2011-12-15

A General Model for Continuous Noninvasive Pulmonary Artery Pressure Estimation

Robert Anthony Smith
Brigham Young University - Provo

A General Model for Continuous Noninvasive Pulmonary Artery Pressure Estimation

Robert Smith

A thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Master of Science

Dan Ventura, Chair
Christophe Giraud-Carrier
Eric Mercer

Department of Computer Science

Brigham Young University

April 2012

ABSTRACT

A General Model for Continuous Noninvasive Pulmonary Artery Pressure Estimation

Robert Smith
Department of Computer Science, BYU
Master of Science

Elevated pulmonary artery pressure (PAP) is a significant healthcare risk. Continuous monitoring for patients with elevated PAP is crucial for effective treatment, yet the most accurate method is invasive and expensive, and cannot be performed repeatedly. Noninvasive methods exist but are inaccurate, expensive, and cannot be used for continuous monitoring. We present a machine learning model based on heart sounds that estimates pulmonary artery pressure with enough accuracy to exclude an invasive diagnostic operation, allowing for consistent monitoring of heart condition in suspect patients without the cost and risk of invasive monitoring. We conduct a greedy search through 38 possible features using a 109-patient cross-validation to find the most predictive features. Our best general model has a standard estimate of error (SEE) of 8.28 mmHg, which outperforms the previous best performance in the literature on a general set of unseen patient data.

Keywords: Feature Selection, PAP, Medical Diagnostics, SVM Parameter Selection, Neural Networks, Neural Networks Topology, Dimensionality Reduction

Contents

List of Figures

List of Tables

1 INTRODUCTION

Heart disease is the third leading cause of death. Doctors diagnose some heart diseases by measuring pulmonary artery pressure (PAP). Most often, they use right heart catheterization to measure PAP [1]. During right heart catheterization, a doctor cuts an incision into the thigh and feeds a tube up through an artery until it reaches the upper section of the heart, where a sensor detects the pulmonary artery pressure. Over 200,000 patients per year in the US subject themselves to this invasive operation because there is no accurate noninvasive diagnostic for PAP. Right heart catherization's financial cost and physical risk discourage patients from seeking potentially life-saving diagnosis and treatment of elevated PAP. Furthermore, the surgical component of right heart catheterization (placement of a catheter into the heart), precludes frequent or long term PAP evaluation. Researchers are searching for an accurate PAP diagnostic to evaluate PAP frequently.

Doppler echocardiography and phonocardiography are two possible solutions to the PAP estimation problem which do not require surgery.

Doppler echocardiography measures PAP noninvasively by analyzing the speed of blood flowing through the heart [2, 3, 4, 5]. Doppler requires the presence of certain characteristics in the patients heart in order to use it. These requirements prevent a large percentage of patients from using Doppler [4], because including these patients would further degrade the results. With poorly-scoring patients removed from the results, Doppler has an average error of 30.20% compared to right heart catheterization, the ground truth [5]. Additionally, Doppler echocardiography requires a highly specialized doctor to analyze the results, which makes for delayed results and high prices.

Researchers have suggested Doppler echocardiography as a noninvasive alternative to right heart catheterization. Doppler measurement is a tricky procedure performed by specialized physicians who are well-trained in cardiac medicine and Doppler technology, including the many factors which exclude patients. A patient cannot perform Doppler in their own home. In fact, even a trained technician cannot operate the Doppler systems. In addition to the training time, the time it takes for the doctor to actually obtain and interpret the results is prohibitive. These reasons prevent it from being used frequently or continuously.

Additionally, Doppler cannot be used on 50% of patients with normal PAP, 10-20% of patients with increased PAP, and 34-76% of patients with chronic obstructive pulmonary disease [4]. The Doppler results in the literature that we found start with a random sample of patients, but cull those patients for whom Doppler would not work well prior to the experiment. We can assume that the results would be much worse if they included the unfiltered entire patient set.

Brechot *et al.* built a Doppler system that classifies if patients have elevated PAP or normal PAP. Although the medical standard for hypertension is PAP $\leq$ 25 mmHg, they used a threshold of 40 mmHg. Reinterpreting their results with the standard 25 mmHg cutoff for hypertension, 9 of the 15 patients were misclassified as healthy when in fact they were hypertensive [2].

Stephen *et al.* collected Doppler and right heart catheter data on a set of patients, then fit a function of the Doppler features to the true PAP values. To validate they built a model on training data, then tested the model on the same data, with SEE of 1.9 mmHg. This sort of validation is not predictive of clinical performance since they do not use a hold out set or cross-validation to test their model and excluded 31% of their patient base, and since several model classes, such as neural networks, can fit training data on any problem. This is the best Doppler result we can find in the literature [3].

Phonocardiography estimates PAP from noninvasively recorded heart beat sounds. A technician records patients' heart beat sounds with an external microphone. The machine learning practitioner extracts features from these sounds and selects a machine learning algorithm to generate a predictive model for pulmonary artery pressure. Unlike Doppler echocardiography, the model used in heart sound analysis provides an estimate of PAP without a doctor's input.

Phonocardiography works on more patients than Doppler. Additionally, since the computer processes the audio signals, there is no doctor in the loop. However, building a model is a serious challenge. Finding a predictive combination of an algorithm and feature set necessitates a huge search through the combinatoric space of possible features and algorithms. In past work, researchers avoided this problem by limiting the search to a few doctor-suggested features. This limitation prevents the discovery of predictive but unexpected features. The best result we are aware of in the literature is a standard estimate of error (SEE) on training data of 5.8 mmHg [4]. Previous cardiophonic systems used small patient sets, which limit the generality of the solution. The largest patient set was 23 patients [4]. To our knowledge, all cardiophonic results in the literature omit some subset of patients in order to improve results.

Phonocardiography does not exclude patients as Doppler does. The only time a patient is excluded with phonocardiography is when there is a high signal to noise ratio, or in other words, when the recording was not done properly. Xu *et al.* excluded 3 patients out of 25 due to poor signal to noise ratio [4]. Their best predictive accuracy was 5.8 mmHg with just one feature. Tranulis *et al.* obtained a SEE of 6 mmHg using a neural network and three features on a data set of 9 pigs [6]. These results are not statistically significant, however, since they mixed heartbeats from patients across the training and test sets.

Phonocardiography is a widely applicable approach to predicting PAP which does not require a doctor in the loop. But finding a predictive set of features and algorithms is a significant challenge.

Machine learning has long been used in medical applications to address such challenges [7, 8]. We use a combination of machine learning techniques in order to find a predictive feature set and algorithm combination for predicting PAP. We derive features from heart beat sounds and use machine learning to choose a specific model and features which most accurately predicts PAP values based on these features.

In our solution, we find the approximate most predictive feature subset and algorithm combination for PAP prediction. We conduct a greedy search through the possible algorithm and feature subset combinations. We use a patient set of 109 patients with 38 features to find a feature subset and algorithm combination which estimates PAP with a SEE of 8.80 mmHg (without patient exclusion). Other studies have routinely omitted patients with noisy heart beats. We have attempted to make a comparative study by omitting heart beats with missing features, and patients with less than 10 complete heartbeats, and obtain a SEE of 12.33 mmHg. The results of other models in the literature were obtained by testing and training on the same data set. This training error is essentially a measure of the fit of their model to the data they have. In other words, they do not report results on unseen data. Our training-only error fit the data with SEE of 5.57 mmHg.

As a result of our research, we contribute the following:

- Unlike all existing Doppler and most existing phonocardiographic models, our model is accurate without patient exclusion.
- Our model has the lowest SEE we could find in the literature for a cardiophonic approach when tested on new data.
- Our data set—although small in comparison to most machine learning applications—is the largest used in the literature, suggesting that our results are more general than those obtained with smaller data sets.
- Our model does not require a doctor, suggesting lower cost.
- Our model can be used as frequently as patients desire.

In this thesis, we present our model and the experiments that generated it. In this Chapter, we review previous attempts by others to solve the PAP estimation problem and show that there is still no noninvasive method which can accurately predict PAP on an unfiltered general population. In Chapter 2, we explain the model classes and parameter settings that we demonstrate are most likely to produce an accurate model. In Chapter 3 we explain the thorough search through the space of possible model classes, features, and parameters that makes us confident about the accuracy of our model on this data set. Finally, we conclude with a discussion of the experiments, results, and suggestions for future work in Chapter 4.

2 METHODS

Our problem is to find the best algorithm f, parameters $\boldsymbol{\alpha}$, and feature set χ with the lowest SEE for our data set $\boldsymbol{X}$, as measured by a leave-one-out cross-validation For a given patient p, SEE is a measure of fit, defined by

$$\text{SEE}(p) = \sqrt{\frac{\sum_{i=1}^{n}(PAP(p) - y_i)^2}{(n-2)}} \tag{2.1}$$

where n is the number of heartbeats, $PAP(p)$ is the true PAP of patient p, and y_i is the model prediction for heartbeat i. The number 2 is a normalizing term to account for a sample instead of a population. This formula is widely used in the literature as a measure of model goodness for PAP prediction.

A solution is defined by the function $f_{\boldsymbol{\alpha},\chi}$, which defines a model which predicts PAP given a data instance. Here, f is a model class generated by one of a representative subset of all machine learning algorithms, $\boldsymbol{\alpha}$ is the vector of parameters unique to that model class (such as support vectors for an SVM or edge weights for a MLP), and $\chi \subseteq H$, the set of all heart beat features. When presented an instance $\boldsymbol{x}_{p_i}$ of the set of all patient data $\boldsymbol{X}$, $y_i = f_{\boldsymbol{\alpha},\chi}(\mathbf{x}_{p_i})$ returns a prediction of the PAP of patient p for heartbeat i.

Our goal is to find the configuration (choice of $f, \boldsymbol{\alpha}, \chi$) which yields the lowest average SEE score, computed by a leave-one-out cross-validation. In other words, for each patient p, the data for all other patients $\boldsymbol{X}_{-p}$ are used to find $f_{\boldsymbol{\alpha},\chi}$. Then, the SEE for patient p is computed. This value is averaged across p for the given configuration. The best solution is

$$\underset{f,\boldsymbol{\alpha},\chi}{\operatorname{argmin}} \frac{1}{m} \sum_{p=1}^{m} |\mathrm{SEE}(p)| \tag{2.2}$$

where m is the number of patients.

The search through possible $f, \boldsymbol{\alpha}, \chi$ is non-trivial. We use a greedy approach, detailed in the next few sections.

2.1 Data Preparation

Our data set $\boldsymbol{X}$ consists of features extracted from 109 patients whose phonocardiogram was recorded from the V3 position (a specific area of the chest) while undergoing right heart catheterization (the ground truth measurement) and ECG. The mean PAP value was 26.14, the range of PAP values was (9,76) and the standard deviation was 12.16. We segment each patient's heartbeat recording into individual heart beats, from which we extract a vector of features.

