MACHINE LEARNING TECHNIQUES FOR ALLEVIATING INHERENT DIFFICULTIES IN BIOINFORMATICS DATA

by

David J. Dittman II

A Dissertation Submitted to the Faculty of

The College of Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

Florida Atlantic University

Boca Raton, FL

May 2015

MACHINE LEARNING TECHNIQUES FOR ALLEVIATING INHERENT DIFFICULTIES IN BIOINFORMATICS DATA

by

David J. Dittman II

This dissertation was prepared under the direction of the candidate's dissertation advisor, Dr. Taghi M. Khoshgoftaar, Department of Computer and Electrical Engineering and Computer Science, and has been approved by the members of his supervisory committee. It was submitted to the faculty of the College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

SUPERVISORY COMMITTEE:

Taghi M. Khoshgoftaar, Ph.D.
Dissertation Advisor

Martin K. Solomon, Ph.D.

Xingquan Zhu, Ph.D.

Hanqi Zhuang, Ph.D.

Nurgun Erdol, Ph.D.
Chair, Department of Computer and Electrical Engineering and Computer Science

Mohammad Ilyas, Ph.D.
Dean, The College of Engineering and Computer Science

Deborah L. Floyd, Ed.D.
Dean, Graduate College

3/27/15
Date

ACKNOWLEDGEMENTS

Foremost, I would like to acknowledge my graduate advisor and mentor, Dr. Taghi M. Khoshgoftaar. His unwavering support, patience, and knowledge have helped shaped the researcher I am today. I would also like to thank Dr. Martin K. Solomon, Dr. Xingquan Zhu, ands Dr. Hanqi Zhuang, for being on my defense committee.

I want to thank Dr. Randall Wald and Dr. Amri Napolitano who read through countless versions of my papers in my preparation of this dissertation. My thanks also go to the members of the Data Mining and Machine Learning Laboratories at Florida Atlantic University.

Finally, I wish to thank the people of The Ultimate Software Group, Inc. who not only gave me the job which helped finance my graduate education, but were very accommodating in allowing me to set my own hours so as not to interfere with my classes or my research.

ABSTRACT

Author:	David J. Dittman II
Title:	Machine Learning Techniques for Alleviating Inherent Difficulties in Bioinformatics Data
Institution:	Florida Atlantic University
Dissertation Advisor:	Dr. Taghi M. Khoshgoftaar
Degree:	Doctor of Philosophy
Year:	2015

In response to the massive amounts of data that make up a large number of bioinformatics datasets, it has become increasingly necessary for researchers to use computers to aid them in their endeavors. With difficulties such as high-dimensionality, class imbalance, noisy data, and difficult to learn class boundaries, being present within the data, bioinformatics datasets are a challenge to work with. One potential source of assistance is the domain of data mining and machine learning, a field which focuses on working with these large amounts of data and develops techniques to discover new trends and patterns that are hidden within the data and to increases the capability of researchers and practitioners to work with this data. Within this domain there are techniques designed to eliminate irrelevant or redundant features, balance the membership of the classes, handle errors found in the data, and build predictive models for future data.

This dissertation is an in-depth analysis of how the domain of data mining and machine learning is uniquely suited for alleviating the inherent difficulties found within bioinformatics datasets. First, we will present a number of different gene selection techniques in terms of their stability or robustness. Next, we will present an

analysis of the entire process of ensemble gene selection including different approaches for implementing the ensemble and ranked feature list aggregation. Next, we will then provide a framework for using gene selection and classification with the focus of maximizing classification performance while simplifying the machine learning process. Then, we will discuss two new approaches for incorporating ensemble learning along with gene selection while comparing them to the case wherein no ensemble learning approach is applied. Lastly, we will give a detailed analysis of the data sampling process for bioinformatics data including which techniques should be used, when and how they should be applied, and to what extent should the data sampling be performed. Overall, this dissertation presents an thorough analysis on how the use of machine learning techniques can alleviate inherent difficulties found in bioinformatics data.

To my amazing wife, Laura Jane Diaz de Arce, who was of great comfort with her unwavering support and her ability to let me relax no matter how stressed I got during my graduate education. Also, to my loving parents Mr. David J Dittman and Dr. Patricia W Dittman who have instilled in me a sense of hard work and have kept me grounded with their never ending support.

MACHINE LEARNING TECHNIQUES FOR ALLEVIATING INHERENT DIFFICULTIES IN BIOINFORMATICS DATA

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Today, in the field of bioinformatics we have an overabundance of data to work with. However, a large majority of this new data may be irrelevant or redundant in terms of the focus of the research, there may be a smaller number of instances which are of interest to researchers and practitioners, and the data itself can be difficult to extract meaningful information from. Therefore, it is essential to develop methods of alleviating these inherent difficulties.

A perfect example of a type of dataset which has high dimensionality is the DNA microarray or gene expression profile dataset. The DNA microarray is an advancement in the study of genetics, molecular biology, and chemistry which allows researchers to test a sample, or instance, for thousands of different genes simultaneously. This ability is achieved by taking advantage of the fact that mRNA (messenger RNA: the blueprints of proteins) will readily bind to its cDNA (complementary DNA). By creating probes made up of cDNA derived from known genes one can measure the levels of mRNA that are linked to each gene and determine how important the gene is to the sample. These datasets have been used in a number of works for the goal of: identification of diseased tissue [6, 47], identifying and accurately classifying between different types of cancer or subtypes of the same cancer [13, 55], patient response prediction to a cancer treatment [33, 86], and many more. However, these datasets are notorious for being difficult to work with, exhibiting many of the difficulties mentioned above. A possible solution to this issue lies in the domain of data mining.

