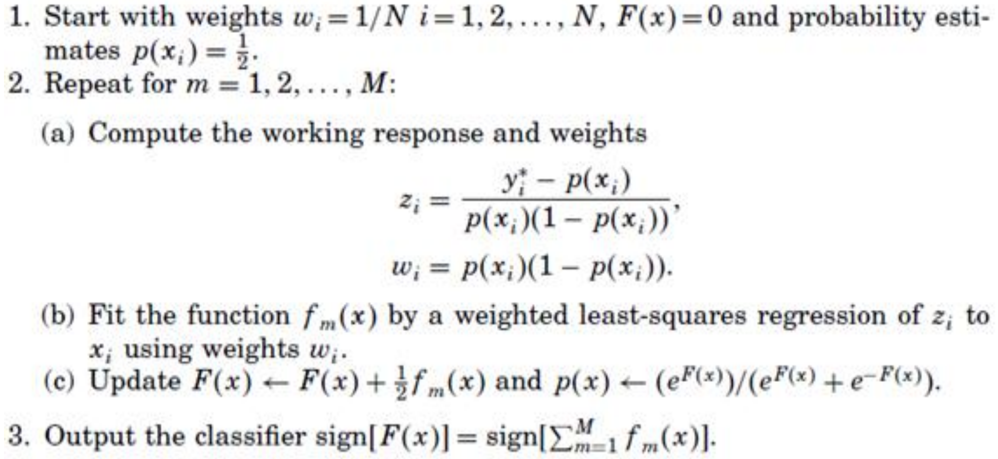
```
1. Start with weights $w_i = 1/N$, $i = 1, 2, \ldots, N$.
2. Repeat for $m = 1, 2, \ldots, M$:
   (a) Fit the classifier to obtain a class probability estimate $p_m(x) = \hat{P}_w(y = 1|x) \in [0, 1]$, using weights $w_i$ on the training data.
   (b) Set $f_m(x) \leftarrow \frac{1}{2} \log p_m(x)/(1 - p_m(x)) \in R$.
   (c) Set $w_i \leftarrow w_i \exp[-y_i f_m(x_i)]$, $i = 1, 2, \ldots, N$, and renormalize so that $\sum_i w_i = 1$.
3. Output the classifier $\mathrm{sign}[\sum_{m=1}^{M} f_m(x)]$.
```

In the 1998 study [Friedman 1998], Adaboost is proven to minimize a loss function that is very similar to the logistic function. The authors propose Additive Logistic Regression (LogitBoost), which works in the same way as Adaboost but uses a simpler function fiter(*X*) of the probability estimates *piter*(*X*),

$$f_{iter}(X) = p_{iter}(X) - (1-p_{iter}(X))$$

(7.3)

in order to update the weights.



```
1. Start with weights $w_i = 1/N$ $i = 1, 2, \ldots, N$, $F(x) = 0$ and probability estimates $p(x_i) = \frac{1}{2}$.
2. Repeat for $m = 1, 2, \ldots, M$:
    (a) Compute the working response and weights
            $z_i = \frac{y_i^* - p(x_i)}{p(x_i)(1 - p(x_i))},$
            $w_i = p(x_i)(1 - p(x_i)).$
    (b) Fit the function $f_m(x)$ by a weighted least-squares regression of $z_i$ to $x_i$ using weights $w_i$.
    (c) Update $F(x) \leftarrow F(x) + \frac{1}{2} f_m(x)$ and $p(x) \leftarrow (e^{F(x)})/(e^{F(x)} + e^{-F(x)})$.
3. Output the classifier sign$[F(x)]$ = sign$[\sum_{m=1}^{M} f_m(x)]$.
```

Since $p_{iter}(X)$ can be estimated directly with logistic regression (as opposed to calculation at the leaves of a decision tree), I use logistic regression as the classifier for Logitboost [Chapter 7 Section B). As in Adaboost, the updating of weights at each iteration results in quicker convergence than bagging. Boosting being an active learning process that progresses through multiple iterations, reduces bias as well as variance (as opposed to bagging which counters only variance). The same concept of boosting iterations is known as ‘epochs’ in the Neural Network context.

7.2 Motivation

A practical induction algorithm will degrade in performance when presented with irrelevant features. Even in the case of bagging and boosting algorithms, a large number of irrelevant features leads to reduction in performance (~5% for our dataset). A well known fact in the ML field is that *a large number of features can easily lead to overfitting*. This statement describes a special source of high variance in the inducted model. The overarching concept is that of high variance, which can also arise from a highly complex model built with a small number of features. Complexity of a model can be varied by growing trees till a small number of instances belong to each leaf, for instance, or by varying the cost or parameter of SVMs or by changing the gamma parameter of the Radial Basis Function kernel (used in multiple ML algorithms). Such an increase in complexity is akin to drawing a decision boundary in high-dimensional feature space. Even for a hyperplane (linear decision boundary) drawn with a large number of irrelevant features, generalization performance can be abysmal.

For the optimal Bayes classifier, adding irrelevant features cannot reduce its performance. However, this is true only for the Bayes classifier because it has access to the entire concept space and can tap into the complete set of information in the feature space. In statistical terms, the underlying distribution is known. In the real world, the underlying distribution is inducted from an incomplete set of information. Therefore, it is necessary that this incomplete set be a relevant one. Furthermore, since a given algorithm explores only a subset of the concept space, the optimal feature subset will depend on the ML algorithm chosen. Objective: To select such a relevant set, I therefore choose an *upstream-sensitive* approach to model building (described further later).

As discussed in second chapter, there are two broad approaches to feature selection. Heuristics yield individual attribute ranking, which is different from ranking by weights (discussed later). For heuristic ranking, each feature's strength is evaluated individually with respect to the class labels. There are two main weaknesses with this approach. The first and more important one is that interactions among features are not considered. Secondly, a given heuristic for feature strength may not quantify the same knowledge inherent in the feature (with respect to class variable) as what the ML algorithm may utilize. Herein, we may end up exploring the intersection of two concept spaces, thereby adding more bias.

For individual ranking, each feature's strength is measured individually through a heuristic, while 'ranking by weights' is a wrapper-based approach where the whole model is built before weights are considered. All feature weights are optimized simultaneously to contribute to the same decision boundary. Then, the feature with larger |*w*| values, where *w* is the weight, are ranked higher. These weights can also be derived for a perceptron model, logistic regression model, or SVMs. Note that the weight assigned to each feature would change if the decision boundary was reconstructed with different assumptions (for instance, a non-linear versus linear boundary). Such wrapper-based ranking is more aptly described as '*ranking geared for subset selection'* since the ranking criteria is calculated for nested subsets. Except for the first feature, all other features are evaluated in combinations with other features. Unlike the traditional wrapper approach, misclassification error is not calculated. Eventually, a certain number of top ranking features are chosen by some criteria to build the model.

For *m* features, the traditional wrapper approach to feature selection explores all subsets of features that are <=*m* in size. Exhaustive enumeration and evaluation of all subsets for a large number of features is computationally infeasible. Even for a small number of features, exhaustive search has been shown to overfit [Guyon 2002, Kohavi 1997]. Therefore, even when it *is* feasible to do exhaustive search, a middle path between greedy and exhaustive search should be chosen (by adopting a method like Best-First). Weight-ranking algorithms are desirable because they cut-down computational expense even further (See Methods).

While model-based ranking accounts for some interactions among features, it does not evaluate redundancy of information among them. Features containing very similar information about the classes would be ranked closely. Objective: Another objective of this work is to select a subset that consists of non-redundant features. Reducing dimensionality by eliminating highly redundant features would reduce model complexity, as discussed earlier, thereby reducing variance.

Projecting the feature space onto a new set of orthogonal axes *Z* is a common technique utilized in many fields ranging from social sciences to microbiology. Usually, this is done to visualize 2 to 3 dimensions of *Z* with respect to classes. The third dimension (representing class) can be added as a color scheme to color the points plotted in two dimensional space (Figure titled “Figure 6.2”). Each axis Z_i in the new coordinate plane is a linear or nonlinear combination of the original features such that it is uncorrelated with other features $Z_1,Z_2,...,Z_{i-1},Z_{i+1},...Z_m$, where m is the total number of features in both, the old and the new coordinate spaces. Variance observed in the original feature space is re-projected in decreasing order of magnitude from the first new dimension to the last one. The technique is used with the hope that the first few dimensions of the new coordinate space *Z* will represent a large majority of the total variance, and that the rest of the dimensions/features can be discarded by making the assumption that the variance represented in them is spurious [Duda 1973]. As such, the modeler would have performed dimensionality reduction for subsequent model building. Note that this is an unsupervised approach with an ‘unsupervised’ assumption about the variance in discarded Z_i. Furthermore, a majority of the variance can be projected onto the first few dimensions only if components of the original feature set is highly correlated. Otherwise, the technique is rendered useless through the traditional perspective. Instead, I look at this method as having the utility of yielding non-redundant features, even if 95% (a commonly used quantity) of variance cannot be represented by the first few reconstructed features.

7.3 Proposed Method

The feature set was first projected onto a new orthogonal set. Each dimension in the new orthogonal space is formed by linear combinations of the original features. With this technique [Duda 1973], each new dimension has an eigenvalue that quantifies the proportion of total variance covered by that dimension. By discarding a total of 1% of the total variance, about 40% of the features from the new set could be discarded. It follows that the original set consisted of highly redundant features. New features were discarded starting from the one with the smallest eigenvalue and continuing till a value close to 1% was reached.

The techniques described herein correspond to blocks D1 and D2 of figure 4.2 titled "Overview of the System" and to the entire figure 4.3 titled "Framework for Wrapper-Based Selection". The feature space was searched by employing Recursive Feature Elimination by Weights (RFE-W) with Support Vector Machines (SVMs) [Guyon 2002]. For a linear SVM, the decision function is given by,

$$f(u) = \sum_{k=1}^{n} w_k u_k + b$$

(7.4)

The weight *w* of each feature, u_k, indicates the extent of each feature's contribution to the classifier's continuous output, and n in the total number of features. RFE-W starts by building a model with all the available features. The one with the smallest |*w*| is eliminated. At each subsequent step, the model is rebuilt and the elimination is repeated. RFE-W is similar to Best First Search with a backwards elimination approach. In contrast, by using *w*, RFE-W can reduce *n* runs of the induction algorithm to 1 run at each elimination step. The key difference is that the elimination is based on the value of *w*. Accuracy of the trained model is *not* calculated.

When producing ranks with *k* fold cross-validation within the training set, I end up with <=*k* ranks for each feature. I then choose the median rank as the final indicator of predictive strength of a feature. As described in section A, the final ranking actually represents nested subsets where the top 5 features are a subset of the top 6 features and so on. This is true because the 6th feature has only been evaluated (eliminated based on *w*) when 5 other (top-ranked) features were also used to build the model.

New Approach for Further Reduction in Model Variance

Wrapper based methods calculate accuracy of the different models and thus provide a direct measure of generalization accuracy. A 'best performing' feature subset can be defined as one that leads to the highest average (cross-validated) accuracy for a given nested run. Here I can put-forth two contrasting solutions: Either a subset that performs best for the greatest number of nested/inner runs can be chosen (thereby, partially accounting for variance or random effects in the data) or, in case where no single subset is chosen for a majority of the inner runs, a union of all chosen subsets (one for each inner run) can presumably yield the best performing feature subset for the outermost test fold. The first approach may have a high bias, especially if the number of folds for which the 'best' subset chosen is relatively small (say <70%). The latter approach may include spurious features. I therefore choose a middle path. Note that two levels of nesting were used to select features *and* parameters in order to remain blind to the test fold while still being able to use cross validation for selection purposes. In order to observe variance in feature selection within the training-set at the top most level, selection-frequencies *fs* were generated for each feature as follows.

$$fs = \frac{S_{L2}}{k_{L1} \cdot k_{L2}}$$

(7.5)

Here, S_{L2} is the number of all inner runs at level 2 (see Figure 4.3 titled "Framework for Wrapper-Based Selection") for which the feature was selected. k_{L1} and k_{L2} are the number of cross-validation folds at level 1 and level 2, respectively. These frequencies showed that 3 to 5 features were selected for only 20% of the innermost runs, indicating further room for reduction in variance through elimination of these spurious features. As an alternative to the traditional wrapper approach [Kohavi 1997], I formulate a new data matrix with features that were found to be members of the best-performing feature-subsets for at least 70% of the runs. This new approach boosted accuracy by approximately 3% without violating blindness to the outermost test folds.