We adapt the heart beat features from Dennis *et al.* as our set of features H (see Table 2.1, [9]). *Binary Features* has_{s3} and has_{s4} indicate whether the $s3$ or $s4$ sounds are detected in the heart beat signal. We calculate the *Dominant Frequency* features by taking the *argmax* over k of the k^{th} frequency of discrete Fourier transform (DFT) of the respective portion of the heart beat sound (HB, S1, S2, A2, or P2). We calculate the *Quality of Resonance* features by dividing the dominant frequency of the particular heart sound by the difference between the frequencies to the right and left of the dominant frequency at which the DFT magnitude drops to half of the maximum. We calculate the *Power* features by summing the squares of the frequencies in the DFT and normalizing by T, the length of the signal. We measure the time difference between different heart sounds in the signal with the *Splitting Interval* features. We compute several *Ratio* features to test if combined features yield more information than their separate parts. The *Systole Duration* features are a time measure between a start event, noted by the subscript, and an end event, noted by

Table 2.1: Heart Sound Features. In this table sig can be one of the following heart sound signals: HB, S1, S2, A2, or P2, where HB is the whole heart sound signal. Terms such as $t_{P2start}$ are the onset times of the indicated heart sound component. t_R is the ECG R-wave time and δ_{RR} is the time between two successive R-waves (From [9]). Dennis *et al.* did not include has_{s3} or has_{s4} because of their lack of medical indication of effect on PAP.

Category	**Features**	**Description**
Binary Features	has_{s3}, has_{s4}	Presence of S2 or S3
Dominant Frequency	F_{HB}, F_{S1}, F_{S2}, F_{A2}, F_{P2}	$\underset{k}{argmax}\ \mathcal{F}(sig)_k$
Quality of Resonance	Q_{HB}, Q_{S1}, Q_{S2}, Q_{A2}, Q_{P2}	$F_{sig}/(R_{sig} - L_{sig})$
Power	P_{HB}, P_{S1}, P_{S2} P_{A2}, P_{P2}	$\frac{1}{T}\sum_{x \in sig} \lvert x \rvert^2$
Splitting Interval	SI_{S1} SI_{S2} NSI_{S1} NSI_{S2}	$t_{T1start} - t_{M1start}$ $t_{P2start} - t_{A2start}$ $\frac{SI_{S1} \times HR}{600}$ $\frac{SI_{S2} \times HR}{600}$
Ratios	$R^{F_{P2}}_{F_{A2}}$ $R^{Q_{P2}}_{Q_{A2}}$ $R^{P_{P2}}_{P_{A2}}$ $R^{P_{A2}}_{P_{S2}}$ $R^{P_{P2}}_{P_{S2}}$ $R^{P_{A2}}_{P_{S1}}$ $R^{P_{P2}}_{P_{S1}}$ $R^{P_{S2}}_{P_{S1}}$	F_{P2}/F_{A2} Q_{P2}/Q_{A2} P_{P2}/P_{A2} P_{A2}/P_{S2} P_{P2}/P_{S2} P_{A2}/P_{S1} P_{P2}/P_{S1} P_{S2}/P_{S1}
Systole Duration	D^{A2}_{R} D^{P2}_{R} D^{A2}_{S1} D^{P2}_{S1} $\tilde{D}^{A2}_{R}$ $\tilde{D}^{P2}_{R}$ $\tilde{D}^{A2}_{S1}$ $\tilde{D}^{P2}_{S1}$	$t_{A2start} - t_R$ $t_{P2start} - t_R$ $t_{A2start} - t_{S1start}$ $t_{P2start} - t_{S1start}$ D^{A2}_{R}/δ_{HB} D^{P2}_{R}/δ_{HB} D^{A2}_{S1}/δ_{HB} D^{P2}_{S1}/δ_{HB}
Heart Rate	HR	$m/\sum_{i=1}^{m} \delta^{i}_{HB}$

the superscript. The tilde systole features are normalized by the heartbeat length to measure what percentage of the heartbeat time they consume.

Due to the variable quality and length of the heartbeat recordings, some features that required identification of specific sounds within the heartbeat, such as the dominant frequency and quality features, were unextractable on some heart beats. To treat the resulting missing values, we construct two data sets, each with a different approach:

- The first data set contains all heart beat rows of the original data set. We impute missing values while preserving the original distribution of available data by replacing the missing value with a value selected at random from the same patient with replacement, or, if all values of this feature are missing for this patient, from another patient randomly selected with replacement. This data set gives us performance results reflecting how well the system predicts PAP without patient exclusion.
- For the second data set, we remove all heartbeats with any missing data. This first step removes 90% of the original heartbeats. Because the feature selection algorithms are designed to work at the patient level, and because we assume the algorithms require at least 10 full heartbeats per patient for a valid result, we remove the 66 patients out of 109 that have less than 10 full heartbeats after missing data beats were removed, leaving 43 patients. In the end, we retain just 3,135 heart beats from the original 33,053. This data set gives us results that we compare to prior work where patients with noisy data were removed (See Table 3.1).

2.2 Feature Selection

Feature selection refers to the problem of selecting the most predictive feature subset χ from a set of features H. For this study, we limit ourselves to H adapted from Dennis *et al.* [9]. For any data set and machine learning pair there exists an optimally predictive feature subset [10]. Using too few or too many features, or the wrong features, increases error. Filter and wrapper methods are the two most common types of feature selection. Filter methods treat the feature selection process independently from the model class selection process by selecting the features independent of any knowledge of the learning algorithm that will be used. Wrapper methods search through the space of possible features given the algorithm that will be used [11]. We use a wrapper approach given that they have been shown to outperform filter methods [10].

2.2.1 Reverse Search

Given an algorithm f and algorithm parameters $\boldsymbol{\alpha}$, and data set $\boldsymbol{X}$, which feature subset $\chi \in H$ results in the lowest average SEE on a leave-one-out cross-validation? Obviously, an exhaustive search over every possible feature set in H is not feasible, as for a feature set of size n, there are $2^n - 1$ possible feature subsets, or $2^{38} - 1 = 274,877,906,943$ possibilities for just one $f_{\boldsymbol{\alpha}}$. Some practitioners simply guess f and $\boldsymbol{\alpha}$, or only test a few possibilities.

A wider-ranging search is suggested by Guyon and Bennett [12, 13]. Their recursive reverse search starts with all features and incrementally removes the feature whose removal results in the lowest error. Though more computationally expensive than a greedy forward search, recursive reverse selection captures features whose mutual information is greater than their individual information. For each iteration in the algorithm, one model must be trained and tested for each attribute remaining in the feature set. To mitigate the computational time, Guyon uses support vector weight as an iterative feature ranking and selection method. By using support vector weight as the feature rank, only one model must be constructed per iteration of the algorithm. The feature with the lowest support vector weight is removed on each iteration.

Since the ranking metric of the Guyon method is specific to linear support vector machines, it will not produce the optimal feature set for other algorithms. Because we investigate the performance of several other algorithms other than linear support vector machines, we use a standard but more expensive technique. Instead of building one model for each iteration, we build one model for each attribute in the feature set for each iteration.

Using a leave-one-out cross-validation, we calculate the average SEE resulting from removing each remaining feature for each iteration. The feature whose removal results in the lowest SEE is removed permanently at each iteration (see Algorithm 2.1). This wrapper-based method allows for a feature selection process specific to each model class.

In order to obtain the most accurate measure of how our model would perform in the clinical setting, we use a leave-one-out cross-validation—one split for each patient. We perform the cross-validation on feature set sizes from 38 down to 1, iteratively dropping the least useful feature. We use this greedy performance test to limit the number of experiments required. Our method requires just 80,769 iterations.

For each $f_{\boldsymbol{\alpha}}$, we initialize the current features χ to the full 38 attribute feature set H (See Algorithm 2.1). At each iteration, we loop through the current features, using the selected feature as a hold out feature (Line 3). For each of the features in the set, we evaluate the results of holding out the current feature as follows: We split the data into N segments, one for each patient (Line 5, keep in mind that each patient has 10-125 rows of data). We use one patient at a time as a hold-out test set. We build the test set by obtaining the data rows for the test patient and only considering the feature columns in $currentFeatures - holdoutFeature$ (Line 6). We build the training patients set in the same manner, and use them to train a model (Lines 7 and 8). We then evaluate the model on the test patient data (Line 9). We add the square of the error for the current patient to a running sum for the current held out feature. At the completion of the cross-validation for all 109 patients for the given held out feature, we calculate and store the SEE (Line 11).

Once we finish this process for each of the features in the current feature set, we remove from the list of current features the held out feature whose removal caused the lowest error, and the next iteration begins. We call Algorithm 2.1 once for every iteration of current features from size 38 down to size 1. Multiple runs are required because the goal is not a list of 38 ranked features, but a set of 38 models of size 38 down to 2 features. Each model is an approximation of the optimal model for each respective size. In other words, each algorithm

Algorithm 2.1: **Feature Select: Reverse Search**. This illustrates assigning a SEE value to one feature subset χ using reverse selection. For each feature in the current feature set *currentFeatures* (the full patient set H minus any features which have already been greedily eliminated in prior iterations of this algorithm), a leave-one-out cross-validation is executed to determine the average SEE when the $holdoutFeature$ is ignored when building the current models. Each averaged SEE becomes an entry in *seeList*. The feature to remove from $currentFeatures$ is the one which has the lowest SEE from this iteration. This algorithm is run once for every feature subset size from $|H|$ to 1. After all iterations, the most predictive subset χ is the subset of H with the lowest average SEE on a per-patient cross-validation.

```
 1: Inputs:
      $data, \quad currentFeatures, \quad patients$
 2: $seeList \leftarrow \{\}$
 3: for $holdoutFeature$ in $currentFeatures$ do
 4:   $totalSquaredError \leftarrow 0$
 5:   for $testPatient$ in $patients$ do
 6:     $testSet \leftarrow$ BUILDTEST$(data, currentFeatures - holdoutFeature, testPatient)$
 7:     $trainSet \leftarrow$ BUILDTRAIN$(data, currentFeatures - holdoutFeature, testPatient)$
 8:     $model \leftarrow$ BUILDMODEL$(trainSet)$
 9:     $error \leftarrow$ ACTUALPAP$(testPatient)$ - TEST$(testSet, model)$
10:     $totalSquaredError = totalSquaredError + error^2$
11:   $seeList[holdoutFeature] \leftarrow \sqrt{totalSquaredError/(numPatients - 2)}$
12: return $\underset{i}{\operatorname{argmin}}(seeList[i])$
```

run does not indicate which features are globally least useful, but which feature is least useful in the specific context of the features considered for the given iteration. For instance, an iteration of 25 features would output the feature that is least predictive out of those 25 features. This iteration could not suggest the globally less predictive feature selected and removed in the previous iteration of 26 features. If a feature ranking was output at each run, the features would change positions at each iteration because each iteration ranks features among the given feature subsets.