Data mining is the process of discovering new trends, patterns, and relationships in data. There are a number of steps within the process of data mining including: in-

tegrating the data, pre-processing the data, building an inductive model, and making critical decisions based on what is learned through the inductive model [120]. There are techniques within this domain used to remove irrelevant or redundant data features, balance the data, and build inductive models to not only analyze the current data but potentially infer information on new data.

1.1 MOTIVATION

Due to the large-scale nature of gene databases and modern technology such as DNA microarrays, working with bioinformatics datasets is a challenging endeavour. Problems like large levels of high-dimensionality, gene selection instability, class imbalance, noisy data, and datasets which are simply difficult to build effective predictive models from, make properly handling and analyzing such data almost impossible to do without the assistance of machine learning techniques.

1.1.1 High-Dimensionality

High-dimensionality is, at it's core, a problem of having too much data to easily work with. Specifically high-dimensionality is when there are a very large number of attributes attached to each sample. With so much data to work with it can be very difficult and time consuming to distinguish what is important and what is not. A common method of dealing with this problem is a series of techniques called feature (gene) selection [32]. Feature selection is a method of selecting a smaller subset of the attributes (features) available and analyzing only those attributes. The use of feature selection can identify and remove irrelevant and redundant features, greatly reduce computation time, and possibly improve classification performance. Though data is lost during feature selection, it can assist with creating more efficient and accurate classifiers [96].

1.1.2 Gene Selection Instability

Normally one compares the effectiveness of a feature (gene) selection technique to other techniques, one uses the performance of the classifier built using the subsets chosen by the techniques [45, 119]. Yet, relatively few techniques ever take into account what affect changes in the data will have on the model being built using the previously built gene subset. However, another method of describing how well a feature selection technique performs is through studying its robustness or stability. Stability refers to the feature subset's ability to resist changing when changes are made to the dataset [45]. Some feature selection techniques are more stable than others and techniques like ensemble gene selection were designed to increase stability by taking multiple feature subsets and aggregating them into a more stable feature subset. By using these robust feature selection methods, feature subsets remain relatively unchanged even when there are changes in the data and the researcher can be more confident in the importance of any features chosen by the technique [1].

1.1.3 Class Imbalance

Class imbalance is a frequent problem within bioinformatics datasets [11]. An example of class imbalance is demonstrated by Van Hulse et al. [106] who compared the correlations among nine rankers on five imbalanced datasets and a number of data sampling approaches (algorithms to improve the balance of datasets). Ramaswamy et al. [93] performed feature selection on a dataset where only 16% of the instances are in the class of interest. The presence of class imbalance has the potential to negatively affect the classification performance of classifiers applied toward these imbalanced datasets. A possible reason why class imbalance tends to affect classification performance may be due to the fact that many classification algorithms assume that the classes will have an equal number of instances in the dataset [64]. This assumption can lead to some serious problems including increased bias against the minority

class and an increased number of misclassifications [5]. Additionally, the problem is compounded by the fact that frequently the class of interest tends to be the minority class.

1.1.4 Difficult to Learn Class Boundaries

Sometimes, it is a challenge to distinguish between the different classes within a particular dataset. This can be due to noise found in the data, difficult to distinguish class boundaries, or other data characteristics. Thus, it is important to know what the difficulty-of-learning is for your dataset. Difficulty-of-learning [40] is a measure of how challenging it is to develop quality predictive models from. Datasets which are considered harder-to-learn from, lead to poor results across a wide range of classification algorithms. Although harder-to-learn datasets do allow for less meaningful results, they also provide a greater opportunity for improvement, and thus are useful from the perspective of comparing different algorithms.

1.2 CONTRIBUTIONS

The contributions of this work lie in the application of machine learning techniques toward alleviating the difficulties found in bioinformatics data and provide recommendations toward improving classification results on said data.

- We provide a thorough analysis of the effect of data characteristics (number of attributes, number of instances, difficulty-of-learning, etc.) on gene selection stability in Chapter 3.

- Introduce two new ensemble feature selection approaches and analyze them along with the commonly used approach. Additionally, a complete analysis of the feature rank aggregation process (combining multiple feature ranked lists into a single final list) is found in Chapter 4.

- Present an effective but simple framework for the application of gene selection and classification in Chapter 5.
- Introduce and analyze two new approaches toward combining feature selection and ensemble learning in Chapter 6
- Provide a thorough analysis of the data sampling process and determine which options are best suited for bioinformatics data in Chapter 7

1.3 DISSERTATION STRUCTURE

This dissertation is organized as follows. In Chapter 2, we introduce all of the various techniques and approaches applied within this work. Chapter 3 analyses how various data characteristics affect the stability of chosen gene subsets. Chapter 4 focuses on the process of ensemble gene selection including both the ensemble gene approach and the feature rank aggregation techniques. A framework for a simple and effective application of gene selection and classification is presented in Chapter 5. Chapter 6 introduces and analyze two approaches toward combining gene selection and ensemble learning. In Chapter 7, we present a complete analysis of the data sampling process and provide guidelines towards its application on bioinformatics data. Lastly, we present our conclusions and future work in Chapter 8.