An Upstream-Sensitive Approach

As the dataset is imbalanced, with unsuccessful to successful ratio of about 2 to 1, classification must be cost-sensitive. The 'cost' refers to the cost of a misclassification as represented by the objective function (accuracy) being maximized. In cost-sensitive classification, the penalty for misclassifying an instance from the smaller class is increased. This affects the optimization of the induction algorithm such that the decision boundary is drawn to separate the two classes and not just to maximize accuracy.

However, a cost insensitive approach upstream, i.e. feature selection, may preclude some features that would contribute to a decision boundary strictly between the two classes. In the absence of such features, even cost-sensitive classification yields a decision boundary that is drawn to maximize accuracy only. Therefore, in order to compensate for the smaller class, false negatives were penalized twice as much as false positives during wrapper-based selection. In other words, feature ranking through RFE-W-SVMs was done with a 2:1 cost of misclassification, where 2 corresponds to the successful class. Expectedly, this changes the ranks of the features when compared to non-differential (same across classes) cost of misclassification.

7.4 Summary

Once a good subset of features is selected, the choice of induction algorithm becomes significantly irrelevant. If the subset has been chosen with a “linear decision boundary” assumption, creating a non-linear decision boundary may lead to unexpected results. However, assuming an upstream-sensitive approach, the strong subset would yield good results with most induction algorithms. Guyon and colleagues [2002] support this notion and note that “*features selected matter more than the classifier used*”. A such, it was appropriate to focus our attention on the subject of feature selection. The choice of algorithm induced <=1.5% variation in accuracy for the highest performing algorithms (Additive Logistic Regression, Random Forest [Breiman 2001], Functional Trees [Gama 2004]). After feature selection, the accuracy showed increases between 3.8% and 5%.

8. Regularization

8.1 Theory

As a professional or academician in the field of statistical/machine learning, building data models necessarily involves extracting the most out of data and thus building the highest performing models. Fortunately, experience in the field teaches us the trade-offs involved in the model building process. When unsure of our instincts or when desirous of finding proper rationale for them, I have the luxury of delving into theory facilitated by the likes of Vapnik, Chernovenkis, Geman, Bienenstock and Doursat [Vapnik 1971],[Geman 1992]. While feature selection (previous chapter) is also a process that reduces model complexity/variance by finding the optimal weight-coefficient binary vector, the discussion here leads up to methods for selection of other parameters of an algorithm.

In engineering, the practice of preferring simpler solutions over complex ones is usually rationalized by citing Occam's Razor. Increasing model complexity decreases *bias*, while *variance* increases (see Chapter 2 for formal definitions). In order to contain this variance, and to thereby preserve bias, a regularization term can be used to indicate convergence to an apriori minimum of generalization error. For a model m, let complexity be indicated by $C(m)$. Then, minimizing the following term G_e, defined as

$$G_e = E_{tr}(m) + \beta \cdot C(m),$$

(8.1)

would yield the best model in terms of G_e. Here E_{tr} is the training error, and β controls the level of reduction in variance. A high β would yield low variance in the 'best' model found by minimizing G_e, while a low β would yield high variance. Therefore, the first term in the previous equation can be seen as the *bias* term, while the second term serves as a measure of *variance* in terms of complexity. It is important to realize that *beta* may not represent a single parameter of an algorithm in reality. Varying any algorithm parameter that can serve to vary the complexity of the final model can conceptually be seen as varying the second term. Vapnik and Chernovenkis [1971] put forth the framework of 'Structural Risk Minimization' where VC-dimension can be calculated for a specific algorithm and can be used for regularization. The VC dimension essentially serves as β. However, VC dimension has theoretical basis and needs to be calculated for the algorithm used. It cannot be empirically adjusted like one of the other parameters.

8.2 Proposed Solutions

8.2.1 Wrapper-Based Method

The techniques described herein correspond to blocks D1 and D2 of Figure 4.2 titled "Overview of the System" and to the entire Figure 4.3 titled "Framework for Wrapper-Based Selection". Wrapper-based parameter tuning [Kohavi 1997] implements an implicit form of regularization that is achieved by separating the training and evaluation (or nested test) sets. Since information is *indirectly* used from the evaluation sets, by calculating the evaluation error E_{ev}, an information gap is intentionally induced between the learning phase and the second objective function E_{ev}. Theoretically, we can view this approach as replacement of the term G_e with E_{ev}. The following is assumed to be true.

$$E_{ev} = E_{tr}(m) + \beta \cdot C(m) \tag{8.2}$$

E_{tr} is minimized for the given parameters and E_{ev} is calculated directly. The variance term is largely ignored because it is assumed to reflect in the total error E_{ev}. If the algorithm consists of multiple parameters (in addition to the ones trained during the regular learning process) that can be varied, then the number of possible combinations can be quite large. Consequently it becomes easier to underestimate G_e. In other words, E_{ev} would have a small bias and large variance such that $Eev < Ge$. So the actual case can be represented as follows.

$$G_e = E_{ev}(m) + ϭ\cdot C(m)$$

(8.3)

Here, $ϭ\cdot C(m)$ represents the variance resulting from induction on the evaluation sets. Such overfitting can also happen when tuning only a few parameters with small or non-homogenous datasets. The terms 'small' and 'non-homogenous' represent the lack of information in the sample set to represent the underlying distribution. To reduce variance further, I find k optimal models by minimizing E_{ev} for each of k subsets within a training set and assume the following.

$$E_{te} = median(E_{ev}(m_1),...,E_{ev}(m_k)) + c^*|A|/k$$

$$\text{where } A = distinct\{m_1,...,m_k\}$$

(8.4)

where the second term represents variance among the k models, $|A|$ is the cardinality of A, c is a scaling factor, and the first term estimates error on the outermost training set. E_{te} can obviously be calculated as the error on the outermost test set. From these k models, I select the final model M as the most common one from 1 through k. Referring to Figure 4.3 titled “Framework for Wrapper-Based Selection”, the combination of parameters that was selected most often (at level 1) among k selections (one for each test fold), i.e. *mode* of the selected combinations, was used for final classification of instances in the outermost test fold.

The method proposed above is k times more expensive computationally than wrapper-based feature selection. As wrapper-based methods are already known to be computationally expensive, the proposed method is applicable where a highly accurate model is desired, such as for medical applications. Note that this method was developed for highly non-homogenous datasets in the first place. To further reduce variance through wrapper-based model selection without increasing computational complexity, I propose an alternative ‘high platform’ method.

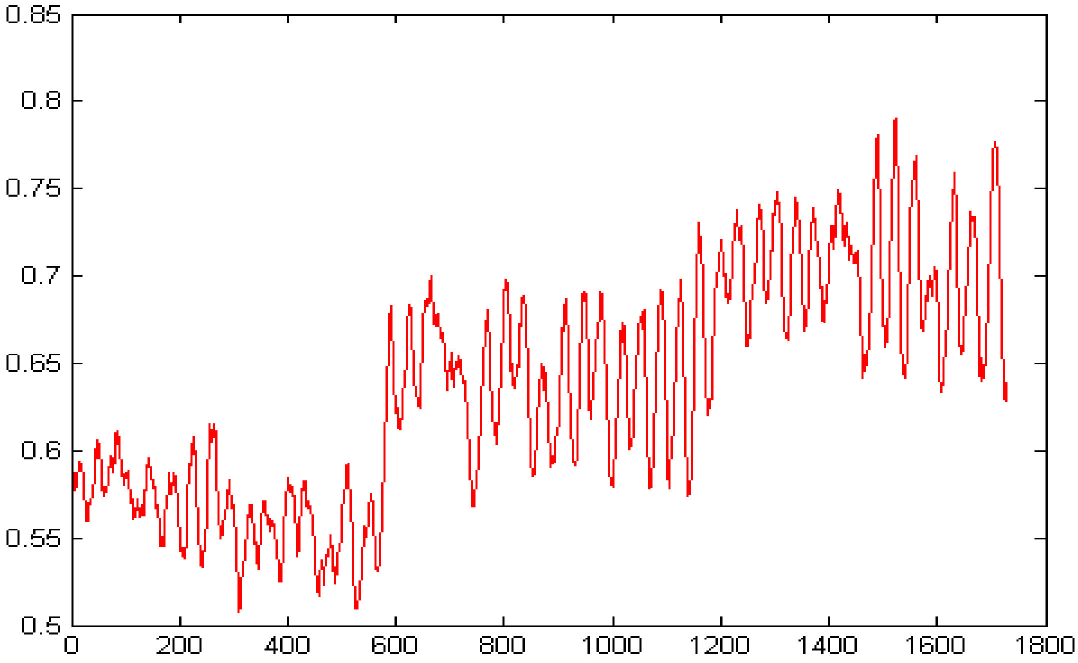
Employing cross-validation yields an additional advantage that the variance in E_{ev} can be observed for different folds. As proposed in [Kohavi 1997], I can select a model as ‘best performing’ if it maximizes accuracy and the variance in accuracy across folds is less than a preset value. Intuitively, for models with high variance a considerable number of folds would have low accuracy while a considerable number would have a high accuracy. As a result, we would end up picking a model that performs well for a majority of the data/folds. In other words, a majority of folds would have accuracy ‘close’ (decided by the preset value) to the mean. An alternative is to observe the *median* accuracy, as I did in equation 8.4, rather than the *mean* accuracy. This would dynamically ignore the few ‘outlier’ folds with high accuracies for any model.

8.2.2 High-Platform Method

Figure 8.1 titled “Finding a High-Platform” represents a plot of median accuracy for different combinations of parameters. I assume that close values of parameters create models that are conceptually and performance-wise similar. The conjecture is that picking a model that does both, maximizes the median accuracy and belongs to a ‘high-performing neighborhood’, would preclude an overfitted model that may have a high accuracy but would be adjacent to other models that do not perform well. After combinations of all parameters are tried and median-accuracies are recorded, I pick a model that

1) exists within the neighborhood that has the highest mean median-accuracy, and

2) also has the highest median-accuracy within that neighborhood.



8.1 Finding a High-Platform. Four parameters (Learning Rate, Momentum, Hidden Neurons, and Epochs) for a Neural Network are varied. X-axis: Index of each combination of parameters' values. Y-axis: Median Accuracy for each cross validation. A region, such as the one between 1200 and 1450, with the highest mean median-accuracy is chosen.

Each neighborhood is defined by a fixed combination of values for the parameters that we want to optimize. Then averaging the accuracy over a neighborhood yields the 'platform', which amounts to nullifying the effect of varying values of the remaining parameters. For instance, optimizing a total of four parameters would involve the following. After all possible combinations of the four parameters are tried:

1) Calculate the average accuracy for each unique combination of the first three parameters

2) Find and fix the combination that has the highest average accuracy,

3) Then, vary values of the fourth parameter and select the model with the highest accuracy.