For further clarification, consider the following walkthrough of Algorithm 2.1. Suppose we have $currentFeatures = \{v_1, v_2, v_3\}$ and $patients = \{p_1, p_2, p_3, p_4\}$, and we are currently searching for an approximately optimal feature set for linear regression with data set $\mathbf{X}$, where $\mathbf{X}$ consists of heartbeat feature rows for each patient. We enter the **for** loop in Line 3. The first $holdoutFeature$ is v_1. Now we enter the **for** loop in line 5 with $testPatient = p_1$.

We construct $testSet = \mathbf{X}_{-v_1,p_1}$ (patient p_1's data from $\mathbf{X}$ with the v_1 column removed). We build $trainSet = \mathbf{X}_{-v_1,-p_1}$ (all patients from $\mathbf{X}$ except p_1 with the v_1 column removed). Now we build a model using the training data, and test it using the test data. We store the results in order to later calculate the SEE. The loop in Line 5 repeats once for every patient. At this point, we have all the data necessary to compute the SEE if we ignore feature v_1. Once we complete this loop (Line 3) for each of the features, we will know the feature to leave out to locally minimize SEE. That is, the minimum SEE for iteration i may in fact be greater than the minimum SEE from the iteration $i - 1$. This is a local greedy search, which is only concerned with the minimum SEE for the current iteration. This feature (say, v_3) will be removed from $currentFeatures$, and the algorithm will be run again with $currentFeatures = \{v_1, v_2\}$. At the conclusion of the feature selection process, we have a SEE value for each feature subset size from $|H|$ to 1 (See Appendix:Table 5.1 to see an example.) Note that this column of SEE values corresponds to one specific machine learning model class f, say linear regression. This whole process must be repeated for each algorithm class f considered.

The reverse approach to feature selection captures features that contribute more information together than if evaluated independently. We use the previously explained algorithm for feature selection with all model classes considered, except neural networks. Because of the long training time for neural networks (14 hours per model), we use a hold out set of two patients at a time until 11 features remain (one feature at a time thereafter), without a significant effect on the result.

2.2.2 Forward Search

Forward selection cannot capture features that contribute more information together than apart. However, it takes much less processing time than reverse search since forward selection cuts out features while building simple models at the beginning, leaving less features to

Algorithm 2.2: **Feature Select: Forward Search**. Given a set of *currentFeatures* that must be included in the model, forward search finds the next locally best feature (the feature that results in the lowest SEE for the current iteration) resulting in the greedy best feature set of size $|currentFeatures| + 1$. Looping through this algorithm n times will provide n feature sets of size (1, 2, $\cdots$, n) and respective SEE scores for each feature set.

```
 1: Inputs:
      $data$,   $patients$,   $currentFeatures$,   $H$
 2: $seeList \leftarrow \{\}$
 3: for $currentFeature$ in $H$ do
 4:    $totalSquaredError \leftarrow 0$
 5:    for $testPatient$ in $patients$ do
 6:       $testSet \leftarrow$ BUILDTEST($data, currentFeature \cup currentFeatures, testPatient$)
 7:       $trainSet \leftarrow$ BUILDTRAIN($data, currentFeature \cup currentFeatures, testPatient$)
 8:       $model \leftarrow$ BUILDMODEL($trainSet$)
 9:       $error \leftarrow$ ACTUALPAP($testPatient$) - TEST($testSet, model$)
10:       $totalSquaredError = totalSquaredError + error^2$
11:    $seeList[currentFeature] \leftarrow \sqrt{totalSquaredError/(numPatients - 2)}$
12: $bestLocalFeature \leftarrow \operatorname*{argmin}_i(seeList[i])$
13: $H \leftarrow H - bestLocalFeature$
14: $currentFeatures \leftarrow currentFeatures \cup bestLocalFeature$
15: return $H, currentFeatures$
```

include in each model towards the end. Given that foward search involves significantly less computation, one would hope that Reverse Search would generate more accurate models.

The algorithm is simple (see Algorithm 2.2). For each call of the algorithm, we iterate through all features in H and each patient in *patients* (Lines 3, 5). We add the feature which, in conjunction with the current H, results in the minimum SEE for this iteration (Line 12). This feature, *bestLocalFeature*, is removed from H so it is not considered in future runs, and added to *currentFeatures* (Lines 13, 14). The result of looping several iterations of this algorithm for several machine learning model classes f is given in the Appendix, Table 5.4.

2.3 Machine Learning Algorithms

We use four representative regression learning algorithms to construct models which we evaluate in search of an accurate model. Linear regression is chosen for its simplicity and pervasive use in the medical field. Support vector machines (SVMs) and multilayer perceptrons, a type of neural network (NN) are used for their power and acceptance in the machine learning domain. Two kernel variants of SVMs are used, linear and radial basis function (RBF) in order to accurately capture the learning capabilities of the model.

For each model class considered, we will explain how the machine learning algorithm finds the $\boldsymbol{\alpha}$ parameters and the method for calculating a PAP prediction once an instance of the model class is defined.

2.3.1 Linear Regression

Linear regression fits a hyperplane to data by minimizing error of fit. Let n be the number of rows in the data set and m the number of features. Linear regression finds the error-minimizing line by solving the following system of equations

$$\begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{m,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{m,2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{m,n} \end{bmatrix} \times \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = \begin{bmatrix} y_1 + \varepsilon_1 \\ y_2 + \varepsilon_2 \\ \vdots \\ y_n + \varepsilon_n \end{bmatrix}$$

where $\mathbf{y}$ is the 1-d vector of target values (PAP values for our problem), $\boldsymbol{X}$ is the n x m matrix of data vectors (heartbeat feature vectors in our problem), $\boldsymbol{\alpha}$ is the 1-d vector of coefficients and $\boldsymbol{\varepsilon}$ is vector of bias terms. Both $\boldsymbol{\alpha}$ and $\boldsymbol{\varepsilon}$ are unknown. The algorithm finds values of $\boldsymbol{\alpha}$ which minimize $\sum\limits_{n} \varepsilon^2$.

A trained instance of linear regression can be expressed by

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = \boldsymbol{\alpha}_1\mathbf{x}_1 + \cdots + \boldsymbol{\alpha}_k\mathbf{x}_k \tag{2.3}$$

where $\mathbf{x}$ is a heart beat data vector, and $f_{\boldsymbol{\alpha}}(\mathbf{x})$ is the predicted PAP value for $\mathbf{x}$. Because the solution of $f_{\boldsymbol{\alpha}}(\mathbf{X})$ for linear regression is definined uniquely by data set $\mathbf{X}$, the search for the most predictive model depends only on the feature subset search. The algorithm is simple, and therefore fast, but performs well only to the extent that a linear relationship can explain the data.

2.3.2 Support Vector Machines

A Support Vector Machine (SVM) also learns a hyperplane that, (in the regression case), best fits a given data set. The function implemented by an SVM model is given by:

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_{i=1}^{l} \boldsymbol{\alpha} \cdot k(\mathbf{x}, \mathbf{x}_i) + \alpha_0 \tag{2.4}$$

To find this model we attempt to minimize

$$\frac{1}{2}||\boldsymbol{\alpha}||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \tag{2.5}$$

subject to the following constraints

$$\left\{ \begin{array}{l} y_i - \langle \boldsymbol{\alpha}, \mathbf{x}_i \rangle - \boldsymbol{\alpha}_0 \leq \varepsilon + \xi_i^* \\ \langle \boldsymbol{\alpha}, \mathbf{x}_i \rangle + \boldsymbol{\alpha}_0 - y_i \leq \varepsilon + \xi_i \\ \xi, \xi_i^* \geq 0 \end{array} \right\} \tag{2.6}$$

In Equation 2.4, $k(\cdot, \cdot)$ is a kernel function, such as the linear or RBF kernel functions, (respectively):

$$k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y}) \tag{2.7}$$

$$k(\mathbf{x}, \mathbf{y}) = \exp(\frac{-\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}) \tag{2.8}$$

When a linear kernel (see Eq. 2.7) is used for regression, the SVM learns a regression hyperplane in the input data space. When a non-linear kernel (see Eq. 2.8) is used for regression, the kernel maps the input data non-linearly into a different (normally higher-

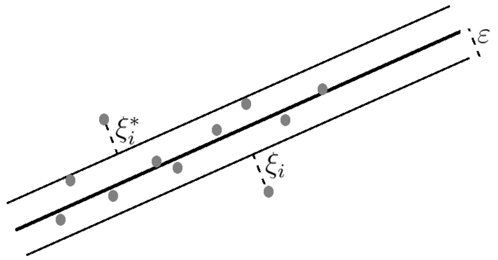

Figure 2.1: Linear SVM regression. Here each dot represents a vector of feature values mapped into the feature space. The width of the allowance for error on each side of the regression line is known as ε. The slack variables ξ control further allowance for error.

dimensional) space called the feature space. In that case, the SVM learns a regression hyperplane in the feature space.

C and ε Parameters

For some data sets a function will not fit even with the allowance of ε. For this reason, the optimization problem includes slack variables ξ_i and ξ_i^*, which allow for deviation from the fit line greater than ε. C is a free parameter in the algorithm which acts as a "knob" to control the strictness of the ε tube, adjusting for more deviation from ε on a point by point basis. The effects of C and ε are similar (as they are related to each other), though they occur for different reasons. The proper selection of C and ε are essential to obtaining optimal performance from the SVM algorithm.

The interval [ε, -ε] defines the ε tube, within which the error between the data point and f is ignored (see Figure 2.1). A support vector is a vector of data mapped into one point in the feature space, which is considered exemplary enough to be retained in the list of vectors used to define the regression hypersurface. As ε increases, more data points are ignored, resulting in less support vectors. If ε is too large, the accuracy of f will diminish. As ε decreases, more importance is given to C, as the number of ξ points increases as the width of the ε tube shrinks. If ε is too small, the performance of the system on new data will diminish as a result of overfitting.

C is the penalty term for data points that lie outside of ε but within ξ (hereafter referenced as ξ data points). As C increases the system will have more incentive to minimize the number of ξ points by adding support vectors to move f so that it minimizes the ξ distance of the ξ points. As C decreases, the model will contain fewer support vectors since ξ points are cheap and the weight magnitude minimization becomes the highest priority of the optimization. At some point the model will begin to lose accuracy as the minimal $||\boldsymbol{\alpha}||^2$ does not permit enough support vectors or high enough weights to learn the behavior of the training data.