CHAPTER 2

METHODOLOGY

This chapter describes the methodologies utilized in the experiments conducted as a part of this research. More specifically, the data is discussed in 2.1. Feature selection is described in Section 2.2. Section 2.3 outlines the particulars of the data sampling process. The process for measuring stability and similarity are presented in Section 2.4. Section 2.5 describes the classifiers and ensemble learning approaches used in this work. Lastly, Section 2.6 presents the cross-validation process and our classification performance metric

2.1 DATASETS

The list of all 39 datasets used in our experiments along with their characteristics is presented in Tables 2.1 and 2.2. The datasets are either tumor classification or patient response data publicly available through a number of different real-world bioinformatics, genetics, and medical projects. For more information on these datasets, we refer interested readers to the provided citations within Tables 2.1 and 2.2. For each dataset we show its balance level, name, total number of minority-class instances, total number of instances, percentage of instances from the minority class, the number of features or genes, and the average AUC values for all datasets. The last column, Average AUC, represents the difficulty-of-learning level for the dataset [39]. This average AUC value is based on classification models built on raw data using no preprocessing technique such as feature selection. To create these AUC scores, five-fold cross-validation was employed and the average performance from six learners was

Table 2.1: Details of the Datasets: A through G

Name	# Minority Instances	Total # of Instances	% Minority Instances	# of Attributes	Average AUC
Acute Lymphoblastic Leukemia [105]	79	327	24.16%	12559	0.8475
ALL AML Leukemia [105]	25	72	34.72%	7130	0.9091
BCancer50k [45]	200	400	50.00%	54,614	0.8564
Brain Tumor [105]	23	90	25.56%	27,680	0.7210
Breast Cancer [105]	46	97	47.42%	24,482	0.6009
Central Nervous System [106]	21	60	35.00%	7,130	0.5181
CNS MAT [25]	30	90	33.33%	7130	0.8355
Colon [106]	22	62	35.48%	2,001	0.7941
Colon50k [45]	130	400	32.50%	54,614	0.8532
DLBCL NIH [105]	102	240	42.50%	7,400	0.5853
DLBCL [106]	23	47	48.94%	4,027	0.8675
ECML Pancreas [106]	8	90	8.89%	27,680	0.6723
GSE1456 [90]	40	159	25.16%	12,066	0.6108
GSE20271 [101]	26	178	14.61%	22,284	0.5867
GSE25055 [61]	57	306	18.63%	22,284	0.6674
GSE25065 [61]	42	182	23.08%	22,284	0.6384
GSE3494-GPL96-ER [85]	34	247	13.77%	22,284	0.7688
GSE3494-GPL96-Grade [85]	54	249	21.69%	22,284	0.8176
GSE3494-GPL97-ER [85]	34	247	13.77%	22,646	0.7674
GSE3494-GPL97-Grade [85]	54	249	21.69%	22,646	0.7722

used: all those discussed in Section 2.5.1 aside from Logistic Regression, along with two versions of C4.5 decision trees (one using default parameter values, one using Laplace smoothing and no pruning [116]). In essence, the lower the Average AUC value the more difficult it is to build effective models from that dataset [112]. These average AUC values do not bear any other effects in our experiments unless difficulty-of-learning is a factor.

Table 2.2: Details of the Datasets: L through W

Name	# Minority Instances	Total # of Instances	% Minority Instances	# of Attributes	Average AUC
Lung 50k [45]	70	400	17.50%	54,614	0.8150
Lung [105]	64	203	31.53%	12,601	0.8685
Lung Cancer [106]	31	181	17.13%	12534	0.9389
Lung Cancer Ontario [117]	15	39	38.46%	2,881	0.7197
Lung Michigan [12]	10	96	10.42%	7130	0.9738
Lymphoma MAT [25]	19	77	24.68%	7130	0.8366
Lymphoma [106]	23	96	23.96%	4,027	0.8511
MLL Leukemia [105]	20	72	27.78%	12583	0.8962
Mulligan-R-NR [86]	84	169	49.70%	22,284	0.5931
Mulligan-R-PD [86]	41	126	32.54%	22,284	0.6527
Ovarian MAT [25]	16	66	24.24%	6,001	0.7896
Ovarian Cancer [105]	91	253	35.97%	15155	0.9739
Prostate MAT [25]	26	89	29.21%	6001	0.9047
Prostate [105]	59	136	43.38%	12,601	0.7823
Raponi 2007 No SD [94]	10	54	18.52%	22,284	0.4420
Raponi 2007 R+SD [94]	14	58	24.14%	22,284	0.4739
SotiriouMatrixData-Grade [98]	45	99	45.45%	7,651	0.6325
Spira2007 [99]	90	192	46.88%	22,216	0.6661
Watanabe 2006 [115]	11	46	23.91%	12,626	0.4487

2.2 FEATURE SELECTION

Feature selection is a process which chooses an optimal subset of features to be used in later analysis, rather than analyzing the entire dataset. What is interesting is that despite the loss of data, feature selection can be useful in creating efficient and accurate classifiers. There are two main approaches to feature selection: filter and wrapper. Filter feature selection techniques analyze the features without any regard to a classifier. The filter approach uses only the raw dataset to decide which features are to be used to create the best classifier. Since no classifier is used, filter methods

must rely on statistical measures. Filters can either be rankers or subset evaluators, depending on whether they examine features one at a time or in groups. Wrappers, unlike filter approaches, use classifiers when making a decision, and often the classifier used to calculate the score of a particular feature subset is the same one that will be used in the post selection analysis. There are two main disadvantages in the use of wrapper based feature selection techniques: limited utility of chosen features and slow computation time. The slow computation time is further compounded by the degree of high dimensionality making both filter-based subset evaluators and wrappers inappropriate for these analysis. In this work we use either filter-based feature rankers (Section 2.2.1) or ensemble feature selection which uses filter-based feature rankers(Section 2.2.2).