To aid comprehension, this is similar to a *pseudo* "best-first" approach for parameter tuning while keeping the order of parameters the same as above. In this procedure, an exhaustive search would be performed for two parameters, then the third and fourth are chosen one at a time. It would progress as follows:

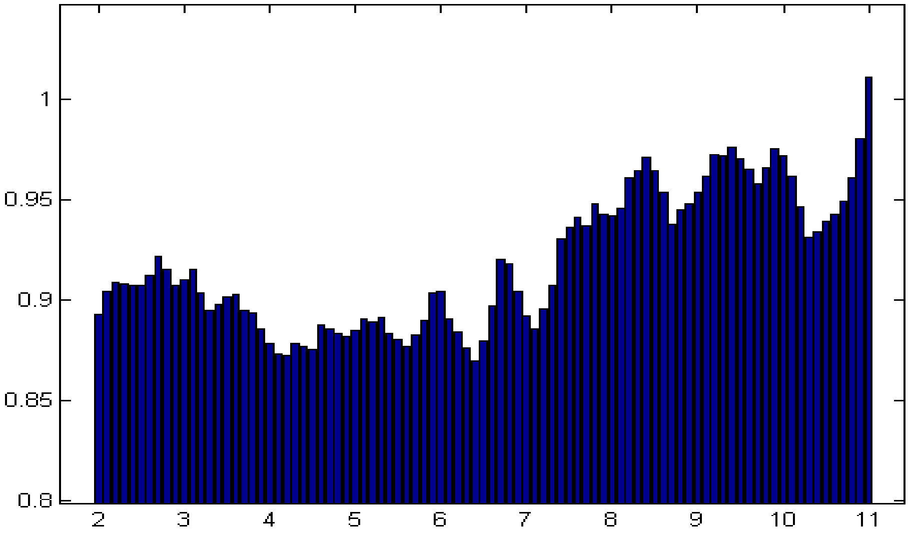
1) Try all possible combinations of the first two parameters, fix the best one, and call it Opt2 ,
2) Try all values for the third parameter (for a fixed value of the fourth parameter) and note the best one,
3) Try all values of the fourth parameter (for a fixed value of the third parameter) and note the best one,
4) From steps 2 and 3 above, pick the model with higher accuracy and call it Opt3,
5) Optimize on the remaining parameter, yielding Opt4.

In the procedure above, values for the third and fourth parameters are selected apriori during the procedure. Instead, the proposed high-platform method searches for the best model by taking one or more steps back from exhaustive search. For one-step-back, it reduces variance at both, penultimate and ultimate levels. Optimizing at the penultimate level is done by averaging variation in performance induced by varying values of the last parameter. For a greater reduction in variance, the search would take two-steps-back and average the variation in performance induced by all combinations of the remaining *two* parameters.

Maximizing Information Content with Heuristics and Statistical Tests

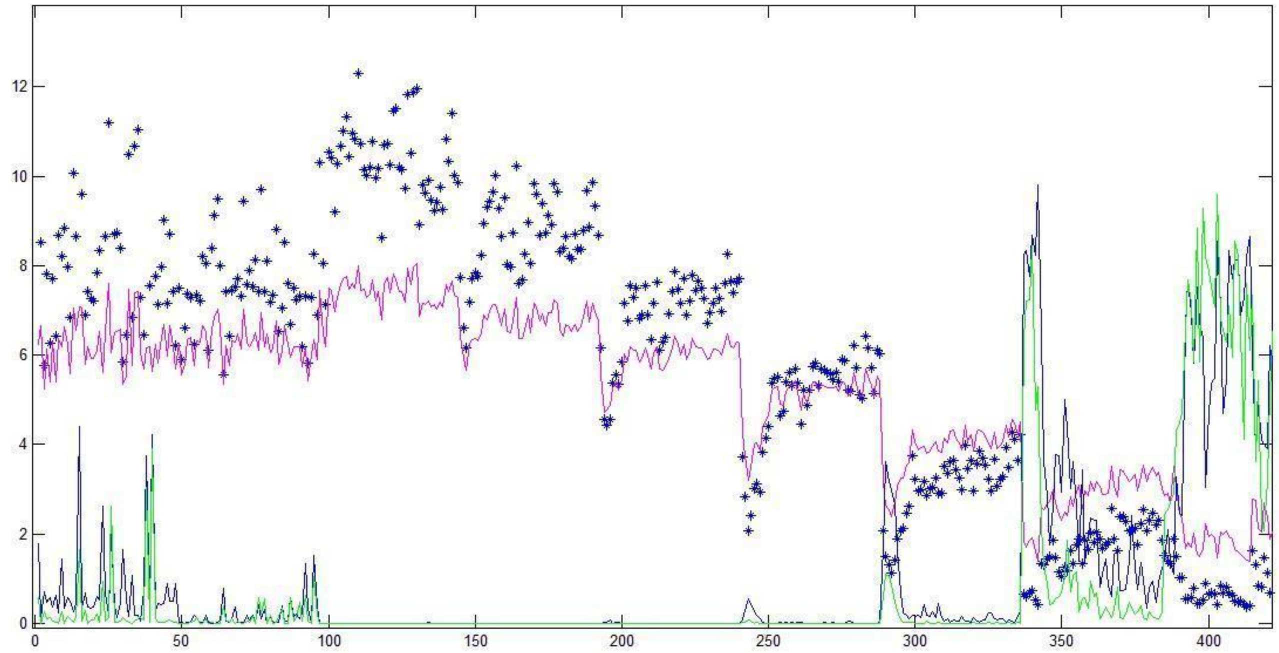
While there are myriad heuristics and parameters that form components of induction algorithm, dynamical model, filtering, and meta approach for parameter and feature selection, finding the appropriate values for all does not have to involve combinatorics. Certain parameters, like length of signal-segment, can be pre-chosen without combinatorics. A shorter duration pre-shock signal segment is desirable because it allows the model to predict outcomes sooner. However, a lack of information can be expected for windows that are ‘too short’.

Figure 8.2 titled “Information Content” represents a tradeoff between window length *l* and prediction power because we want to minimize *l*. Here, the quantity *sep* is calculated for each *l*. Based on the plot, short windows around 4 seconds in length should not contain enough information about outcomes. As such, optimization of all other model parameters at these durations would still yield sub-par results (see Chapter 8 Section titled "Results").

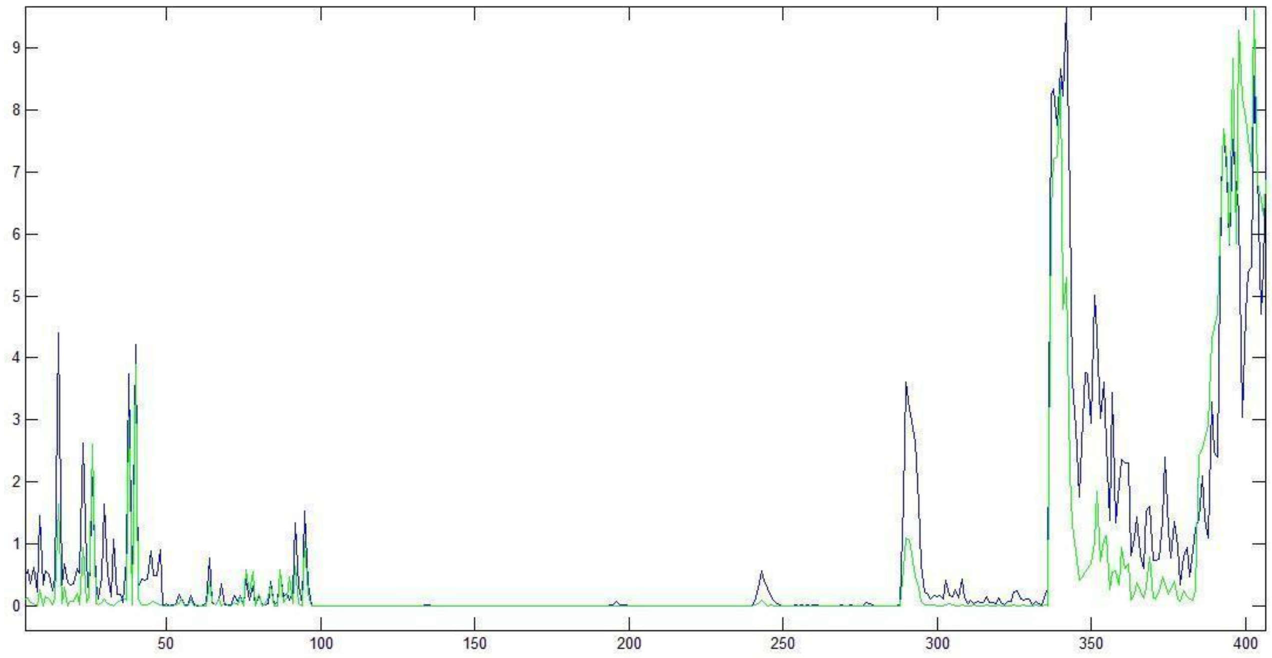


8.2 Information Content. Bar Plot of information content, measured by *sep* (y-axis), as a function of signal duration in seconds (x-axis).

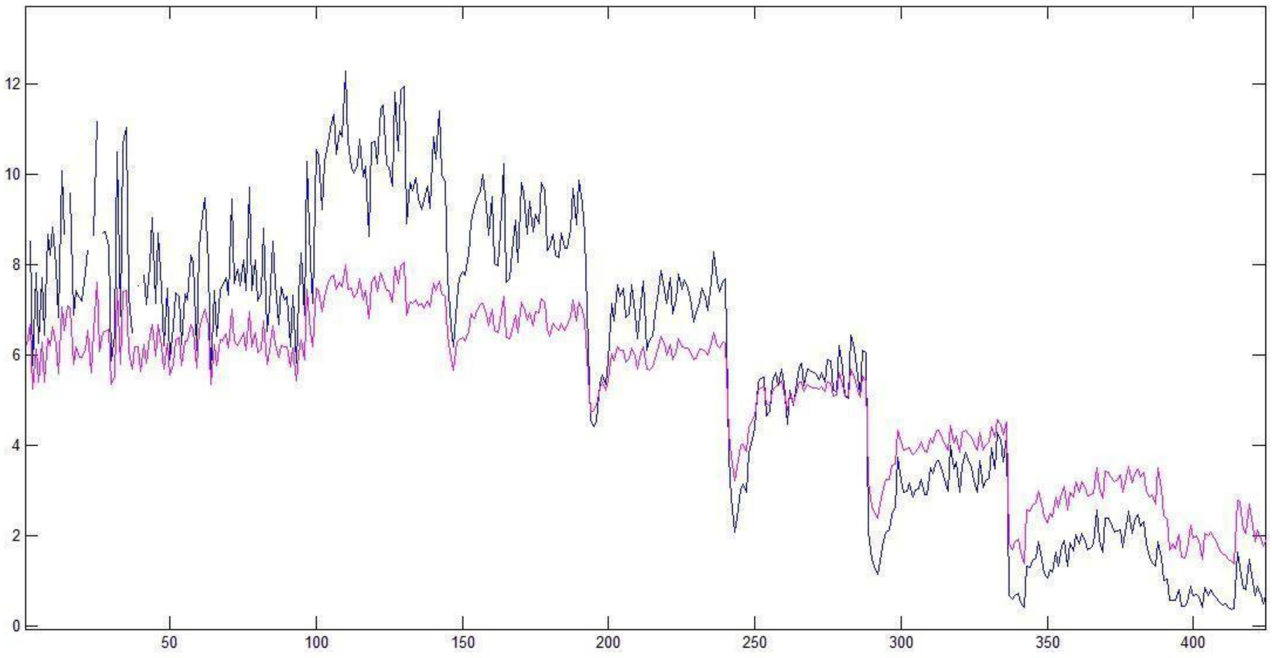
Next, the measure *sep* is compared with the *F* statistic and standard statistical tests, ANOVA and Kruskal-Wallis. For different combinations of parameter values for dynamical modeling proposed in Chapter 4, feature-sets are constructed. ANOVA and Kruskal Wallis test the significance of the feature sets generated. This helps us test the validity of the *sep* measure in light of traditional statistical tests. In figure 8.3 titled “Heuristics”, the *F* measure has been plotted in the same color as ANOVA line to reflect the fact that P(FP) for ANOVA is calculated from the value of the *F* measure and are therefore directly proportional. The probabilities were scaled up for visualization purposes.