We use the parameter selection method found in [14]: $C = 3\sigma$, where σ is the standard deviation of the target. They calculate ε by:

- Initializing ε to $0.01\mu_y$, where μ_y is the mean of the target.
- Building and testing an SVM on a small training and test set from the data.
- Update ε: $\varepsilon_{new} = (\varepsilon_{old} + \mu_{error})/2$
- Rebuild and update until ε converges.

2.3.3 Neural Networks

A multilayer perceptron, feed-forward, artificial neural network (MLP) is a structure consisting of one input node for each feature in the data, some number of hidden layers each consisting of some number of hidden nodes h, an activation function, a learning rate λ, and some number of training epochs. These parameters define a network of nodes which act as a function on the data to output a prediction value (see Figure 2.2).

The function of a two-layer network is defined by

$$f_{bm\alpha}(\mathbf{x}) = \boldsymbol{A}_2[\sigma(\boldsymbol{A}_1\mathbf{x})] \tag{2.9}$$

where $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ are matrices of weights and $\sigma(z)$ computes an element-wise sigmoid function on the components of $\mathbf{z}$:

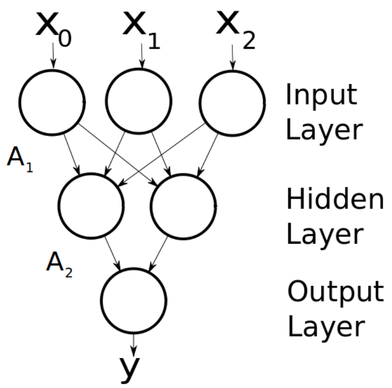

Figure 2.2: An artificial neural network. Each edge between an input node i and a hidden node j has weight $\boldsymbol{A}_{ij}$.

$$\sigma(\mathbf{z}) = \begin{bmatrix} \frac{1}{1+\exp(-z_1)} \\ \vdots \\ \frac{1}{1+\exp(-z_i)} \end{bmatrix} \tag{2.10}$$

Strictly speaking, $\boldsymbol{\alpha}$ contains not only the values of weights in $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$, but also h, the size of the hidden layer. Back propagation seeks to minimize predictive error by changing the network weights in $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$. Since gradient descent does not solve for h, we treat it as an algorithm parameter.

Hidden Nodes, Learning Rate, and Training Epoch Parameters

The parameter settings $\boldsymbol{\alpha}$ chosen for a neural network have a dramatic effect on its predictive accuracy.

Each iteration of feed-forward and backpropogation through the network is called an epoch. The number of training epochs represent a trade-off between accuracy and time to train. The more epochs taken to train, the finer tuned a NN becomes, but the longer it takes to train.

The more hidden nodes h, the more capable the net becomes of learning complex functions. More hidden nodes allow greater functional complexity, but require more time to train and tend to overfit. If more hidden nodes are used than necessary, more training time will be required to get the same results as a NN with less hidden nodes, and overfit is more likely.

The learning rate determines how much the weights can be adjusted at each epoch. The higher the learning rate, the more quickly the NN will converge on a solution, but the more likely it will be to miss the best solution. The lower the learning rate the less likely the NN will overshoot the best solution, but the more time will be required for the solution to converge to a global minima, since the adjustment rate is smaller.

We used an incremental algorithm to discover the number of hidden nodes, learning rate, and training epoch parameters for the NNs used in these experiments. The other parameters of the network were left to the Weka defaults.

- We tested the effectiveness of the following learning rates: λ =0.0005, 0.001, 0.005, 0.01, 0.05, 0.1. First, we fixed the epochs to $-500\log(\lambda)$ to allow for an exponentially increasing amount of training time for a smaller learning rate. We also set the number of hidden nodes to twice the number of features. We conducted 50 random trials of 20 train and test patients from the full data set, each using all combinations of learning rates 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1 and 5, 15, 25, and 35 randomly selected features. Training error was minimized with $\lambda = 0.05$.
- Having fixed $\lambda = 0.05$, we determined the best number of hidden nodes by trying random combinations of feature sets of 35, 25, 15, and 5 with the following numbers of hidden nodes:
  - $numberFeatures$
  - $2\log(numberFeatures)$
  - $2 * numberFeatures$
  - $3 * numberFeatures$

The number of hidden nodes with the lowest training error over 50 random trials of 20 patients on each combination was $2 \times numberFeatures$.

- Having fixed $\lambda = 0.05$ and $h = 2n$, we determined the best number of training epochs by building a learning curve from 100 to 4000 epochs in increments of 100. Each point consists of the averaged results from 20 training patients for random feature sets of length 5, 10, 15, 25, and 35. The number of hidden nodes and the learning rate were fixed at the values computed above. SEE converged at 650 epochs.

3 RESULTS

The overall computational time for our experiments, run on an internationally ranked supercomputer, would require 40 years to run on a single-node machine. Perhaps this explains why it is not usual for practitioners to test a wide variety of model classes, feature sets, and parameters when building predictive models. Our two data sets are evaluated on four machine learning model classes in a full reverse search and forward search. Each search requires the construction of 80,769 models for each algorithm and data set. The forward search requires less computational time than the reverse search since it begins evaluating and discarding features using few features, whereas the reverse search begins with the full feature set. Though these computational times may seem excessive, our methods are greedy and are significantly more restrictive than a full search, which would be computationally intractable.

3.1 Reverse Feature Search

Table 5.1 (see Appendix) shows the average per-patient cross-validation SEE of each feature set size for the four model classes considered. The lowest SEE score for the data set with patients excluded has 11 features and uses SVM with a linear kernel, with SEE 12.33 mmHg. MLP produces the lowest SEE score for the full data set (9.40 mmHg) using 18 features. The only model class to produce SEEs < 10 mmHg was MLP.

Tables 5.2 and 5.3 (see Appendix) show which features are dropped at each iteration for the large and small data sets, respectively. From top to bottom this list shows the predictive power of each feature, for each algorithm, in descending order. Some features rate well on the same model class for both data sets. For instance, P_{S2} rated most predictive on the full patient data set and second most predictive on the culled patient data set for MLP. Medical practitioners may be surprised to note that the same feature could rank very differently between two data sets even if both data sets represented the same physiological problem. Although variance in inter-data set feature performance variance is well known in machine learning, the degree of inter-data set variance is noteworthy. One example is HR in linear regression, which ranked 6th on the full data set and 31st on the culled data set. Of more significance is the intra-data set variance. One such example for the culled data set is the feature F_{HB}, which ranks as the least predictive feature for linear regression, yet the most predictive feature for MLP.

The culled data set feature rankings agree for the most part that $R_{P_{S1}}^{P_{S2}}$ is the most predictive feature. There is less agreement amongst the model classes for the full patient data set, which is no surprise granted the significantly increased number of patients and number of heartbeats per patient in the full data set. A more complicated data set may pose a greater opportunity for the computational differences in the learning algorithms to manifest themselves in the form of greater variance in results, and therefore a greater variance in the feature ranking.

3.2 Forward Feature Search

Table 5.4 (see Appendix) shows the average per-patient cross-validation SEE of each feature set size for the four model classes considered. The lowest SEE score for the data set with patients excluded has 19 features and uses MLP, with SEE 10.28 mmHg. MLP also produces the lowest SEE score for the full data set (8.28 mmHg) using 8 features. The only model class to produce SEEs < 10 mmHg was MLP, with the exception of one model—SVM-Linear with 1 feature. It is interesting to note that the 8.28 mmHg SVM-Linear model's only feature is has_{s3}, a binary feature that was deemed so medically insignificant that Dennis *et al.* didn't even include it in their classification experiments (see [9]). This result highlights the fact that

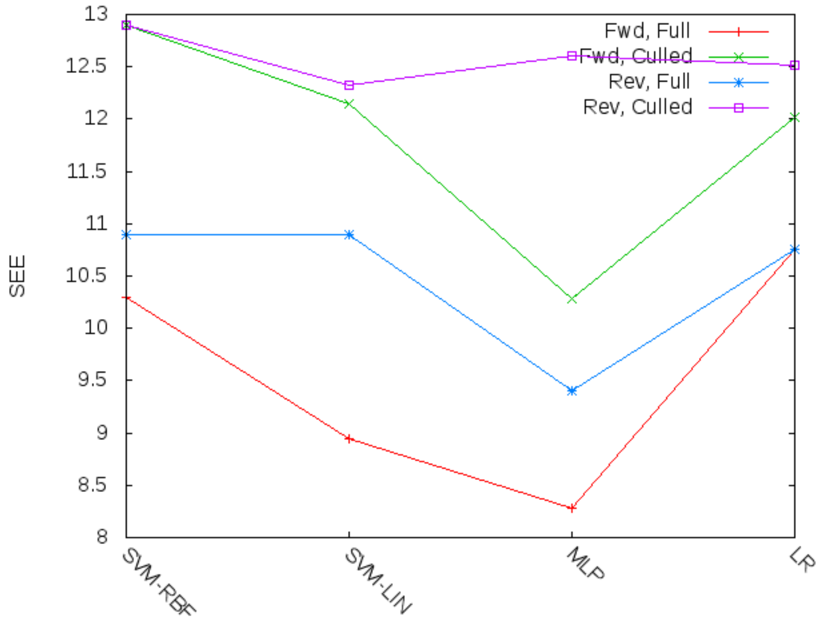

Figure 3.1: A comparison of the best performing models across feature selection algorithm and data set combinations. The forward search feature selection algorithm on the full patient data set outperforms all other feature selection and data set combinations.

machine learning can rank features independent of domain-specific insight. Additionally it suggests the need to provide data-driven methods with all available data so that counter-intuitive results are not overlooked.

Beginning with just one feature, Tables 5.5 and 5.6 detail the most predictive (error lowering) feature for each iteration on the large and small data sets, respectively.

Just as with the reverse feature search, we see a strong agreement between model classes for the most predictive features for the culled set, and slightly less agreement for the full patient data set. For example, for the culled feature set all four model classes agree that $R^{P_{S2}}_{P_{S1}}$ is the most predictive.