2.2.1 Filter-Based Feature Rankers

The feature rankers (filters) chosen can be placed into three categories: commonly used filter-based feature selection techniques, threshold-based feature selection techniques (TBFS) that were developed by our research team, and First Order Statistics (FOS) based feature selection.

Commonly Used Feature Selection Techniques

Seven commonly used filter-based feature ranking techniques were used in this work: chi-squared [118], information gain [59, 92, 118], gain ratio [92, 118], two versions of ReliefF [78, 75], symmetric uncertainty [60, 118], and SVMAtt [118, 113]. All of these feature selection methods, with the exception of signal-to-noise, are available within the Weka machine learning tool [118], and Weka's default parameter values were used unless otherwise noted. Since most of these methods are widely known, only a brief summary is provided; the interested reader can consult with the included references for further details.

The chi-squared method (CS) utilizes the χ^2 statistic to measure the strength of the relationship between each independent variable and the class. Information Gain (IG) determines the significance of a feature based on the amount by which the entropy of the class decreases when considering that feature. Gain Ratio (GR) is a refinement of Information Gain, adjusting for features that have a large number of values. GR attempts to maximize the information gain of the feature while minimizing its number of values. Symmetric Uncertainty (SU) also adjusts IG to account for attributes with more values, and normalizes its value to lie in the range $[0, 1]$. These techniques utilize the method of Fayyad and Irani [50] to discretize continuous attributes, and all four methods are bivariate, considering the relationship between each attribute and the class, excluding the other independent variables.

ReliefF (RF) randomly samples an example instance from the data and finds its nearest neighbor from the same and opposite class. The values of the attributes of the nearest neighbors are compared to the sampled instance and used to update relevance scores for each attribute. This process is repeated for m examples, as specified by the user. ReliefF extends Relief by handling noise and multi-class datasets [78]. RF is implemented within Weka [118] with the "weight nearest neighbors by their distance" parameter set to false. ReliefF-W (RFW) is similar to RF except the "weight nearest neighbors by their distance" parameter is set to true.

In SVM-Att we begin with the SVM–RFE feature selection technique. However, in SVMAtt we do not perform the recursive feature elimination but we used the weighted list of features created prior to the first elimination as our rankings. We do not perform the recursive feature selection because that would require the building of the SVM model and that defeats the purpose of filter-based feature selection and the process is very computationally expensive

Threshold-Based Feature Selection Techniques

This section describes the TBFS method for feature ranking. These feature ranking techniques were proposed and implemented recently by our research group [45, 105]. In TBFS, each attribute is evaluated against the class, independent of all other features in the dataset. After normalizing each attribute to have a range between 0 and 1, simple classifiers are built for each threshold value $t \in [0, 1]$ according to two different classification rules. The normalized values are treated as posterior probabilities, but since no real classifiers are being built, the TBFS techniques are still considered filter and not wrapper methods. For classification rule 1, examples with a normalized value greater than t are classified P while examples with a normalized value less than t are classified as N (assuming each instance x is assigned to one of two classes $c(x) \in \{P, N\}$). For classification rule 2, examples with a normalized value greater than t are classified N while examples with a normalized value less than t are classified as P. Two different classification rules must be considered to account for the fact that for some attributes, large values of the attribute may have a greater correlation with the positive class, while for other attributes, large values of the attribute may have a greater correlation with the negative class. Metric ω is calculated using the formulas related to each metric either at each threshold t or across all thresholds for both classification rules. Finally, the metric resulting from the classification rule which provides the best value is used as the relevancy measure for that attribute relative to the class.

Many of the metrics ω (e.g., Area Under the ROC Curve (ROC), Area Under the Pecision-Recall Curve (PRC), Geometric Mean (GeoMean), F-Measure (F), and the Kolmogorov-Smirnov statistic (KS)) are primarily used to measure the performance of classification models, using the posterior probabilities computed by such models to classify examples as either negative or positive depending on the classification threshold. The normalized attribute values can be thought of as posterior probabilities, e.g.,

$p(P \mid x) = \hat{X}^j(x)$ for classification rule 1, and the metrics ω are computed against this "posterior." Intuitively, attributes where positive and negative examples are evenly distributed along the distribution of X (feature) produce weak measures ω and poor relevancy scores in a similar manner that poor predictive models have positive and negative examples evenly distributed along the distribution of the posterior probability produced by the model. Note further that TBFS can easily be extended to include additional metrics. The TBFS metrics used in this work are explained below.