8.3 Heuristics X-axis:Different combinations of parameter values for the QPD-PD method. Y-axis: Scaled Probability of False Positive (for Blue and Green lines) or Values of Measure (for Blue Stars and Pink Line). Blue Stars: *F* measure, Pink Line: *Sep* measure, Blue Line: ANOVA Probability of False Positive, Green Line: Kruskal-Wallis Probability of False Positive.



8.4 Traditional Hypothesis Tests. A comparison of ANOVA and Kruskal Wallis. Y-axis::scaled Probability of False Positives, X-axis: Index of Unique Combinations of Parameters



8.5 *Sep* versus *F*.Lines are plotted for the values of both *Sep* (Pink) and *F* (Dark Blue). X-axis: Index of Unique Combinations of Parameters. Both curves were scaled to have the same mean for visualization purposes. The proportional variance is higher for *Sep* as it takes on much smaller absolute values (between 0 and 1.5).

ANOVA assumes a normal distribution for a feature with respect to each class, while Kruskal-Wallis is a non-parametric equivalent of ANOVA (Figure 8.4). Kruskal-Wallis can therefore assess non-normally distributed features but does so at a loss of some information. This loss is incurred when continuous features are converted to ranks. About 20% of the features extracted showed non-normal type histograms. Still, ANOVA agreed with Kruskal-Wallis for a vast majority of the tests and seemed to have a better resolution, indicated by a larger (than Kruskal-Wallis) probability of false positives *P*(*fp*) when KW also showed an increased *P*(*fp*). KW showed a higher probability of false positives (than ANOVA) for only a few cases. *Sep* and *F measure* agree with each other for all cases (Figure 8.5), while *sep* shows a greater amount of proportional variance (variance normalized by the mean value) as compared to *F*. For combinations 0 through 100, where both *F* and *Sep* measures showed relatively large variation, certain combinations showed large P(FP), even though the values of the measures were relatively large. Therefore, increased relative variance within a neighborhood may be indicative of spuriously over fitted models.

8.3 Summary

Multiple comparisons were performed with *sep*, ANOVA, Kruskal-Wallis and the *F* measure. Comparing ANOVA and the *F* measure is trivial because *P*(*fp*) for ANOVA is calculated from the F distribution using the value of the measure. Based on the preceding plot titled "Heuristics", both measures give “statistically significant” results by parametric *and* non-parametric standards.

For model selection, a smaller number for *k,* as in *k-fold* cross-validation, is preferred over a large *k* [Zhang 1992]. The rationale for this principle is similar to the one for our method (Equation 8.4). Since a smaller *k* produces smaller training sets, the resulting individual models would vary more from each other. This is akin to bootstrapping with a smaller percentage of the sample set, which decreases the ‘overlap’ in the training sets. As such, finding a spurious combination of parameters that would create high-performing models for all (or a preset majority of) the evaluation folds would become difficult when presented with training sets containing varying information.

The methods proposed in this chapter are computationally feasible for "small to mid-sized" datasets (or “large” datasets when working with highly parallel computing structures). What qualifies as “big” data is dependent on the resources available at an organization. In contrast, the comparison of heuristics and statistical tests serves to show that the simplest solutions can be the most powerful tools when used with domain knowledge. A measure such as *sep* or *F* can elucidate a vast majority of the information present when used with the right framework for the context.

9. A Decision-Support System

9.1 More Time-Series Modeling

9.1.1 Decomposing a Signal

In addition to the methods described in previous chapters, features were also extracted by decomposing the signal into individual components. Methods described herein correspond to block C2 in figure 4.2 titled "Overview of the System".

Fourier Transform based measures [Ristagno 2008] assume a linear, deterministic basis for all signals, and prove to be impracticable for our purpose. Other methods ([Strohmenger 2008], [Watson 2004], [Neurauter 2007]), with somewhat more feasible definitions of post-shock success, have focused on extracting features based on the *real* Discrete Wavelet Transform (DWT).

For a signal expressed as a function of time, *t*, the wavelet transform is described by the following basis set:

$$\phi_{(S,l)}(x) = 2^{-S/2}\phi(2^{-S}x - l)$$

(9.1)

Here, *S* gives the wavelet's width and *l* gives its position. The 'mother function', *Φ*, is a decaying wave-like function, altered to form the basis and subject to constraints that all members of the set are orthonormal, which provide a linearly independent set of functions. In Discrete Wavelet Transform (DWT), the scaling function, defined as follows, plays a central role in forming the basis.

$$W(t) = \sum_{k=-1}^{M-2}(-1)^k c_{k+1}\phi(2t + k)$$

(9.2)

where C_k's are the wavelet coefficients, and k and M stand for time-shift and signal length, respectively. The figure 9.1 titled "Wavelet Based Decomposition of VF" [Addison 2005] displays a heatmap of the values of detail coefficients at multiple (scaling) levels (Y-axis) of decomposition for an unsuccessful countershock. With FT, all variation seen across the X-axis would have been averaged. As such, the figure is presented here to illustrate the advantage of using wavelet based decomposition. Small high-frequency spikes in the original signal are effectively discerned from the low-frequency components, which exhibit considerable amplitudes (in yellow and orange) pre-shock.

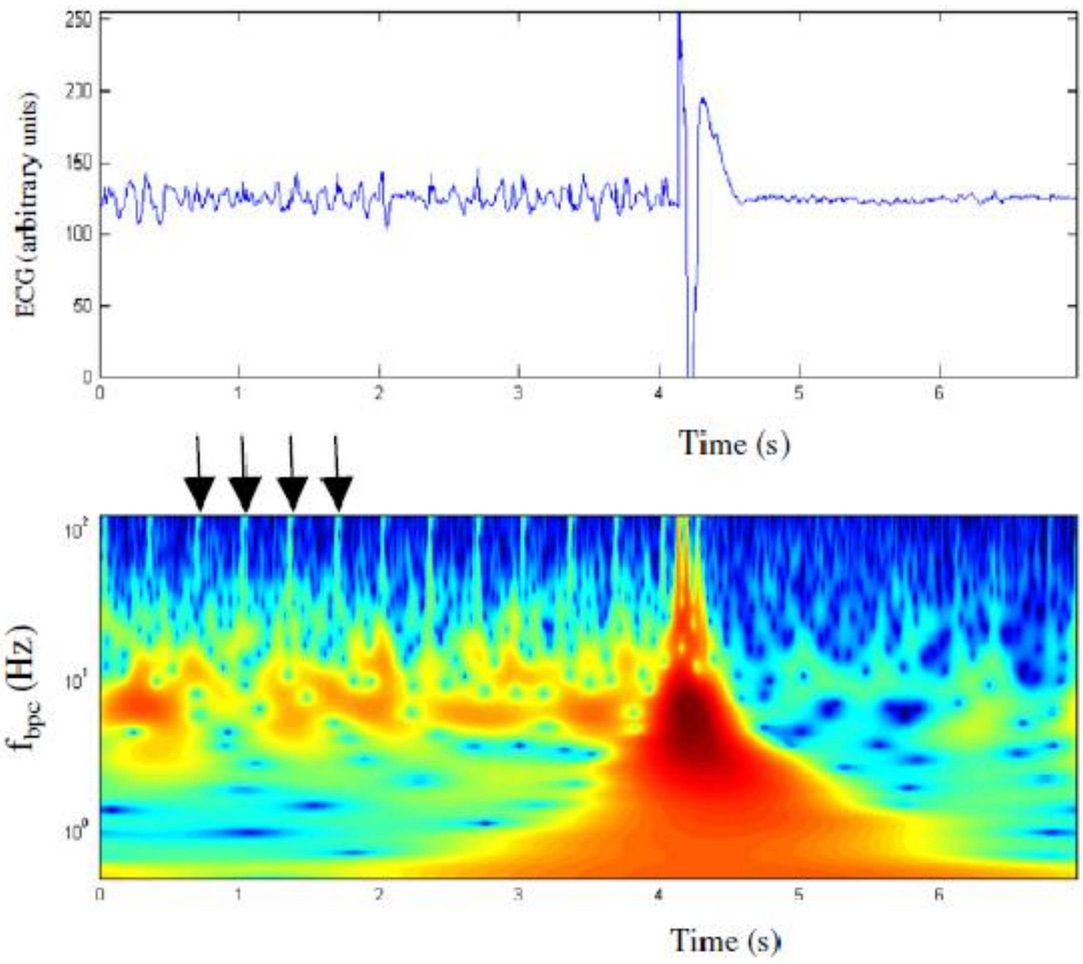


Figure 9.1. Wavelet Based Decomposition of VF f_{bpc} represents different scales at which time-series is decomposed

[Addison 2005]

Traditional DWT suffers from shift variance. During the signal decomposition, DWT shifts the signal by a small amount, which causes artificial changes in the decomposed signal represented by coefficients. Notably, multiple signal segments (one for each shock) are contributed by each subject and each short-term signal segment represents a highly dynamic system. Shift variance can yield spurious features that have false correlations with outcomes. As such, the predictive model generalizes poorly, or put another way, is not discriminative. Complex Wavelet decomposition, under certain conditions, can be approximately shift-invariant without a considerable increase in computational complexity for low-dimensional signals; for our case, one-dimensional. Here, the mother function and scaling function, both have a real as well as a complex component.

$$\phi_C(t) = \phi_r(t) + j\phi_i(t)$$

(9.3)

Specifically, when Φ_r and Φ_i are Hilbert transform pairs, the decomposition coefficients approach the desired shift-invariant property. This version of Complex Wavelet Transform was implemented using a 'dual-tree' decomposition as previously proposed [16,Box 2008]. Multiple attributes were then derived from the resulting coefficients at each level of decomposition, including mean, median, standard deviation, energy and entropy. Entropy was calculated as follows.

$$E = -\sum_{i=1}^{V} C_i \cdot \log(C_i)$$

(9.4)

Here, V is the total number of discrete values that the signal takes, and C is the number of times the signal takes a particular value *i*.

9.1.2 Auto-Recursive Piecewise-Linear Model

The signal was also modeled through auto-recursive piecewise-linear modeling. The auto-recursive and piecewise nature of this technique makes it similar to nonlinear dynamical modeling. In contrast, the end-goal is to quantify the predictability of the signal based on its recent values rather than decomposing it into quasi-periods through time-delay embedding. The model is given by

$$sig(t)+c_1sig(t-1)+...+c_nsig(t-n) = z_1x(t-del)+...+z_mx(t-del-m+1) + e(t)$$

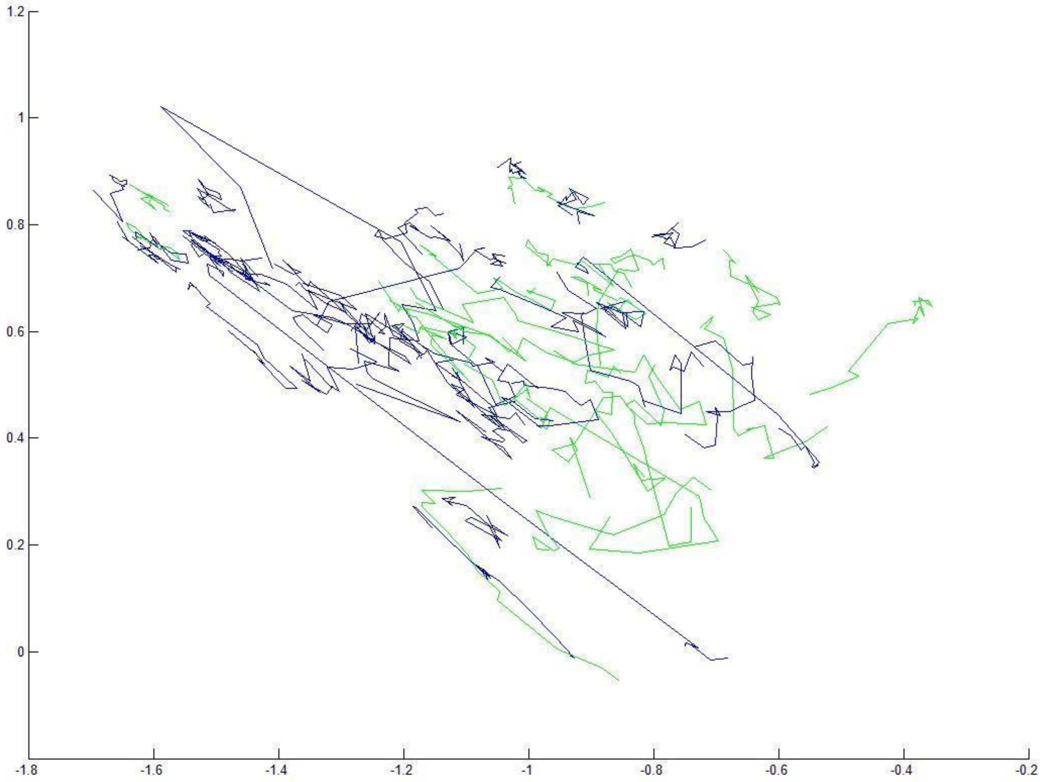
(9.5)