3.3 Predictive Results

Addressing all models, we see two patterns emerge. Figure 3.1 compares how each data set and feature search algorithm pair performed across each model class. Each plotted point

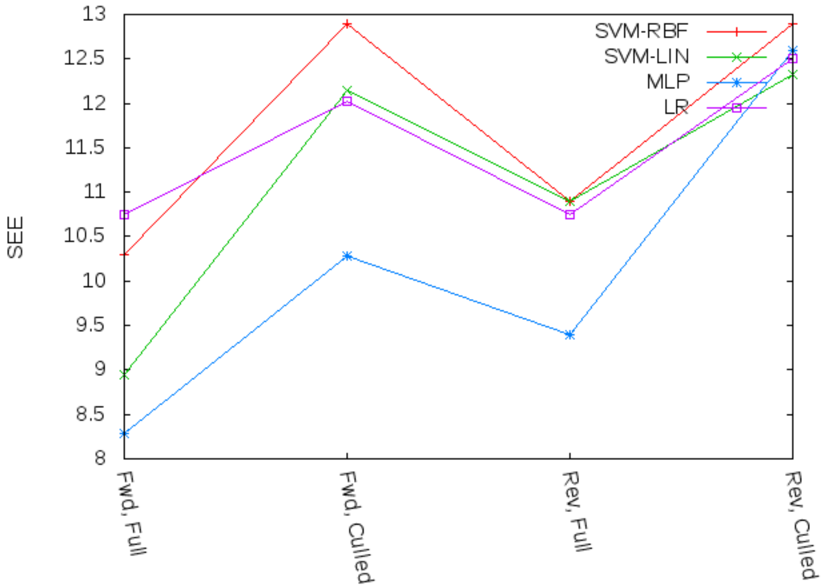

Figure 3.2: A comparison of the best performing models across model classes. The MLP model class dominates all other model classes on all data set and feature selection combinations except reverse search using the culled patient data set.

shows the SEE for the best performing model with the given parameters. The full data set with forward feature set dominates the performance of the other parameter choices.

The poor performance of the culled data set shows that the length of the sound recording is quite important. Missing values, whose rows were removed in the culled data set, resulted from signals whose recording lengths were insufficiently long to conduct the Fourier transform necessary to generate many of the features. The initial hypothesis was that, since most competing studies have culled patients with poor heart beats, perhaps the culled data set would perform better than the full data set. Clearly this is not the case. Other studies succeeded with this method because they culled patients based on poor heart beats. Since their discrimination methods were not replicable by us, we instead decided sound quality based on length of heartbeat. If a heartbeat was insufficiently long to perform a Fourier transform, it was considered too poor to count and removed from the culled data set.

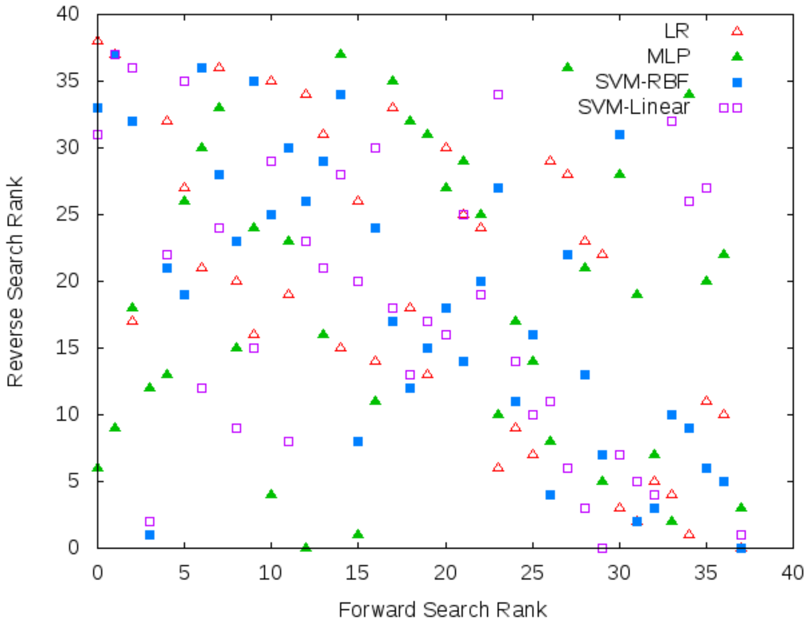

Figure 3.3: A comparison of feature ranking between forward search and reverse search for each model class using the full data set. Note the feature rank variance between the model classes and between the search algorithms.

When the performance data is transposed to plot the model class performance on each feature selection algorithm and data set combination, we see that MLP dominates the other model classes for all parameters except reverse search with the culled patient data set (see Figure 3.2). We also notice that this exception had the worst general SEE score among the parameter choices.

The question arises whether there is any correlation between the forward and reverse search feature ranks. The graph in Figure 3.3 shows the rank of each feature for the full data set on each search strategy. Although there are some feature rankings that are highly corraborative, most are not. There seems to be no clear agreement in feature rank amongst model classes.

In order to analyze general behavior of the system across both reverse and forward search, we combine the results into a bubble chart of the performance of each of the features across all algorithms and data sets (see Figure 3.4). We would expect that if some physiological feature is strongly correlated with PAP, we would see that feature in the bottom

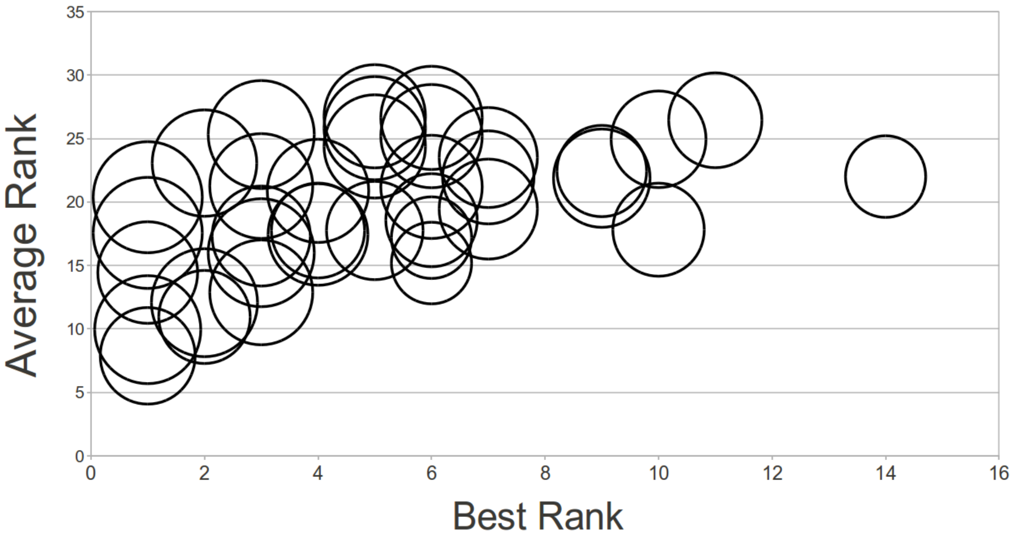

Figure 3.4: An agglomerative view of feature performance for forward and reverse search, for all model classes and patient data sets. Each of the 38 features is represented with a bubble with bubble diameter representing the range of ranks for this feature. The unscaled diameters range from 20 to 37. The smaller the bubble, the more consistent the performance of the feature.

left of the graph, indicating that it ensures a high score when it is used, and also that the feature would have a small variance (indicated by a small diameter in the figure). Though there are several features located in the bottom left of the graph, none of those features has low variance. In fact, not even the features with very low best ranking are consistently good. This confirms that the feature search must include the choice of model class and data set. Slightly more interesting is the fact that, in this case study, even if we fix the data set there is still significant variation in the feature ranks across model classes. Even if the model class and data set are fixed, there is significant difference in the ranks (see Appendix).

The variation of rankings for features among feature search algorithms, model classes, and data sets suggests that the selection of χ cannot be decoupled from the selection of f and $\boldsymbol{\alpha}$. In other words, it is highly unlikely that there exists a set of features which is predictive across all choices of f and $\boldsymbol{\alpha}$. For clinical thinkers this is an unexpected result, since they operate under the assumption that some heart sound features must be more clinically correlated with PAP than others. Perhaps the medically-suggested features we used did not include these univerally relevant features. However, given the fact that Dennis *et al.* built their feature list starting with those suggested in the medical literature, it is possible that such features do not exist—a surprising prospect.

3.4 Comparison to Previous Work

Table 3.1 outlines the results of our experiments described in this paper, as well as the results of two leading PAP predictive studies from the literature. The experiments by Xu *et al.* yield an error rate of 5.8 mmHg [4]. However, their model is derived and tested on the same set of data. In other words, their model has not been tested on held out data to estimate clinical performance on unseen data. To show that models built and tested on the same set of data are biased on training data and can perform worse on new data, and to create a fair baseline of comparison, we create a subset of our data consisting of 23 randomly selected patients. We train and test models on this subset of patients with different combinations of four features (they used two) until we find one combination which yields a SEE of 5.57 mmHg. We then apply the model to our other 86 hold out patients. The resulting SEE is 11.14 mmHg. Without replicating their experiments on new patients, we cannot say with certainty just how much more error their model would produce, but it is well-known that fitted models tend to perform much worse on new data than their fitted error measures indicate.

One study uses data from pigs [6]. Unlike our experiments, their heartbeats are randomly assigned to training and test sets without regard to maintaining each subject's heartbeats in the same set. Since intra-subject heartbeat variation is likely significantly less than extra-subject heartbeat variation, the error rates reported in this study are likely much lower than they would be with each subject's heartbeats restricted to either the training or test set. Additionally, another study showed that pig results do not correlate with human results [4].

Table 3.1: *Best Results Compared to Existing Phonocardiographical Studies.* All studies measure mean PAP. Study [6] has a low SEE, but patient data was mixed between the training and holdout sets, making their results biased. The model from [4] was trained and tested on the same data set. We show that it is easy to get a low SEE with this method (see Rand Select-Training), but when the same model is tested on previously unseen data, the results are far worse (see Rand Select-Holdout).

Study	**Algorithm**	**Validation**	**Pats**	**Pats Excl**	**SEE**
Reverse Select(Full)	MLP	Per-patient CV	109	0	9.40
Reverse Select(Cull)	SVM-LINEAR	Per-patient CV	109	66	12.33
Forward Select(Full)	MLP	Per-patient CV	109	0	8.28
Forward Select(Cull)	MLP	Per-patient CV	109	66	10.28
Rand Select	SVM-RBF	Training	23	0	5.57
Rand Select	SVM-RBF	23 Training 86 Holdout	109	0	11.14
Xu *et al.* [4]	LR	Training	23	2	5.8
Tranulis *et al.* [6]	NN	1/3 Mixed Holdout	9 Pigs	0	6.0

3.5 Most Predictive Configuration

The most predictive general model is our Forward Select model on the full data set, which has a SEE of 8.28 mmHg using MLP and a feature set of size 8. Because this result uses a two-patient (or leave-two-out) cross-validation, we claim that it is not overtrained. Thus, we expect our model would prove more accurate on new data than the other models in the literature. This result is arguably better than previous results, even before considering that, unlike prior work, we have used cross-validation to minimize overtraining and no patients have been excluded. Thus, we expect this model will perform better on unseen data, and also will perform better on the general population.