Mutual information (MI), like information gain, is a measure of entropy or uncertainty. They differ in that mutual information measures the joint probability of a feature to a class, whereas information gain measures the entropy of the feature within the dataset. The actual definition of mutual information is "the amount by which the knowledge provided by the feature vector decreases the uncertainty about the class" [10]. The equation for mutual information is:

$$\mathrm{MI} = \max_{t \in [0,1]} \sum_{\hat{c}^t \in \{P,N\}} \sum_{c \in \{P,N\}} p(\hat{c}^t, c) \log \frac{p(\hat{c}^t, c)}{p(\hat{c}^t)p(c)}$$

where c represents the actual class of the instance and $\hat{c}^t$ is the predicted class of the instance [10]

F-Measure (F) is derived from the true positive rate (TPR) and precision (PRE). The formula for the F-measure maximized over all thresholds is:

$$\mathrm{F} = \max_{t \in [0,1]} \frac{(1+\beta^2) \times TPR(t) \times PRE(t)}{\beta^2 \times TPR(t) + PRE(t)}$$

β is a parameter that can be changed by the user to place more weight on either the true positive rate or precision. We decided to use a value of 1 for β. Both the true positive rate and precision are measured throughout the range of thresholds and applied to the equation. The value that is the largest becomes the official measurement for the attribute [118].

Odds ratio (OR) is another TBFS technique. Odds ratio is defined as the product of the number of true positives and the number of true negatives divided by the

product of the number of false positives and the number of false negatives. After applying the odds ratio metric across the range of thresholds the largest value is the recorded value of the feature [51].

Deviance (Dev) measures the sum of the squared errors from the mean class based on a threshold t [106]. Because deviance represents error, the minimum value over all the thresholds is the chosen value for the attribute.

Geometric mean (GM) is a quick and useful metric for feature selection. The equation for the geometric mean is the square root of the product of the true positive rate and the true negative rate. A geometric mean of one would mean that the attribute is perfectly correlated. The most useful feature of the geometric mean is the fact that not only does it maximize the true positive rate and the true negative rate but it also keeps them balanced, which is often the preferred state [106]. The maximum geometric mean across the thresholds is the score of the attribute.

$$GM = \max_{t \in [0,1]} \sqrt{TPR(t) \times TNR(t)},$$

where TNR is the true negative rate

The Kolmogorov-Smirnov statistic (KS) is a measurement of separability. The goal of KS is to measure the maximum difference between the distributions of the members of each class. The formula for KS is calculated as the absolute value of the true positive rate minus the false positive rate [97]:

$$KS = \max_{t \in [0,1]} |TPR(t) - FPR(t)|$$

Power (Pow) is very similar to KS in that it is the maximum distance between the curves of $1 - FPR$ and $1 - TPR$, where FPR is the false positive rate and TPR is the true positive rate. $1 - FPR$ is also known as the true negative rate, TNR, and $1 - TPR$ is also known as the false negative rate, FNR. The important difference between Power and KS is that there is an additional variable k, whose value

is assigned by the user (in this work we used $k = 5$); both the true negative rate and the false negative are raised to the power of k prior to finding their difference. [51]:

Probability ratio (PR) is a simple and convenient TBFS method. The ratio is defined as the ratio of the true positive rate to the false positive rate. In the end, this metric searches for the threshold that maximizes precision [51]. The equation is shown below.

$$PR = \max_{t \in [0,1]} \frac{TPR(t)}{FPR(t)}$$

Gini index (GI) was introduced by Breiman et al. [21] as an aspect of the CART algorithm. The Gini index is a measurement of how likely it is that an instance will be labeled incorrectly. An example of incorrect labeling is a positive value that was labeled as a negative. The equation for the Gini index is:

$$\text{GI} = \min_{t \in [0,1]} [2PRE(t)(1 - PRE(t)) + 2NPV(t)(1 - NPV(t))]$$

where $NPV(t)$ is the negative predictive value, or the percentage of instances predicted to be negative that are actually negative at threshold t. Since lower values here mean lower chances of misclassification, lower is better, and so the minimum Gini index score is the chosen score for the attribute [21].

Receiver Operating Characteristic, or ROC, curves are a graph of the true positive rate on the y-axis versus the false positive rate on the x-axis. This curve is created by mapping the points along the range of the thresholds. The curve itself represents the trade-off between the rate of detection and the rate of false alarms. In order to acquire a single numeric value for the purpose of ranking the predictive power of the attribute, the Area Under the ROC Curve (ROC) is measured and recorded. The larger the area, between zero and one, the more power the attribute has [26].

The Precision-Recall Curve (PRC) is a curve which plots the precision on the x-axis and the recall on the y-axis across the entire range of thresholds. The concept is very similar to that of the ROC curve. Like the ROC curve, it is the area under the

PRC curve that is used as a single numeric value for ranking purposes. The closer the area is to one, the stronger the predictive power of the attribute [97].

First Order Statistics Feature Selection Techniques

This section presents a set of seven univariate feature selection techniques which we have combined into a family of techniques we name First Order Statistics (FOS) based feature selection. This name was chosen because all seven techniques exhibit the use of first order statistical measurements such as mean and standard deviation. Although some of these techniques have been utilized in earlier papers, in 2012 our research group [70] combined them into a single family and studied their simlarity to each other, and how they perform in classification.

Fisher score [56] (FS) is a feature selection technique that selects each attribute independently according to their scores under the Fisher criterion. The FS score is calculated as:

$$FS = \frac{n_P[\mu_P - \mu_T]^2 + n_N[\mu_N - \mu_T]^2}{\sigma^2}$$

where n_P and n_N are the number of instances in the positive and negative classes respectively. Likewise, μ_P is the mean of the positive class and μ_N is the mean of the negative class. The variance σ^2 is the variance of the attribute across instances from both classes collectively. Lastly, μ_T is the average of the attribute over all instances.