where *sig* represent the signal and *sig(t)* is the output at time *t*. *sig(t-1)* to *sig(t-n)* represent previous signal values on which the current value depends. *x* is the external input that affects the system and *del* is the delay after which *x* start to take effect. *c* represent poles and *z* represent zeroes. *e* represents white noise. Since left hand side of equation 9.5 yields the autorecursive nature of the model, we can ignore the right hand side and model the signal as

$$sig(t) = c_1sig(t-1)+...+c_nsig(t-n)$$

(9.6)

Then, various properties of the system can be extracted through statistical measures of the coefficients *c*. For instance, the value of mean(*c*) across classes would quantify the nature of autorecursion for a given class for specific time-delays *n*. Variance of *c* would be an indirect measure of the *amount* of autorecursion for a given class. A larger variance would represent random effects due to other inputs and/or noise. Figure 9.2 titled "State Space of Auto-Recursive Model" shows a second order state-space of the system modeled in terms of two time delays (one corresponding to each axis). Each continuous line segment displays the states that the coefficients/poles represent as time progresses for a given signal/instance. Higher order characteristics such as variance, entropy are then calculated from *c*.

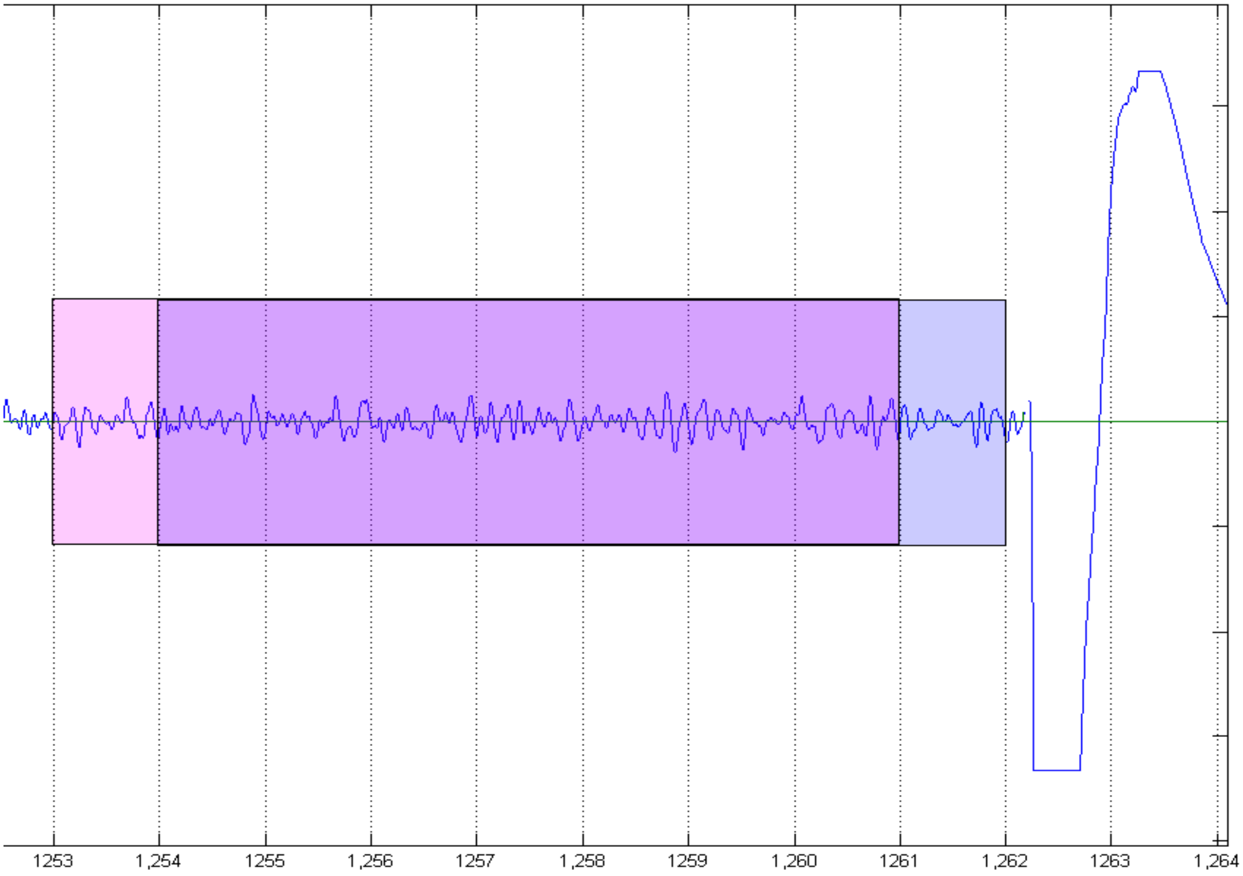


9.2 State-Space of Auto-Recursive Model (Y-axis) First Coefficient and (X-axis) Second Coefficient of a second-order model. Each coefficient corresponds to a time-delay. The classes (represented by colors) look to be separated in this phase space.

9.1.3 Second Order Time-Lapse Features

Novel measures of change in the signal over a short period of time prior to shock served to be powerful discriminative features. Each one of the methods described in this work estimate the signal with some sort of modeling. While I have extracted features (such as the features from auto-recursive modeling) that quantify how signal values vary over the length of the signal-segment, I would also like to know how a time-shift in the segment would represent a change in the overall state of the system.

Through this perspective, the entire feature set represents a state of the system, since each feature holds a single scalar value each signal segment. First, the entire feature set is calculated with an 8 second window that ends 1 second prior to shock. Then, this 8 second window is shifted by 1 second to end immediately before shock, and the feature set is recalculated. Then, one of the matrices thus obtained is subtracted from the other one. Features thus calculated can be seen as second order time-lapse features that quantify the change in state of the system. 60% of these features were statistically significant with $p < .05$ (ANOVA).



9.3 Delta State. Y-axis: Each tick represents 1mV; X-axis: Time in seconds. As the segment window shifts from pink to blue, the change in state of the system is quantified.

9.2 Comparing ML Paradigms and Algorithms

Inductive ML algorithms can induct a mathematically expressible function, as in the case of logistic regression, induct a decision tree, as in the case of C4.5 [Quinlan 1993] algorithm, or something else. I employ algorithms from different types of learning paradigms to test their performance for classification with a fixed feature set. All results are presented in table titled "ML Performance Comparisons".

Functional Induction

A backpropagation neural network was implemented with two nodes in the output layer. Parameters *learning rate*, *momentum*, and *iterations* were varied with cross validation. Best performing combination was selected.

Tree Induction

Random Forest is a well known bagging that builds multiple trees in order to reduce variance. Parameters *number of trees*, *number of features tried at each node,* and *minimum number of instances allowed at leaves* were optimized using the same procedure.

Bayesian

Bayesian Logistic Regression [Genkin 2004] with a Gaussian prior was employed. A Laplace prior can be favored when sparseness needs to be emphasized, which is not the priority here. Therefore, an assumption was made that the trained weights follow the Gaussian distribution and are most likely near 0 (distribution is assumed to have a mean of 0). The method employed for finding optimal weight values is called Maximum Aposteriori and is equivalent to Maximum

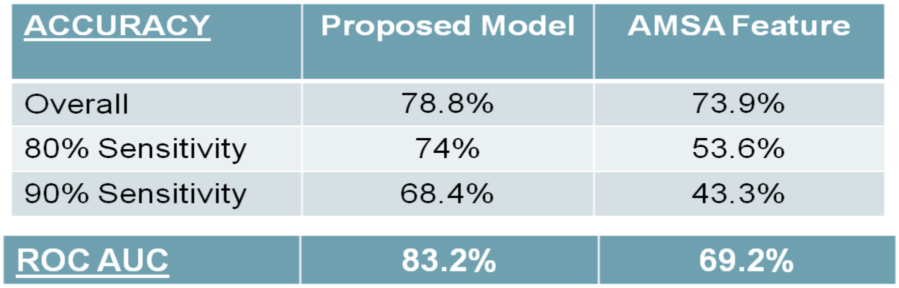
Likelihood with an apriori distribution. Parameters were then selected over a pre-specified range of .01 to 300 through cross-validation.

Boosting

While the *Iterations* parameter for a backpropagation neural network represents boosting in a fashion that is similar to the Adaboost algorithm [Friedman 1998], I tested the Adaboost algorithm with C4.5 trees. The number of *Iterations* was optimized with cross validation.

9.3 Results

ROC analysis was used to evaluate reliability of all models by calculating area under the curve (AUC). Accuracy was calculated as the average percentage, over all cross-validation runs, of instances that were correctly classified. All accuracy, sensitivity and specificity values are reported for the best decision threshold found for the given test and/or algorithm.



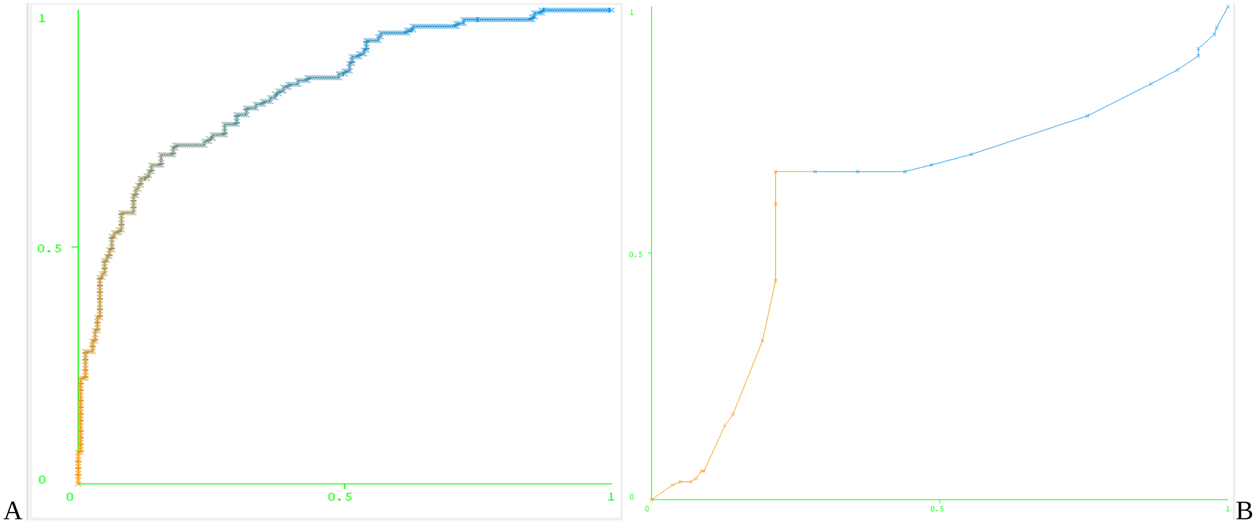
ACCURACY	Proposed Model	AMSA Feature
Overall	78.8%	73.9%
80% Sensitivity	74%	53.6%
90% Sensitivity	68.4%	43.3%
ROC AUC	83.2%	69.2%