Our best result removing short-recording patients has a SEE of 10.28 mmHg using MLP with a feature set size of 19.

While our patient set is larger than others in the literature, it is still small. The poorer scores among the small data set experiments suggests that, for this problem, more data, even noisy data, increases accuracy. Perhaps patient exclusion would help improve our models if we had a larger number of patients.

4 CONCLUSION

We have shown that our system is accurate and can be used for continuous PAP estimation. Additionally, because our data came from external microphone recordings, our system is noninvasive. Our results suggest that phonocardiography should be considered as a clinical alternative for surgical PAP measurement. Healthy PAP is less than 25 mmHg. Our best general model has a margin of error of ±8.28 mmHg. Thus, patients who obtain a result of <16.72 mmHg are most likely healthy while those with a result of > 33.28 mmHg are most likely hypertensive. However, the real strength of this model is that patients who know they are hypertensive can use our system to monitor their PAP daily (if need be) in their homes. Our results suggest that some time in the future, surgical PAP measurements could be replaced with simple drug store blood-pressure-style kiosks. In other words, less surgery, less trips to the doctor, and no need for long-term stays for monitoring. Either for classification or for continuous monitoring, our margin of error suggests that this system is useful in its current stage of development.

By definition, our cross-validation techniques yield a model which is generalized across the 109 patient data set used. However, to make such a claim of generality on the entire patient population, our techniques must be used with more of the data from the 200,000 patients who undergo right heart catheterization each year. The larger the data set used in these experiments, the more confidently we can assert that the error rates will be consistently low. We anticipate that our results will motivate the collection of a larger data set in order to create an even more accurate model.

As discussed in the Results section, our feature selection results suggest that the feature selection problem is really the algorithm-and-feature selection problem. The choice of a feature subset is coupled to the choice of algorithm. Without using a true wrapper approach (using the same model class for both feature selection and prediction, as opposed to Guyon and Bennett's method of using different model classes for each), we could not have provided the evidence we have to support this claim. The implications are that saving time by using an algorithm-dependent feature selection technique (such as that of Guyon), one can completely overlook a valid and better-performing feature subset. This implies that an adequate feature selection search is time consuming, and that future work should be dedicated to finding a faster, more analytical method for finding an approximately-optimal feature subset and algorithm given a data set.

References

[1] S. Aggio, E. Baracca, C. Longhini, C. Brunazzi, L. Longhini, G. Musacci, and C. Fersini. Noninvasive estimation of the pulmonary systolic pressure from the spectral analysis of the second heart sound. *Acta Cardiologica*, 45(3):199–202, 1990.

[2] N. Brechot, L. Gambotti, S. Lafitte, and R. Roundaut. Usefulness of right ventricular isovolumic relaxation time in predicting systolic pulmonary artery pressure. *European Journal of Echocardiography*, 9:547–554, 2008.

[3] B. Stephen, P. Dalal, M. Berger, P. Schweitzer, and S. Hecht. Noninvasive estimation of pulmonary artery diastolic pressure in patients with tricuspid regurgitation by doppler echocardiography. *Chest*, 116:73–77, 1999.

[4] J. Xu, L. Durand, and P. Pibarot. A new, simple, and accurate method for non-invasive estimation of pulmonary arterial pressure. *Heart (British Cardiac Society)*, 88(1):76–80, 2002.

[5] J. Testani, M. Sutton, S. Wiegers, A. Khera, R. Shannon, and J. Kirkpatrick. Accuracy of noninvasively determined pulmonary artery systolic pressure. *American Journal of Cardiology*, 105:1192–1197, 2010.

[6] C. Tranulis, L. Durand, L. Senhadji, and P. Pibarot. Estimation of pulmonary arterial pressure by a neural network analysis using features based on time-frequency representations of the second heart sound. *Medical and Biological Engineering and Computing*, 40(2):205–212, 2002.

[7] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. *Artificial Intelligence in Medicine*, 23(1):89–109, 2001.

[8] Paul Sajda. Machine learning for detection and diagnosis of disease. *Annual Review of Biomedical Engineering*, 8(1):537–565, 2006.

[9] A. Dennis, A. Michaels, P. Arand, and D. Ventura. Noninvasive diagnosis of pulmonary hypertension using heart sound analysis. *Computers in Biology and Medicine*, 40(9):758–764, 2010.

[10] I. Tsamardinos and C. Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[11] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

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

[13] J. Bi, K. Bennett, M. Embrechts, C. Breneman, and M. Song. Dimensionality reduction via sparse support vector machines. *Journal of Machine Learning Research*, 3:1229–1243, 2003.

[14] B.L. Milenova, J.S. Yarmus, and M.M. Campos. SVM in oracle database 10 g: Removing the barriers to widespread adoption of support vector machines. In *Proceedings of the 31st International Conference on Very Large Data Bases*, pages 1152–1163, 2005.

5 APPENDIX

Table 5.1: *Reverse Select Results on Both Data Sets*. Here we compare the standard estimate of error (SEE) results of each feature set size on the full patient set (left) and culled patient set (right) for each model class, where features were selected using reverse search (see Algorithm 2.1). The best results for each data set are in bold with ties broken by lowest number of features. Each SEE is averaged over a per-patient cross-validation.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Regression**
38	11.58/13.52	13.69/13.22	10.95/16.19	13.69/14.50
37	11.51/13.36	13.03/13.30	10.66/13.50	13.29/13.72
36	11.29/13.19	12.33/12.95	9.41/**12.60**	13.08/13.51
35	11.27/13.19	12.33/12.83	10.32/14.75	12.83/13.44
34	11.36/13.25	12.26/12.77	10.42/14.35	12.64/13.37
33	11.38/13.24	11.90/12.75	10.94/13.37	12.31/13.31
32	11.26/13.16	11.15/12.70	10.75/13.40	12.03/13.26
31	11.25/12.99	11.43/12.70	10.75/13.90	11.81/13.21
30	11.10/13.06	11.52/12.64	10.00/13.26	11.70/13.18
29	11.10/12.96	11.47/12.59	11.57/13.05	11.68/13.14
28	11.00/13.03	11.49/12.55	10.54/13.67	11.21/13.13
27	10.96/13.02	11.36/12.54	10.88/13.39	11.15/13.12
26	10.97/12.99	11.19/12.49	10.11/12.98	11.11/13.11
25	10.99/12.93	11.33/11.76	11.33/13.51	11.09/13.11
24	11.03/12.96	11.34/12.49	11.39/13.24	11.07/13.11
23	10.92/**12.90**	11.31/12.44	11.10/13.13	11.06/13.08
22	10.92/12.90	11.29/12.41	9.73/12.87	11.05/13.05
21	10.97/13.01	11.09/12.42	9.60/13.70	11.04/13.05
20	11.01/12.98	11.14/12.43	10.26/13.71	11.02/13.04
19	10.92/12.91	11.23/12.39	10.65/13.64	11.02/13.01
18	10.92/12.96	11.07/12.39	**9.40**/14.93	11.02/13.00
17	11.00/12.97	11.09/12.42	9.78/13.46	11.02/13.00
16	10.91/13.01	11.29/12.42	10.07/13.15	10.95/13.00
15	10.93/13.04	11.23/12.35	10.71/15.49	10.95/12.93
14	**10.90**/12.93	11.21/12.38	10.56/13.45	10.88/12.91
13	10.94/12.94	11.23/12.36	10.64/13.85	10.88/12.91
12	10.91/13.00	11.21/12.45	11.14/13.17	10.88/12.93
11	11.02/12.93	11.15/**12.33**	10.26/12.88	10.88/13.00
10	10.93/12.98	11.14/12.38	11.24/15.16	**10.75**/12.97
9	10.95/13.06	11.08/12.35	11.38/13.85	10.75/12.54
8	10.96/12.98	11.03/12.45	11.21/13.70	10.76/12.52
7	10.96/13.02	11.20/12.48	11.13/13.64	10.79/**12.51**
6	10.94/13.16	11.12/12.47	11.05/13.61	10.83/12.57
5	10.94/13.19	**10.90**/12.35	11.01/13.58	10.95/12.61
4	10.98/13.29	10.92/12.36	11.00/13.55	10.96/12.55
3	11.02/13.62	10.96/13.19	10.98/13.57	10.96/13.32
2	11.05/13.97	11.05/12.75	11.08/14.52	11.13/13.13
1	20.50/19.50	20.50/19.50	11.33/14.84	11.33/14.84