Fold Change Difference (FCD) [66] uses the mean value of the attribute across all instances in the positive class and the mean value of the attribute across all instances in the negative class. FCD takes the difference between the mean of the attribute for the positive class to the mean of the attribute for the negative class.

The signal-to-noise ratio, or S2N, as it relates to classification or feature selection, represents how well a feature separates two classes. The equation for signal to noise is:

$$\text{S2N} = (\mu_P - \mu_N)/(\sigma_P + \sigma_N)$$

where μ_P and μ_N are the mean values of that particular attribute in all of the instances which belong to a specific class, either P or N (the positive and negative classes). σ_P and σ_N are the standard deviations of that particular attribute as it relates to the class. The larger the S2N ratio, the more relevant a feature is to the dataset [25]. We are one of the few groups that use S2N as a feature ranker and because of that we use our own implementation.

The Welch T-Statistic [103] (WTS) is a modified version of the t-statistic which does not assume equal variance with each of the classes. The modified equation is shown here:

$$WTS = \frac{\mu_P - \mu_N}{\sqrt{\frac{\sigma_P^2}{n_P} + \frac{\sigma_N^2}{n_N}}}$$

Wilcoxon Rank Sum [22] (WRS) is different from the standard t-statistic in that it makes no assumptions on whether or not the distribution is normal. The first step is to rank the all of the instances based on the value of the attribute. The next step is to take the sum of all of the rankings in the positive class which we will denote as W_P. Finally, the WRS is found as follows:

$$WRS = \frac{(W_P - \frac{n_P(n_P+1)}{2}) - \frac{n_P n_N}{2}}{\sqrt{\frac{n_P n_N (n_p+n_N+1)}{12}}}$$

Significance Analysis of Microarrays [103] (SAM) is a statistical technique to determine whether changes in attribute value (gene expression in the bioinformatics application domain) are statistically significant. The SAM technique identifies relevant attributes by performing attribute specific t-tests for each feature that measure the strength of the correlation between each independent feature and the class attribute. The equation of SAM is:

$$SAM = \frac{\mu_P - \mu_N}{\sigma^* + \sigma_0}$$

where σ_0 represents the exchangeability constant and σ^* represents an overall standard deviation. The role of σ_0 is to prevent attributes whose standard deviations are

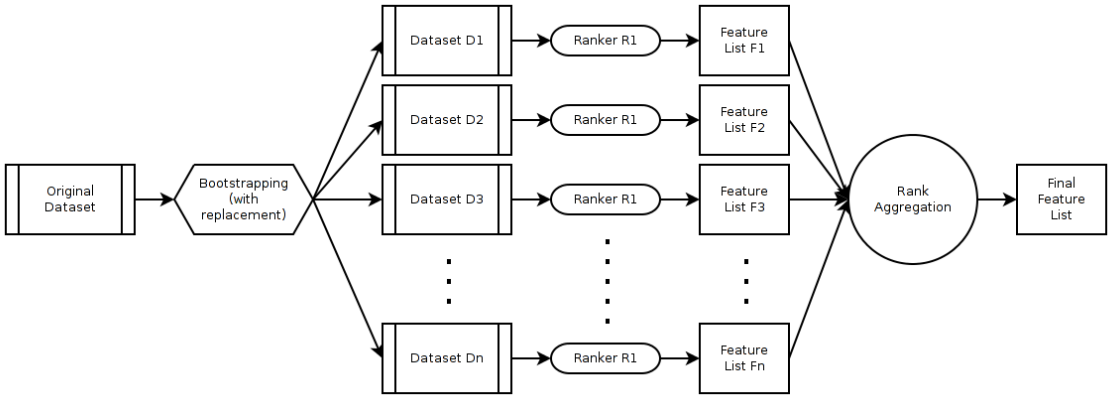
small from having large score. σ_0 is a customizable factor which is generally the top 90-percentile of standard deviations. For this experiment we use this value. σ is calculated as:

$$\sigma^* = \sqrt{\frac{n_T}{n_P n_N (n_T - 2)} \Big(\sum_{j=1}^{n_P} [x_j - \mu_P]^2 + \sum_{j=1}^{n_N} [x_j - \mu_N]^2 \Big)}$$

where $\sum_{j=1}^{n_P}$ and $\sum_{j=1}^{n_N}$ represent the sum across the instances of the positive class and the instances of the negative class respectively. Additionally n_T is equal to the total number of instances in the dataset.