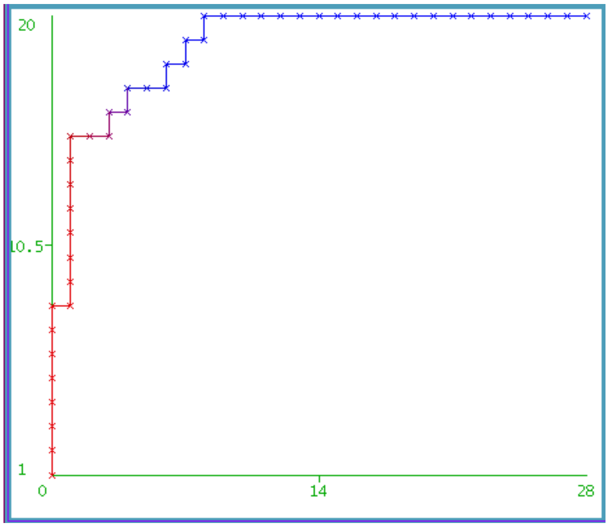
Multiple comparisons of the proposed and AMSA methods were performed using 10-fold cross validation:

- ⦿ Classification using our machine-learning approach with 16 to 20 features yielded an ROC AUC of 83.2% and accuracy of 78.8%, for the model built with ECG data only (Figure 9.4 titled "ROC Curves"). Comparison at 80% sensitivity: In this study, the two algorithms, proposed model and AMSA, were trained to provide sensitivity of 80%. In this case, our model provided an accuracy of 74% and specificity of 70.2%. For the same level of sensitivity, AMSA provided an accuracy of 53.6% and specificity of 36.7%.

- Comparison at 90% sensitivity: A similar analysis was conducted, except that both algorithms were trained to provide a sensitivity of 90%. our method provided an accuracy of 68.4% and specificity of 54.6%. For the same level of sensitivity, AMSA provided an accuracy of 43.3% and specificity of 13.3%.

Integrating PetCO2 features boosted ROC AUC and Accuracy to 93.8% and 83.3%, respectively, for a total of 48 shocks with usable CO2 segments. A large ROC AUC allowed for 90% Sensitivity and 78.6% Specificity at a classifier-output threshold value of P(Y|X) = 0.22, which represents the probability of a successful shock according to the model. Classifier (LogitBoost with Logistic Regression) output for each instance is compared to this value before it is assigned to a class. For classification problems, varying this threshold is a common way to assign more weight to one class than the other.

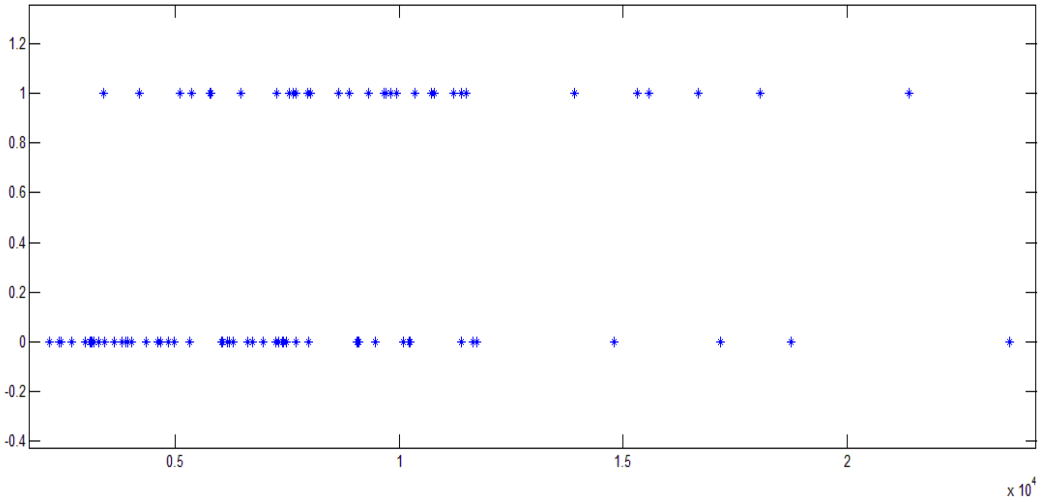




C

9.4 ROC curves. Receiver Operating Characteristic curves (A) for a model built using all 358 shocks and ECG signal only, (B) for Zoll Medical Corp's AMSA method, and (C) for a model built using 48 shocks and ECG + PetCO2. (A&B) X-axis::1-Specificity, Y-axis::Sensitivity. (C) X-axis::False Positives, Y-axis::True Positives. Threshold ranges from 0 to 1 as color transitions from one end to the other.

As only a limited number (48) of usable CO2 signals were available, these results will need to be confirmed on larger datasets. I have compared our ECG-only based method to the AMSA method [Ristagno 2008], which decomposes ECG signals with Fourier Transform. AMSA is calculated as the sum of frequencies weighted by their amplitudes. I replicated the procedure to calculate AMSA and tried to discern a threshold.



A

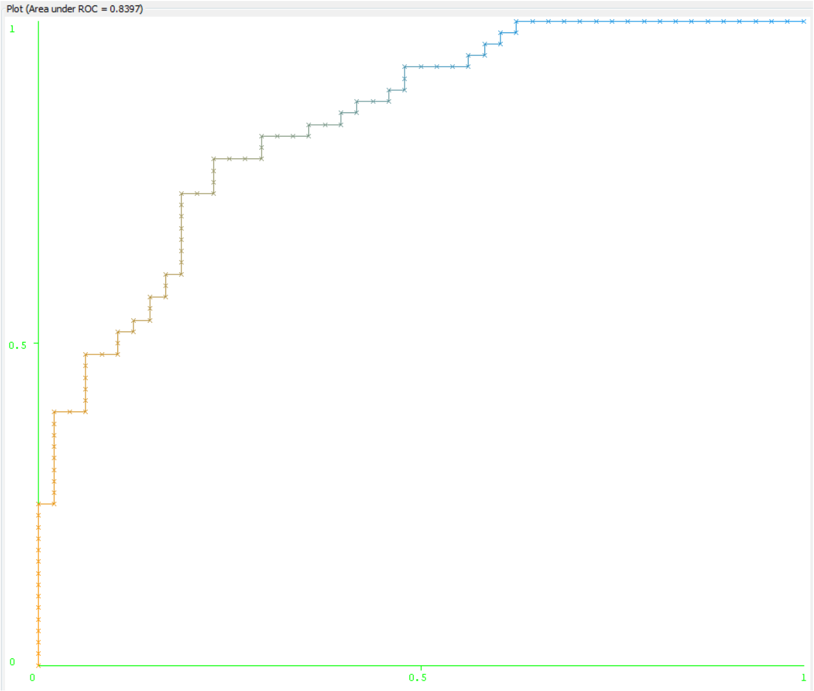
AMSA	**Mean ± Std Dev**
Successful shocks	10.2 ± 5.31
Unsuccessful shocks	6.65 ± 4.36

B

9.5 AMSA. (A) AMSA values (x-axis) for each instance/shock are plotted against classes (y-axis) ‘0’(unsuccessful) and ‘1’(successful). No clear threshold can be identified for separating the classes. (B) Means and Standard Deviations present significant overlap.

Using the methodology proposed by Ristagno and colleagues, no clear AMSA threshold could be identified (Figure 9.5 titled “AMSA”) to distinguish successful shocks from unsuccessful ones. Employing a C4.5 [Quinlan 1993] based decision stump or 1-rule for AMSA values yielded 73.9% accuracy. ROC AUC for AMSA was 69.2%. PetCO2 data was not used in the examination of AMSA.

The Re-arrest Prediction Model (RPM) was evaluated using the same methodology. 10-fold cross validation lead to selection of 8-12 features for classification of 104 signals. Accuracy was 75%, with Sensitivity of 78.6% and Specificity of 70.8%. Sensitivity equals the number of successful shocks that were correctly predicted to lead to "no re-arrest" as a proportion of total successful shocks that lead to "no re-arrest". Specificity represents the same proportion but for shocks that eventually lead to recurrence of arrest. Figure 9.6 titled "RPM's ROC Curve" displays the curve obtained by varying the decision threshold. ROC AUC for RPM was 84%, which shows that the model is robust and behaves predictably. Furthermore, a high sensitivity can be achieved if desired by the medical experts.



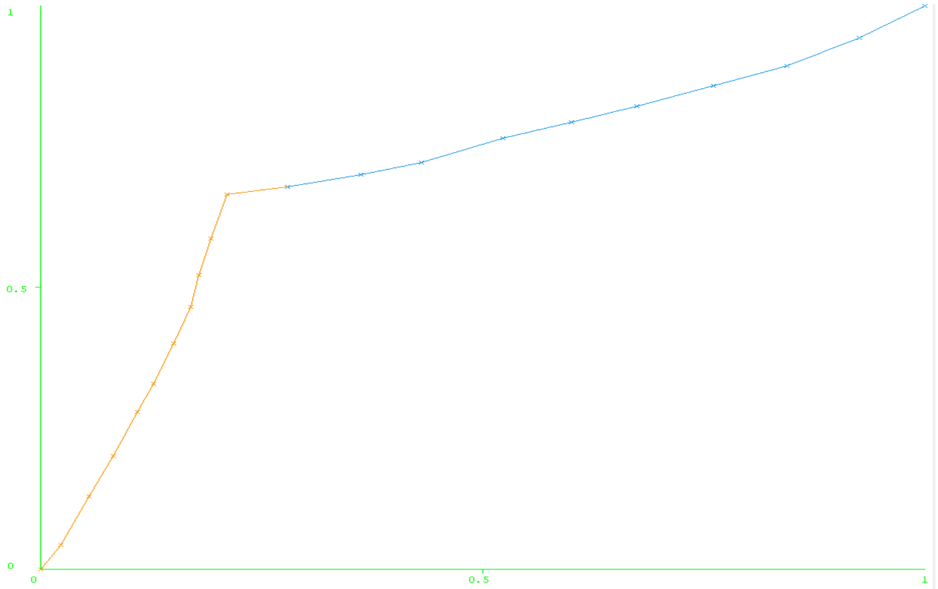
9.6 RPM's ROC Curve Y-axis::Sensitivity, X-axis::1-Specificity

Pre-shock signal length may also be optimized to provide maximum information content, and thus more discriminative features. In order to visualize how information content changes with signal duration, the signal's window size is incremented from 2 seconds to 11 seconds with 0.1 second steps. Separation along each dimension of the feature space is calculated by equation 6.5 and the mean of the top 5 most discriminating dimensions is plotted (Figure 8.2 titled "Information Content"). As a heuristic, I consider a separation of less than 0.8 (*sep* < 0.8) to be non-discriminative. The local maximum around 2.5 seconds was also tested. For this segment length, classification resulted in a much lower accuracy of 75.1%.