Table 5.2: *Feature Dropped on Each Iteration (Full Data Set, Reverse Search).* This table shows the least predictive feature for each feature set size for the large data set, or a ranking of the worst features, with the least predictive at the top.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Reg**
38	F_{HB}	F_{HB}	$R^{Q_{P2}}_{Q_{A2}}$	F_{HB}
37	$R^{P_{A2}}_{P_{S2}}$	Q_{S2}	has_{s3}	F_{S2}
36	F_{S2}	$R^{P_{A2}}_{P_{S2}}$	$\tilde{D}^{P2}_{S1}$	P_{A2}
35	F_{S1}	F_{S2}	HR	P_{S2}
34	$R^{Q_{P2}}_{Q_{A2}}$	P_{A2}	$R^{F_{P2}}_{F_{A2}}$	Q_{S2}
33	D^{A2}_{S1}	F_{S1}	D^{P2}_{S1}	Q_{HB}
32	D^{P2}_{S1}	P_{S1}	P_{P2}	F_{P2}
31	P_{A2}	Q_{S1}	$R^{P_{A2}}_{P_{S1}}$	P_{S1}
30	$R^{P_{P2}}_{P_{S2}}$	Q_{HB}	D^{A2}_{S1}	P_{HB}
29	D^{P2}_{R}	$R^{F_{P2}}_{F_{A2}}$	has_{s4}	Q_{S1}
28	D^{A2}_{R}	P_{S2}	D^{P2}_{R}	F_{S1}
27	Q_{S2}	P_{HB}	SI_{S1}	has_{s4}
26	HR	has_{s4}	SI_{S2}	$R^{F_{P2}}_{F_{A2}}$
25	F_{P2}	NSI_{S2}	NSI_{S1}	NSI_{S2}
24	Q_{HB}	F_{P2}	$\tilde{D}^{A2}_{S1}$	$R^{Q_{P2}}_{Q_{A2}}$
23	SI_{S1}	F_{A2}	$R^{P_{A2}}_{P_{S2}}$	$R^{P_{P2}}_{P_{S2}}$
22	$R^{F_{P2}}_{F_{A2}}$	$R^{Q_{P2}}_{Q_{A2}}$	NSI_{S2}	NSI_{S1}
21	Q_{P2}	SI_{S1}	D^{A2}_{R}	SI_{S1}
20	NSI_{S2}	HR	F_{HB}	F_{A2}
19	F_{A2}	Q_{P2}	$\tilde{D}^{A2}_{R}$	$R^{P_{P2}}_{P_{A2}}$
18	Q_{S1}	$R^{P_{P2}}_{P_{A2}}$	$R^{P_{S2}}_{P_{S1}}$	has_{s3}
17	P_{P2}	$\tilde{D}^{A2}_{R}$	$\tilde{D}^{P2}_{S1}$	D^{A2}_{R}
16	P_{S1}	NSI_{S1}	$R^{P_{P2}}_{P_{S2}}$	D^{P2}_{R}
15	$\tilde{D}^{P2}_{S1}$	$R^{P_{P2}}_{P_{S1}}$	P_{HB}	$\tilde{D}^{P2}_{S1}$
14	$R^{P_{P2}}_{P_{A2}}$	$\tilde{D}^{A2}_{S1}$	P_{S1}	$\tilde{D}^{P2}_{S1}$
13	$\tilde{D}^{P2}_{S1}$	$\tilde{D}^{P2}_{S1}$	F_{S2}	$\tilde{D}^{A2}_{S1}$
12	$R^{P_{A2}}_{P_{S1}}$	D^{A2}_{R}	F_{A2}	$\tilde{D}^{A2}_{R}$
11	P_{S2}	D^{P2}_{R}	F_{S1}	D^{P2}_{S1}
10	has_{s3}	$R^{P_{S2}}_{P_{S1}}$	$R^{P_{P2}}_{P_{S1}}$	D^{A2}_{S1}
9	$\tilde{D}^{A2}_{R}$	$R^{P_{A2}}_{P_{S1}}$	Q_{S1}	Q_{P2}
8	$R^{P_{P2}}_{P_{S1}}$	$R^{P_{P2}}_{P_{S2}}$	Q_{S2}	$R^{P_{P2}}_{P_{S1}}$
7	$R^{P_{S2}}_{P_{S1}}$	has_{s3}	F_{P2}	Q_{A2}
6	$\tilde{D}^{A2}_{S1}$	D^{A2}_{S1}	P_{A2}	HR
5	Q_{A2}	D^{P2}_{S1}	Q_{HB}	$R^{P_{A2}}_{P_{S2}}$
4	has_{s4}	$\tilde{D}^{P2}_{S1}$	Q_{P2}	$R^{P_{S2}}_{P_{S1}}$
3	NSI_{S1}	Q_{A2}	$R^{P_{P2}}_{P_{A2}}$	$R^{P_{A2}}_{P_{S1}}$
2	SI_{S2}	SI_{S2}	Q_{A2}	P_{P2}
1	P_{HB}	P_{P2}	P_{S2}	SI_{S2}

Table 5.3: *Feature Dropped on Each Iteration (Culled Data Set, Reverse Search)*. This table shows the least predictive feature for each feature set size for the small data set, or a ranking of the worst features, with the least predictive at the top.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Regression**
38	SI_{S1}	$R_{P_{S2}}^{P_{A2}}$	has_{s3}	F_{HB}
37	P_{HB}	P_{P2}	has_{s4}	$R_{P_{S2}}^{P_{A2}}$
36	P_{S1}	$R_{P_{S1}}^{P_{A2}}$	NSI_{S1}	$R_{P_{S1}}^{P_{A2}}$
35	D_{S1}^{P2}	SI_{S1}	SI_{S1}	SI_{S1}
34	D_{S1}^{A2}	NSI_{S1}	D_{S1}^{A2}	NSI_{S2}
33	D_{R}^{A2}	$R_{P_{A2}}^{P_{P2}}$	$R_{P_{S1}}^{P_{S2}}$	Q_{S1}
32	D_{R}^{P2}	$R_{F_{A2}}^{F_{P2}}$	$R_{P_{S1}}^{P_{P2}}$	$R_{Q_{A2}}^{Q_{P2}}$
31	HR	$R_{P_{S2}}^{P_{P2}}$	$R_{P_{S1}}^{P_{A2}}$	HR
30	$R_{P_{S1}}^{P_{A2}}$	P_{A2}	$R_{P_{S2}}^{P_{P2}}$	F_{P2}
29	$R_{P_{S2}}^{P_{P2}}$	F_{S1}	SI_{S2}	$R_{P_{A2}}^{P_{P2}}$
28	P_{P2}	$R_{P_{S1}}^{P_{P2}}$	$\tilde{D}_{S1}^{P2}$	Q_{A2}
27	$R_{Q_{A2}}^{Q_{P2}}$	F_{P2}	D_{S1}^{P2}	F_{S2}
26	$R_{F_{A2}}^{F_{P2}}$	HR	F_{S2}	F_{A2}
25	F_{A2}	$\tilde{D}_{S1}^{A2}$	HR	has_{s3}
24	$\tilde{D}_{R}^{A2}$	Q_{S1}	NSI_{S2}	$\tilde{D}_{S1}^{P2}$
23	$R_{P_{A2}}^{P_{P2}}$	$R_{Q_{A2}}^{Q_{P2}}$	D_{R}^{A2}	$\tilde{D}_{S1}^{A2}$
22	has_{s3}	$\tilde{D}_{R}^{A2}$	$R_{P_{S2}}^{P_{A2}}$	$\tilde{D}_{S1}^{P2}$
21	NSI_{S2}	NSI_{S2}	D_{R}^{P2}	$\tilde{D}_{R}^{A2}$
20	Q_{S1}	has_{s3}	$R_{F_{A2}}^{F_{P2}}$	$R_{F_{A2}}^{F_{P2}}$
19	P_{A2}	Q_{P2}	$\tilde{D}_{R}^{A2}$	Q_{P2}
18	P_{S2}	F_{S2}	$R_{Q_{A2}}^{Q_{P2}}$	Q_{S2}
17	$\tilde{D}_{S1}^{A2}$	SI_{S2}	$\tilde{D}_{S1}^{A2}$	SI_{S2}
16	Q_{S2}	F_{A2}	P_{S1}	D_{R}^{A2}
15	F_{P2}	Q_{HB}	P_{HB}	D_{R}^{P2}
14	F_{S1}	Q_{S2}	F_{P2}	Q_{HB}
13	F_{S2}	D_{R}^{P2}	Q_{HB}	$R_{P_{S1}}^{P_{P2}}$
12	Q_{A2}	$\tilde{D}_{S1}^{P2}$	$R_{P_{A2}}^{P_{P2}}$	F_{S1}
11	$\tilde{D}_{S1}^{P2}$	$\tilde{D}_{S1}^{P2}$	P_{P2}	$R_{P_{S2}}^{P_{P2}}$
10	Q_{HB}	P_{S2}	$\tilde{D}_{S1}^{P2}$	D_{S1}^{P2}
9	$\tilde{D}_{S1}^{P2}$	Q_{A2}	P_{A2}	D_{S1}^{A2}
8	Q_{P2}	F_{HB}	F_{A2}	P_{S2}
7	F_{HB}	D_{S1}^{P2}	Q_{S1}	has_{s4}
6	SI_{S2}	D_{S1}^{A2}	Q_{S2}	NSI_{S1}
5	NSI_{S1}	D_{R}^{A2}	Q_{P2}	P_{A2}
4	$R_{P_{S2}}^{P_{A2}}$	has_{s4}	F_{S1}	P_{P2}
3	$R_{P_{S1}}^{P_{P2}}$	P_{HB}	Q_{A2}	P_{S1}
2	has_{s4}	P_{S1}	P_{S2}	P_{HB}
1	$R_{P_{S1}}^{P_{S2}}$	$R_{P_{S1}}^{P_{S2}}$	F_{HB}	$R_{P_{S1}}^{P_{S2}}$

Table 5.4: *Forward Select Results on Both Data Sets.* Here we compare the standard estimate of error (SEE) results of each feature set size on the full patient set (left) and culled patient set (right) for each model class, where features were selected using forward search (see Algorithm 2.2). The best results for each data set are in bold with ties broken by lowest number of features. Each SEE is averaged over a per-patient cross-validation.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Regression**
1	11.07/14.11	**8.94**/12.72	10.53/11.33	11.13/13.13
2	11.03/13.60	11.05/12.59	10.26/12.25	10.96/12.83
3	11.02/13.41	10.95/12.48	9.52/10.64	10.89/12.60
4	11.02/13.14	10.91/12.38	10.25/13.45	10.85/12.46
5	10.93/13.10	10.81/12.41	10.15/11.83	10.81/12.34
6	10.95/13.04	10.82/12.32	9.42/10.75	10.80/12.21
7	10.91/12.93	10.80/12.26	8.95/10.82	10.80/12.14
8	10.90/13.02	10.80/12.15	**8.28**/11.01	10.80/12.10
9	10.91/13.01	10.81/12.16	9.45/10.50	10.80/12.07
10	10.98/12.90	10.90/**12.14**	9.85/11.12	10.81/12.07
11	10.94/13.01	10.83/12.19	8.65/11.72	10.74/12.07
12	10.92/**12.90**	10.87/12.18	8.84/11.48	**10.75**/12.08
13	10.91/13.02	11.01/12.14	10.19/11.16	10.76/12.08
14	11.01/13.00	10.92/12.16	10.41/11.05	10.74/12.09
15	10.92/12.90	11.00/12.21	10.55/11.08	10.76/12.11
16	10.95/13.01	11.02/12.22	10.63/10.68	10.78/12.13
17	10.94/13.02	10.90/12.16	10.57/10.38	10.80/12.17
18	10.94/13.01	11.07/12.17	10.64/11.57	10.81/12.22
19	10.98/13.01	11.09/12.18	10.60/**10.28**	10.83/12.05
20	10.97/12.96	11.07/12.22	10.69/10.93	10.86/**12.02**
21	11.04/13.00	11.11/12.21	10.66/11.26	10.92/12.06
22	10.97/12.90	11.21/12.29	10.67/11.03	10.98/12.10
23	10.97/12.91	11.25/12.24	10.65/10.91	10.98/12.14
24	**10.29**/13.00	11.06/12.24	10.78/11.11	11.07/12.19
25	11.01/12.94	11.21/12.28	10.77/10.93	11.18/12.16
26	11.04/12.91	11.37/12.43	10.84/10.76	11.34/12.25
27	11.05/13.01	11.53/12.59	11.69/10.36	11.53/12.32
28	11.10/13.02	11.70/12.65	11.59/11.42	11.53/12.42
29	11.16/13.01	11.00/12.56	11.51/12.47	11.58/12.56
30	11.13/13.08	11.81/12.71	11.70/12.56	11.58/12.68
31	11.29/13.08	16.39/12.76	12.10/12.75	11.84/12.68
32	11.30/13.15	15.71/12.97	11.51/12.77	12.07/12.84
33	11.33/13.32	16.09/13.12	13.74/13.01	12.34/12.84
34	11.39/13.32	17.74/13.23	14.32/13.67	12.70/13.19
35	11.46/13.34	14.31/13.40	12.27/14.04	13.05/13.33
36	11.55/13.42	17.75/13.15	12.58/14.48	13.67/13.48
37	11.56/13.45	16.66/13.27	14.42/14.72	13.69/15.14
38	11.61/13.55	15.25/13.50	15.36/14.80	16.67/15.48