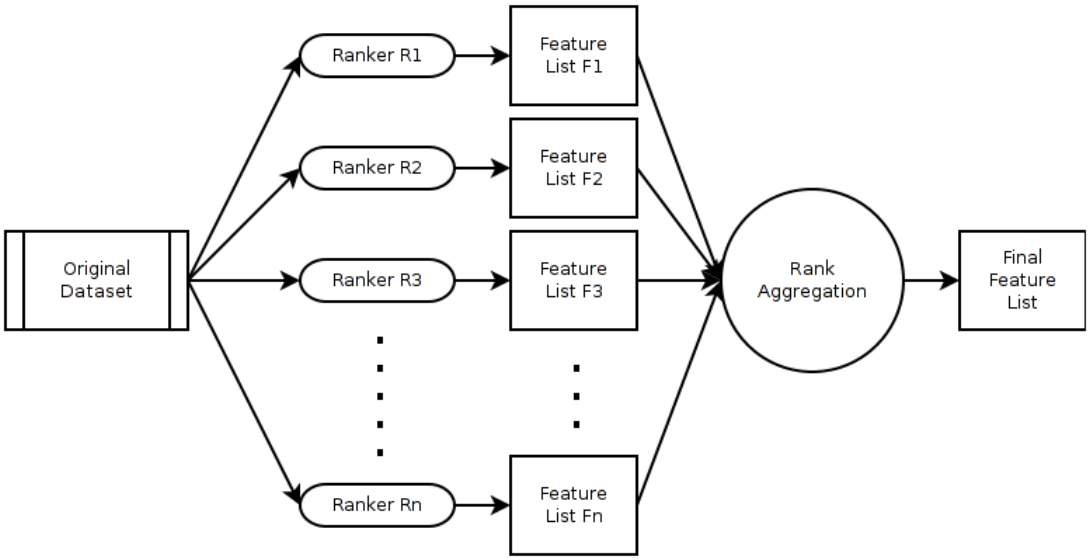
2.2.2 Feature Selection Ensemble Approaches

Figure 2.1: Ensemble: Data Diversity



At the core of creating an ensemble feature selection technique is how to approach the concept of ensemble [109]. Ensemble feature selection is a subset of feature selection techniques which applies feature selection algorithms multiple times and combines the results into one decision. The idea for ensemble feature selection is derived from ensemble learning methods wherein different classifiers are applied to a dataset and their results are aggregated [35]. We have decided upon comparing three approaches of ensemble: data diversity, functional diversity, and a hybrid approach

Figure 2.2: Ensemble: Functional Diversity

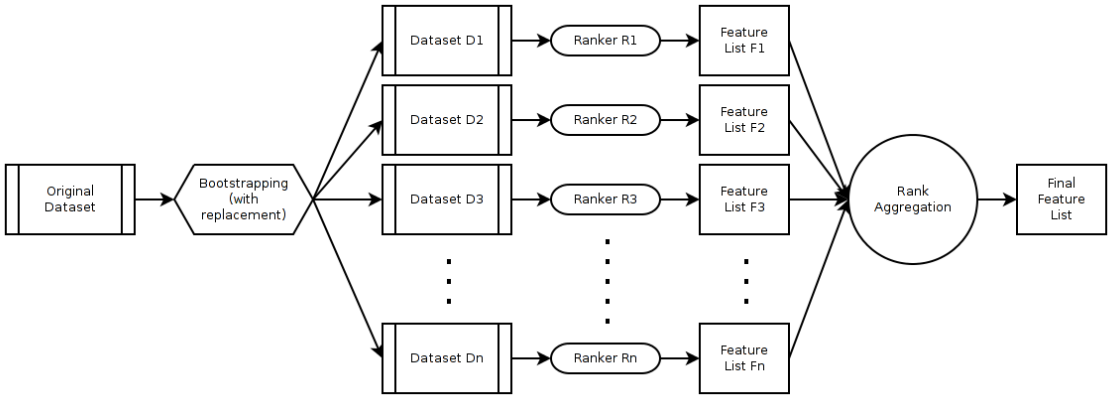


which combines data and functional diversity. This section explains the concept behind the approaches used in this study. Note that all three approaches as used in this work use feature rankers.

Data Diversity

Data diversity (see Figure 2.5), as its name suggests, achieves its diversity through the use of different sets of data. The process for data diversity occurs in three steps. The first step involves creating the different datasets in order to achieve the desired diversity. This can be achieved through the use of different compiled datasets which use the same set of features or, more commonly used, the creation of multiple sets of sampled data derived from the original dataset. The next step is to apply the same feature selection technique on each of these new datasets. Lastly, we aggregate the results from each of the datasets and end with a single feature subset for use in subsequent analysis. All recent works discussing the use of ensemble feature selection

Figure 2.3: Ensemble: Hybrid Diversity



techniques use data diversity when creating ensemble techniques [95]. It should be noted that outside of the number of iterations experiment, we decided to use bagging (sampling with replacement) [95] to create ten sets of sampled data (bags) from each of the twenty-six biomedical datasets used for both the data diversity and hybrid diversity approaches in Chapter 4. We do use the same process to create the different number of bags for the optimum number of iterations experiment also found in Chapter 4.

Functional Diversity

Functional diversity (see Figure 2.2) uses a completely different methodology than that of data diversity. In functional diversity, the same dataset is used throughout the entire process. The process of functional diversity takes the original dataset and applies a number of different feature selection techniques to create a ranked list for each technique. After all of the chosen techniques have been performed, the results are aggregated into a single feature ranking. To our knowledge, there has been no study which uses functional diversity within the domain of bioinformatics.

Hybrid Approach

The two previous techniques take vastly different paths in order to achieve their diversity. Both paths have their benefits and detriments when being implemented. However, using one of these approaches does not preclude the implementation of the other approach. Therefore we propose a hybrid methods which combines the ideas of these two approaches. The hybrid method (Figure 2.3) begins (like with data diversity) with the creation of the different datasets for use in the technique. The next step (as with functional diversity) will apply a different feature selection method to each of the different datasets. Finally, as with the previous two methods the results are aggregated into one ranked feature list. As with functional diversity, there has no study which uses hybrid methods within the field of bioinformatics.