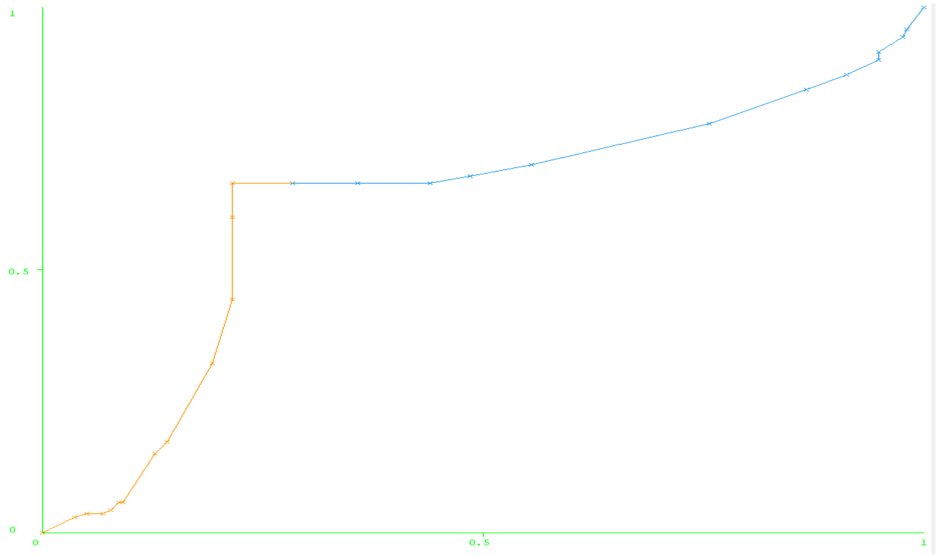
Comparing the ML algorithms that span across disparate paradigms of learning yielded the following results. All algorithms perform comparably, except for Random Forest, which performed relatively poorly. However, decision trees serve great utility with categorical data. All of the features presented were numeric.

ML Approach	Accuracy	ROC-Area	Optimized Parameter Values
Random Forest	75.1	79.9	Trees = 100, % of features = 80
Bayesian Logistic Regression	78.8	76.8	Gaussian Prior (versus Laplace Prior)
Backpropagation NN	77.4	83.7	Iterations = 500; Learning Rate = 0.3; Momentum = 0.4
Adaboost C4.5 Trees	78.2	78.4	Iterations=100

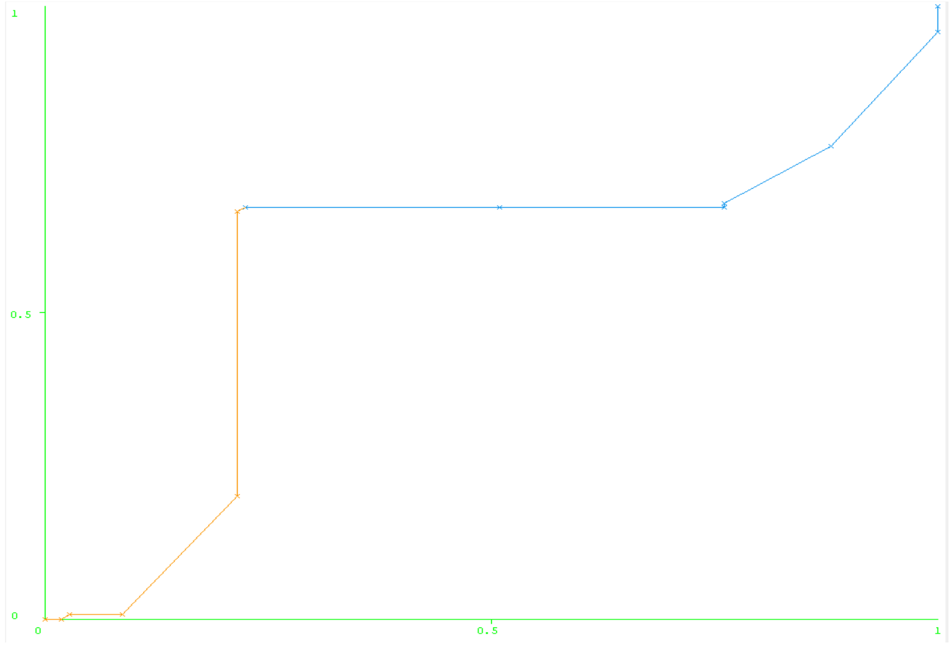
Post-hoc ROC curves (ones drawn from probability calculations at the leaves of a decision-tree) should be plotted with LOOCV (Figure 9.7 titled "ROCs for increasing *k*"). The leaves become more pure, with "purity" quantified by entropy of class labels at a leaf, as more data is presented for training. Consequently, the probability estimates (for classifier output) with leave-one-out cross-validation are more discrete and show less variation.



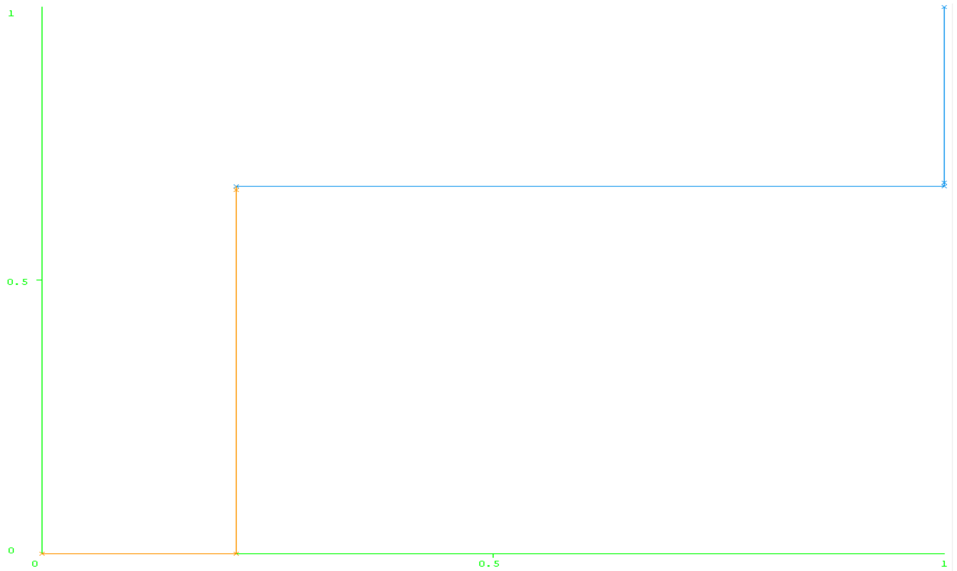
9.7A ROC with k=10 for k-fold cross validation. Y-axis: Sensitivity, X-axis: 1-Specificity



9.7B ROC with k=50 for k-fold cross validation. Y-axis: Sensitivity, X-axis: 1-Specificity



9.7C ROC with k=150 for k-fold cross validation. Y-axis: Sensitivity, X-axis: 1-Specificity



9.7D ROC with k=10 for k-fold cross validation. Y-axis: Sensitivity, X-axis: 1-Specificity

9.7 ROCs for Increasing *k*.A pattern can be observed from top to second to third to bottom curves. The k used for k-fold cross-validation is increased (up to leave-one-out cross-validation). Curves are plotted with entropy-based decision stumps on the AMSA measure.

9.4 Discussion and Conclusions

Once VF has transitioned into the mother rotor form [Zaitsev 2000], defibrillation should occur as soon as possible. Passage of time, in any pulseless rhythm, is the most significant of survival determinants [Eilevstjonn 2007, Becker 1991]. Effects of VF duration, which may or may not be countered by CPR, can be a pre-determining factor for defibrillation outcome. Many previous studies have aimed to quantify VF duration. The focus, instead, should be on improvement (of chances of ROSC) as CPR is delivered, thereby directly targeting and identifying features that are related to outcome. Such an approach will also be effective in identifying treatments that will maximize chances of ROSC.

Previous studies [Watson 2004], [Neurauter 2007], [Watson 2006] have established the advantages of a ‘wavelet’ approach over FT in evaluation of VF. However, their definitions of shock success are similar to that of Ristagno and colleagues [Ristagno 2008]. In order to overcome limitations such as the shift variance of traditional DWT, I report a first-use of

Complex Wavelet decomposition designed for defibrillation outcome prediction (and for any ECG analysis). Additionally, instead of quantifying the presumably varying degree of aperiodicity across classes through time-delay embedding [Little 2007], QPD-PD separates distributions of quasi-frequency content; thereby distinguishing two signals that differ in more ways than just perceived ‘randomness’.

Whenever cross-validation is employed with feature selection or parameter tuning, a twice-nested implementation is requisite for obtaining results that are unbiased by information in the test set. This follows from the assumption that field application will produce previously unseen data, providing a true test for the model. Additionally, there is usually a tradeoff between complexity of the predictive model and its generalization power. As complexity is partly defined by the number of features and values of the machine learning algorithm parameters, nested cross-validation also provides a way to optimize this tradeoff. For small or non-homogenous datasets, further reduction in variance is necessary.

The discussions in chapters 1 and 4, about sources of *bias*, result in two recommendations. In order to counter bias, 1) the ML algorithm should be carefully chosen based on the properties of the data and its performance should be compared to the performance of other appropriate algorithms on the same data. 2) The more the data, the more robust and high performing the model will be. This second recommendation is already common knowledge for computational scientists.

Furthermore, results on consistency (or lack thereof) of classifiers may not translate to implications for real world application due to the “No Free Lunch Theorem” in statistical learning [Wolpert 1996]. Given a finite amount of data, there is no guarantee which induction algorithm will perform better. As long as the capabilities (for instance, handling of numeric as well as categorical data) and assumptions of an induction algorithm fall in-line with the properties of the data, it may be a candidate for best performer.

While the number of subjects with usable PetCO2 values was small, the addition of PetCO2 to the algorithm appears to significantly improve performance. This is not surprising given the positive correlation between PetCO2, cardiac output, and coronary perfusion pressure produced during CPR [Ward 1998a], [Ward 1998b].

Conclusions

In this work, I propose new methods of feature extraction, feature selection, data characterization/modeling, and measures for parametric discrimination and feature calculation. These methods have implications for data mining, machine learning, and information theory; all considered to be either sub-fields of or inseparably intertwined with computer science.

I have developed a novel decision-support system in order to guide intervention during cardiac arrest. The models are built upon knowledge extracted with signal-processing, non-linear dynamical and machine-learning methods. The proposed ECG characterization, combined with information extracted from PetCO2 signals, shows viability for decision-assistance in clinical settings. The approach, which has focused on integration of multiple features through machine learning techniques, suits well to inclusion of multiple physiologic signals.

For a given desired sensitivity, the proposed model provides a significantly higher accuracy and specificity. Notably, within the range of 80-90% of sensitivity, the method provides about 40% higher specificity. This means that when trained to have the same level of sensitivity, the model will yield far fewer false positives (unnecessary shocks).

Also introduced is a new model that predicts recurrence of arrest after a successful countershock is delivered. To date, no other work has sought to build such a model. I validate the method by reporting multiple performance metrics calculated on (blind) test sets.

Based on the results obtained, I can also draw confidence in our hypothesis that random effects, as proved by Gundersen and colleagues [25,Gundersen 2008], can be countered by inclusion of multiple physiological signals. Concurrent analysis of additional physiologic signals during CPR when combined with our VF waveform analysis technique will lead to the ability to offer decision-assistance and guidance to those resuscitating a victim of cardiac arrest. Such strategies will enhance survival from cardiac arrest. Success of an integrative, information-theoretic approach should bode well for the field of defibrillation outcome prediction, which suffers from low specificities. Moreover, crucial steps are being taken for application of the system in the field as a life-saving technology.

10. Future Work

- A model should be built to predict three classes: 'Successful with No Rearrest', 'Successful with Rearrest', and 'Unsuccessful'. Such a model would represent a combination of the two models proposed here.
- Time-sensitive labeling of signals would allow for training a model, based on the same features, that would be able to predict the post-shock window of time during which ROSC would sustain. Such post-shock windows can be preset to 15sec, 30sec, 45sec and so on.

Bibliography

1 Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* 1965; 90: 297–301.

2 Akansu AN, Haddad RA. *Multiresolution Signal Decomposition: Transforms, Subbands, Wavelets*. San Diego Academic Press. 1992.