Table 5.5: *Feature Dropped on Each Iteration (Full Data Set, Forward Search)*. This table shows the most predictive feature for each feature set size for the large data set, or a ranking of the best features, with the most predictive at the top.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Regression**
1	P_{HB}	has_{s3}	P_{P2}	SI_{S2}
2	SI_{S2}	P_{P2}	has_{s4}	P_{P2}
3	NSI_{S1}	SI_{S2}	F_{HB}	NSI_{S1}
4	$R_{P_{S2}}^{P_{A2}}$	$R_{P_{S2}}^{P_{A2}}$	SI_{S2}	has_{s4}
5	P_{P2}	NSI_{S1}	NSI_{S1}	Q_{A2}
6	F_{A2}	Q_{A2}	F_{A2}	$\tilde{D}_{R}^{A2}$
7	Q_{A2}	has_{s4}	Q_{S2}	has_{s3}
8	has_{s3}	$\tilde{D}_{S1}^{A2}$	Q_{HB}	$R_{P_{S1}}^{P_{A2}}$
9	$\tilde{D}_{S1}^{P2}$	$R_{F_{A2}}^{F_{P2}}$	$R_{P_{S2}}^{P_{A2}}$	$R_{P_{A2}}^{P_{P2}}$
10	$R_{P_{S1}}^{P_{S2}}$	F_{A2}	P_{S1}	$R_{P_{S2}}^{P_{P2}}$
11	$\tilde{D}_{S1}^{P2}$	$R_{P_{S1}}^{P_{A2}}$	$R_{F_{A2}}^{F_{P2}}$	$R_{P_{S1}}^{P_{S2}}$
12	$R_{P_{S1}}^{P_{P2}}$	Q_{HB}	P_{HB}	F_{A2}
13	$R_{P_{S1}}^{P_{A2}}$	$R_{P_{S1}}^{P_{P2}}$	$R_{Q_{A2}}^{Q_{P2}}$	$R_{P_{S2}}^{P_{A2}}$
14	$\tilde{D}_{R}^{A2}$	$\tilde{D}_{R}^{A2}$	NSI_{S2}	$R_{P_{S1}}^{P_{P2}}$
15	$\tilde{D}_{S1}^{A2}$	$R_{P_{S1}}^{P_{S2}}$	P_{S2}	$R_{Q_{A2}}^{Q_{P2}}$
16	$R_{P_{S2}}^{P_{P2}}$	$R_{P_{A2}}^{P_{P2}}$	has_{s3}	$\tilde{D}_{S1}^{A2}$
17	$R_{P_{A2}}^{P_{P2}}$	$R_{P_{S2}}^{P_{P2}}$	SI_{S1}	NSI_{S2}
18	Q_{P2}	HR	$R_{P_{A2}}^{P_{P2}}$	HR
19	HR	NSI_{S2}	P_{A2}	SI_{S1}
20	SI_{S1}	SI_{S1}	F_{P2}	$R_{F_{A2}}^{F_{P2}}$
21	NSI_{S2}	$R_{Q_{A2}}^{Q_{P2}}$	F_{S1}	Q_{P2}
22	Q_{HB}	$\tilde{D}_{S1}^{P2}$	Q_{S1}	$\tilde{D}_{S1}^{P2}$
23	Q_{S1}	Q_{P2}	F_{S2}	$\tilde{D}_{S1}^{P2}$
24	P_{S2}	$\tilde{D}_{S1}^{P2}$	D_{R}^{P2}	F_{P2}
25	Q_{S2}	F_{P2}	D_{R}^{A2}	P_{HB}
26	$R_{F_{A2}}^{F_{P2}}$	P_{S2}	$\tilde{D}_{S1}^{A2}$	P_{S1}
27	$R_{Q_{A2}}^{Q_{P2}}$	P_{HB}	D_{S1}^{A2}	D_{S1}^{A2}
28	P_{S1}	P_{S1}	Q_{A2}	D_{S1}^{P2}
29	F_{P2}	F_{S2}	$\tilde{D}_{S1}^{P2}$	D_{R}^{P2}
30	P_{A2}	F_{HB}	D_{S1}^{P2}	D_{R}^{A2}
31	has_{s4}	Q_{S1}	$R_{P_{S1}}^{P_{P2}}$	P_{S2}
32	F_{S2}	F_{S1}	$\tilde{D}_{R}^{A2}$	P_{A2}
33	F_{S1}	P_{A2}	$R_{P_{S1}}^{P_{A2}}$	Q_{HB}
34	D_{R}^{A2}	D_{S1}^{A2}	$\tilde{D}_{S1}^{P2}$	Q_{S2}
35	D_{R}^{P2}	D_{R}^{A2}	Q_{P2}	F_{S2}
36	D_{S1}^{P2}	D_{R}^{P2}	$R_{P_{S1}}^{P_{S2}}$	F_{S1}
37	D_{S1}^{A2}	D_{S1}^{P2}	$R_{P_{S2}}^{P_{P2}}$	Q_{S1}
38	F_{HB}	Q_{S2}	HR	F_{HB}

Table 5.6: *Feature Dropped on Each Iteration (Culled Data Set, Forward Search)*. This table shows the most predictive feature for each feature set size for the small data set, or a ranking of the best features, with the most predictive at the top.

Features	**SVM-RBF**	**SVM-Linear**	**MLP**	**Linear Regression**
1	$R^{P_{S2}}_{P_{S1}}$	$R^{P_{P2}}_{P_{S1}}$	$R^{P_{S2}}_{P_{S1}}$	$R^{P_{S2}}_{P_{S1}}$
2	has_{s4}	has_{s4}	SI_{S2}	F_{HB}
3	$R^{P_{P2}}_{P_{S1}}$	SI_{S2}	SI_{S1}	$R^{P_{A2}}_{P_{S2}}$
4	$R^{P_{A2}}_{P_{S2}}$	F_{HB}	F_{S1}	has_{s4}
5	SI_{S2}	NSI_{S1}	has_{s4}	NSI_{S1}
6	NSI_{S1}	F_{S1}	P_{HB}	SI_{S2}
7	$\tilde{D}^{P2}_{S1}$	$R^{P_{A2}}_{P_{S2}}$	$R^{P_{P2}}_{P_{S2}}$	Q_{HB}
8	P_{S2}	$R^{P_{P2}}_{P_{S2}}$	F_{A2}	$R^{P_{P2}}_{P_{S2}}$
9	F_{HB}	$R^{P_{S2}}_{P_{S1}}$	Q_{A2}	$R^{Q_{P2}}_{Q_{A2}}$
10	F_{S2}	Q_{HB}	$R^{P_{A2}}_{P_{S2}}$	$\tilde{D}^{A2}_{S1}$
11	F_{S1}	$\tilde{D}^{P2}_{S1}$	$R^{P_{P2}}_{P_{A2}}$	Q_{A2}
12	$\tilde{D}^{P2}_{S1}$	has_{s3}	$R^{Q_{P2}}_{Q_{A2}}$	has_{s3}
13	Q_{S1}	F_{A2}	$R^{P_{A2}}_{P_{S1}}$	Q_{S2}
14	$R^{Q_{P2}}_{Q_{A2}}$	$\tilde{D}^{P2}_{S1}$	F_{P2}	Q_{P2}
15	NSI_{S2}	$R^{P_{P2}}_{P_{A2}}$	$R^{F_{P2}}_{F_{A2}}$	$R^{P_{P2}}_{P_{A2}}$
16	Q_{S2}	$\tilde{D}^{A2}_{S1}$	$\tilde{D}^{A2}_{R}$	F_{S2}
17	Q_{A2}	Q_{S2}	$\tilde{D}^{P2}_{S1}$	F_{A2}
18	has_{s3}	Q_{P2}	F_{HB}	$R^{P_{P2}}_{P_{S1}}$
19	F_{P2}	$\tilde{D}^{A2}_{R}$	P_{A2}	$R^{P_{A2}}_{P_{S1}}$
20	P_{A2}	$R^{F_{P2}}_{F_{A2}}$	NSI_{S2}	$\tilde{D}^{P2}_{S1}$
21	Q_{P2}	$R^{Q_{P2}}_{Q_{A2}}$	$\tilde{D}^{A2}_{S1}$	F_{P2}
22	F_{A2}	F_{S2}	$R^{P_{P2}}_{P_{S1}}$	$R^{F_{P2}}_{F_{A2}}$
23	Q_{HB}	Q_{A2}	P_{S1}	HR
24	$\tilde{D}^{A2}_{S1}$	F_{P2}	Q_{HB}	Q_{S1}
25	$R^{P_{P2}}_{P_{A2}}$	Q_{S1}	Q_{P2}	F_{S1}
26	$\tilde{D}^{A2}_{R}$	$R^{P_{A2}}_{P_{S1}}$	F_{S2}	NSI_{S2}
27	$R^{F_{P2}}_{F_{A2}}$	HR	NSI_{S1}	$\tilde{D}^{A2}_{R}$
28	P_{P2}	NSI_{S2}	Q_{S2}	SI_{S1}
29	$R^{P_{A2}}_{P_{S1}}$	SI_{S1}	HR	$\tilde{D}^{P2}_{S1}$
30	$R^{P_{P2}}_{P_{S2}}$	D^{P2}_{R}	Q_{S1}	D^{A2}_{R}
31	HR	D^{A2}_{R}	$\tilde{D}^{P2}_{S1}$	D^{P2}_{R}
32	P_{HB}	P_{A2}	has_{s3}	D^{P2}_{S1}
33	P_{S1}	D^{A2}_{S1}	P_{S2}	D^{A2}_{S1}
34	SI_{S1}	D^{P2}_{S1}	D^{P2}_{S1}	P_{A2}
35	D^{P2}_{R}	P_{S1}	D^{A2}_{S1}	P_{S1}
36	D^{A2}_{R}	P_{HB}	P_{P2}	P_{HB}
37	D^{P2}_{S1}	P_{S2}	D^{A2}_{R}	P_{P2}
38	D^{A2}_{S1}	P_{P2}	D^{P2}_{R}	P_{S2}