3 Mordecai A. *Nonlinear Programming: Analysis and Methods*. Dover Publishing, 2003.

4 Scheffé H. *The Analysis of Variance*. New York Wiley, 1959.

5 Kruskal WH, Wallis WA. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association* 1952; 260: 583–621

6 Lloyd-Jones D et al. American Heart Association Statistics Committee and Stroke Statistics Subcommittee. Heart disease and stroke statistics 2010 update: a report from the American Heart Association. Circulation 2010; 121: 46–215.

7 Nichol G, Thomas E, Callaway CW, et al. Regional variation in out-of-hospital cardiac arrest incidence and outcome. J Am Med Assoc 2008; 300:1423–1431.

8 Nadkarni VM, GL Larkin, MA Peberdy, SM Carey, W Kaye, ME Mancini, G Nichol, T Lane-Truitt, J Potts, JP Ornato, RA Berg. First documented rhythm and clinical outcome from in-hospital cardiac arrest among children and adults. JAMA. 2006;295:50–57.

9 Valenzuela TD, Roe DJ, Cretin S, Spaite DW, Larsen MP. Estimating effectiveness of cardiac arrest interventions: a logistic regression survival model. Circulation. 1997; 96: 3308–3313.

10 Weisfeldt ML, Becker LB. Resuscitation after cardiac arrest: a 3-phase time-sensitive model. JAMA 2002; 288 (23)3008-13.

11 Strohmenger H, “Predicting Defibrillation Success”, Cardiopulmonary Resuscitation, 2008; 14:311-316.

12 Zaitsev AV et al. “Distribution of excitation frequencies on the epicardial and endocardial surfaces of fibrillating ventricular wall of the sheep heart”, Circ Res., 2000; 86:408–417.

13 Weiss JN, Z. Qu, P.S. Chen, S.F. Lin, H.S. Karagueuzian, H. Hayashi, A. Garfinkel, and A. Karma, “The Dynamics of Cardiac Fibrillation”, Circulation, 2005; 112:1232--1240.

14 Eilevstjonn J, J. Kramer-Johansen, K. Sunde, “Shock outcome is related to prior rhythm and duration of ventricular fibrillation”, Resuscitation, 2007, 75: 60–6.

15 Ristagno G, Gullo A, Berlot G, Lucangelo U, Geheb F, Bisera J. “Prediction of successful defibrillation in human victims of out-of-hospital cardiac arrest: a retrospective electrocardiographic analysis. Anaesth Intensive Care 2008; 36: 46-50

16 Watson JN, Uchaipichat N, Addison PS, Clegg GR, Robertson CE, Eftestol T, Steen PA. Improved prediction of defibrillation success for out-of-hospital VF cardiac arrest using wavelet transform methods. Resuscitation 63: 269–275, 2004.

17 Neurauter A, T Eftestøl, H-U Strohmenger. “Prediction of countershock success using single features from multiple ventricular fibrillation frequency bands and feature combinations using neural networks”. Resuscitation 73, 253-263, 2007.

18 Berg et al. Part 5: Adult Basic Life support: 2010 AHA guidleines for Cardiopulmonary Resuscitation and Emergency Cardiovascular Care. Circulation 2010;122;S685-S705.

19 Shandilya S, MC Kurz, KR Ward, K Najarian. Predicting defibrillation success with a multiple-domain model using machine learning. IEEE Complex Medical Engineering. 2011, 22-25

20 Kingsbury NG, “The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement,” in Proc. European Signal Processing Conf., Rhodes, 1998, 319–322.

21 Box MS et al., “Shock outcome prediction before and after CPR: A comparative study of manual and automated active compression-decompression CPR”. Resuscitation 2008; 78:265–274

22 Kantz H, T Schreiber: Nonlinear Time Series Analysis. new edition Cambridge; New York: Cambridge University Press; 1999.

23 Kohavi R and G John, “Wrappers for feature subset selection”, Artificial Intelligence, Vol. 97, pp. 273-324, 1997.

24 Quinlan R: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA. 1993.

25 Becker LB, MP Ostrander, J Barrett, GT Kindus, “Outcome of CPR in a large metropolitan area—where are the survivors?”, Ann Emerg Med., 1991; 20: 355-361.

26 Watson JN, Addison PS, Clegg GR, Steen PA, Robertson CE. “Practical issues in the evaluation of methods for the prediction of shock outcome success in out-of-hospital cardiac arrest patients”. Resuscitation. 2006; 68(1):51-9.

27 Little MA, PE McSharry, SJ Roberts, DA Costello, and IM Moroz, "Exploiting Nonlinear recurrence and Fractal scaling properties for voice disorder detection," Biomedical Engineering Online, vol. 6, 2007.

28 Savitzky A, M. J. E. Golay, “Smoothing and Differentiation of Data by Simplified Least Squares Procedures”, Anal. Chem., July 1964, 36 (8):1627–1639.

29 Guyon I, Weston J, Barnhill S, and Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning, 2002;46:389–422.

30 Gundersen K et al. Identifying approaches to improve the accuracy of shock outcome prediction for out-of-hospital cardiac arrest. Resuscitation. Volume 76, Issue 2, February 2008, Pages 279–284

31 Ward KR, Yealy DM: End-tidal carbon dioxide monitoring in emergency medicine: Basic principles. Acad Emerg Med 1998a; 5:628-636.

32 Ward KR, Yealy DM: End-tidal carbon dioxide monitoring in emergency medicine: Clinical applications. Acad Emerg Med 1998b; 5:637-646.

33 Mitchell, T. (1997). *Machine Learning*, McGraw Hill. p.2.

34 Zhang, P. , "On the distributional properties of model selection criteria", Journal of the American Statistical Association (1992b) 87(419), 732-737

35 Vapnik V. and A. Chervonenkis. "On the uniform convergence of relative frequencies of events to their probabilities." *Theory of Probability and its Applications*, 16(2):264–280, 1971.

36 Geman, S. E. Bienenstock, and R. Doursat . Neural networks and the bias/variance dilemma. Neural Computation 1992 4, 1–58.

37 Kennel MB, Brown R and Abarbanel H D I Determining embedding dimension for phase space reconstruction using a geometrical construction Phys. Rev. A 1992 45 3403–11

38 Vellekoop M, Berglund, R. "On Intervals, Transitivity = Chaos". *The American Mathematical Monthly* (April 1994) 101 (4): 353–5.

39 Takens F Detecting Strange Attractors in Turbulence (Lecture Notes in Mathematics vol 898) p 366 1981.

40 Sauer T, Yorke J A and Casdagli M. Embedology J. Stat. Phys 65 579–616 1991.

41 Duda RO, Hart PE, Stork DG. *Pattern Classification*. Wiley, p 114, 2001.

42 Rumelhart, David E.; Hinton, Geoffrey E., Williams, Ronald J.. "Learning representations by back-propagating errors". *Nature* 323 (6088): 533–536 (8 October 1986).

43 Arthur Earl Bryson, Yu-Chi Ho . *Applied optimal control: optimization, estimation, and control*. Blaisdell Publishing Company or Xerox College Publishing. pp. 481(1969)

44 Breiman, Leo; Friedman, J. H., Olshen, R. A., & Stone, C. J. . *Classification and regression trees*. Monterey, CA: Wadsworth & Brooks/Cole Advanced Books & Software (1984).

45 Mitchell T, *Machine Learning*, 1997, pp. 175

46 Wolpert, D. , "The Lack of A Priori Distinctions between Learning Algorithms", *Neural Computation*, pp. (1996) 1341-1390

47 Geman,S. E Bienenstock, and R. Doursat . Neural networks and the bias/variance dilemma. Neural Computation 4, 1–58 (1992).

48 Joar Eilevstjønn, Jo Kramer-Johansen, Kjetil Sunde. Shock outcome is related to prior rhythm and duration of ventricular fibrillation *Resuscitation* October 2007. vol 75 issue 1 Pages 60-67

49 Nowak R, Baranuik R. Wavelet-Based Transformations for Nonlinear Signal Processing. IEEE Transactions on Signal Processing. 1998, 12-13.

50 Freund Y, Schapire, RE. *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*. 1995.

51 Friedman J, Hastie T and Tibshirani R. Additive logistic regression: a statistical view of boosting. Annals of Statistics 28(2), 2000. 337-407.

52 Breiman, L. "Bagging predictors". *Machine Learning* 24 (2): 123–140,1996.

53 Breiman, L. "Random Forests". *Machine Learning* 45 (1): 5–32, 2001.

54 Gama J. “Functional Trees”. Machine Learning. Volume 55 Issue 3, 219 - 250, 2004.

55 Addison PS. Wavelet Transforms and the ECG: A Review. *Physiol. Meas.* Issue 26**,** 155–199, 2005.

56 Genkin A, Lewis DD, Madigan D. Large-scale bayesian logistic regression for text categorization. 2004.

List of Figures

About the Author

Sharad Shandilya has a Bachelor of Science in Biomedical Engineering and a Master of Science in Computer Science, both from Virginia Commonwealth University in Richmond VA.

He received the Provost scholarship for his undergraduate education and is a recipient of the Who's Who award for graduate students. He served in a statistical research position, as a *Research Specialist*, at VCU while pursuing his PhD. Sharad has also built new decision support systems and algorithms in the financial and internet advertising verticals. Following is a list of Sharad's publications to date.

- **S Shandilya,** X Qi, K Ward, M Kurz, K Najarian. Finding an Optimal Model for Prediction of Shock Outcomes through Machine Learning. *To be published*. July 2013.
- **S Shandilya**, KR Ward, M Kurz, K Najarian. Integrating Physiologic Signals with Machine Learning for Predicting Defibrillation Success. *Circulation*, A182, November 2012.
- **S Shandilya**, KR Ward, M Kurz, K Najarian. Non-Linear Dynamical Time-Series Characterization for Prediction of Defibrillation Success through Machine Learning. *BMC Informatics and Decision Making.* 12:116, 2012.
- **S Shandilya**, KR Ward, M Kurz, K Najarian. Comparing a Novel Stochastic Integrative Machine Learning Model with 'AMSA' for Predicting Defibrillation Success. *Circulation*, A304, November 2012.
- **S Shandilya**, KR Ward, M Kurz, K Najarian, Predicting Defibrillation Success with a Multiple-Domain Model Using Machine Learning, *IEEE Complex Medical Engineering*, May 2011. (chosen for oral presentation)
- **S Shandilya**, KR Ward, K Najarian, A Time-Series Approach for Shock Outcome Prediction Using Machine Learning, *IEEE BIBM*, December 2010. (selected for full paper-presentation)
- **S Shandilya**, SY Ji, K Ward, K Najarian, Prediction of Shock Outcome Using Signal Processing and Machine Learning, *Circulation*, November 2010.
- SY Ji, AA Bsoul, **S Shandilya**, R Hakimzadeh, KR Ward, and K Najarian, Monitoring Severity of Hemorrhage by Integrating Knowledge from Multiple- Physiological Signals Using Wavelet Transform Analysis, *BIOTECHNO,* March 2010.
- X Qi, A Belle, **S Shandilya**, W Chen, C Cockrell, Y Tang, KR Ward, RH Hargraves, K Najarian. Ideal Midline Detection using Automated Processing of Brain CT image. *Open Journal of Medical Imaging*, 2013.

- X Qi, A Belle, **S Shandilya**, RH Hargraves, C Cockrell, Y Tang, KR Ward, K Najarian. Actual Brain Midline Detection using Level Set Segmentation and Window Selection. *The Eighth International Multi-Conference on Computing in the Global Information Technology*, 2013.

- X Qi, **S Shandilya**, A Belle, RH Hargraves, C Cockrell, Y Tang, KR Ward, K Najarian. Automated Analysis of CT Slices for Detection of Ideal Midline from Brain CT Scans. *The Eighth International Multi-Conference on Computing in the Global Information Technology*, 2013.

- X Qi, W Chen, A Belle, **S Shandilya**, RH Hargraves, C Cockrell, K Najarian. Automated intracranial pressure prediction using multiple features sources. *IEEE International Conference on Information Science and Applications*, 2013.