Rank Aggregation Techniques

Many techniques are used throughout the literature to combine multiple ranked lists into one final product; these are referred to as rank aggregation techniques. In this work, we study nine such techniques: Mean, Median, Highest Rank, Lowest Rank, Stability Selection, Exponential Weighting, Enhanced Borda, Round Robin, and Robust Rank Aggregation. All of these techniques assume that the ranked lists being combined assign a value to each feature, from 1 to N (for N features), where the best feature is assigned number 1, the second-best feature is 2, and so on until the worst feature is assigned N. Unless otherwise noted, the ranked lists produced by the ensemble techniques will also assign values to each feature such that lower values are better [107].

While some rank aggregation techniques employ complex algorithms to assign values to the features based on their position in the various lists being combined, some are relatively straightforward [73]. Mean aggregation simply finds the mean value of the feature's rank across all the lists and uses this as that feature's value.

Similarly, Median aggregation finds the median rank value across all the lists being combined, using the mean of the middle two values if there are an even number of lists. Highest Rank and Lowest Rank use related strategies: either the highest (best, smallest) or lowest (worst, largest) rank value across all the lists is assigned as the value for the feature in question. In all cases, once each feature has been given a single value based on the mean, median, highest, or lowest value, all features are ranked based on these new values. Note that for all four of these it is possible for two features to end up tied, even if this was not the case in any of the lists being combined; this tie is resolved randomly if necessary [37, 38].

Stability Selection is based on a very simple principle: a feature is good if it appears towards the top of many of the ranked lists being aggregated. To this end, a threshold is chosen. This is often the threshold which will be used for selecting the features for downstream classification, but may be any appropriate value. For each list where the feature in question meets or exceeds that threshold, that feature is given a single point. For those lists where it fails to meet the threshold, it is given zero points. This calculation is performed for all of the ranked lists, and each feature is given all the points it deserves. Finally, the features are ranked from most to least points [62].

While Stability Selection performs its task of discovering those features which are most often towards the top of the list, it fails to account for how close to the top they are, and penalizes features which are just slightly under the threshold. A refinement upon this procedure is Exponential Weighting, which assigns points based on $e^{-r/s}$, where r is the feature's rank and s is the threshold being used. This satisfies the same goal as Stability Selection, while allowing for additional weight to be allocated as appropriate. As with Stability Selection, those feature which collect the most value are placed at the front of the final list [62].

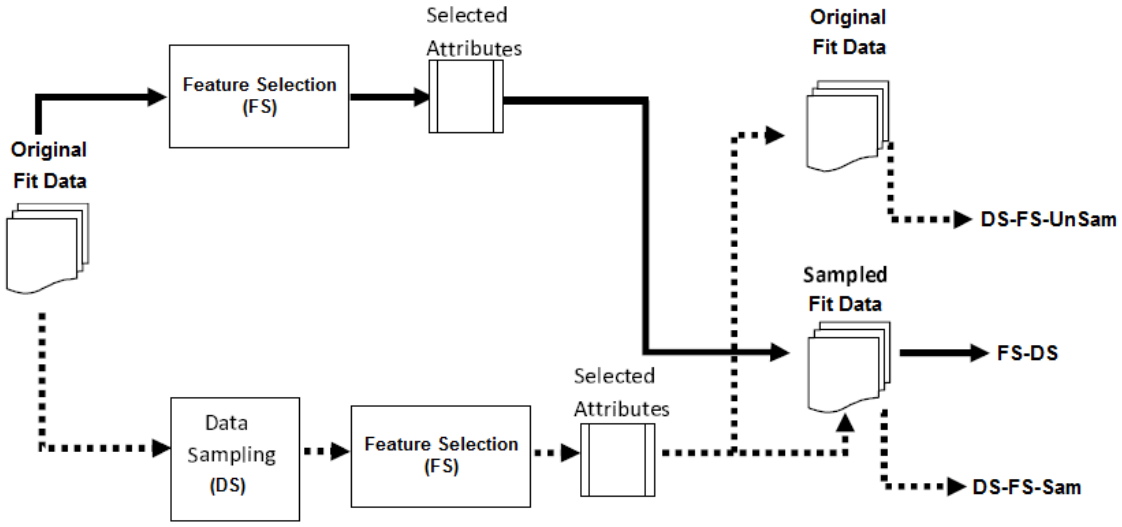
It has been shown that using the Borda count [7] as the basis of feature rank

aggregation is mathematically equivalent to simple Mean Aggregation. However, a variation known as Enhanced Borda expands upon the original method by multiplying each feature's Borda score ($\sum_L N - r_l$) by its Stability Selection score (number of times the feature meets or exceeds a specified threshold). This ensures that features which are frequently above the threshold get extra weight, while features also get more weight based on how high up they are on the list. Again, as with Borda (and Stability Selection), higher values are better in the results.

The Round Robin rank aggregation technique combines the lists in a fairly simple, although random, fashion: the lists themselves are randomly assigned an order, and then the first feature from the first list is chosen as the first feature for the final list, then the first feature from the second list, then the third list, and so on until the first feature from each list has been selected in order. After this, the second feature from each list (again examining the lists in the same order as before) are chosen. This proceeds until no features have not been found at least once. If while proceeding in this fashion a feature is found which is already on the final list, it is disregarded (and left in its existing, higher position) [89].

When combining ranked lists of features, one important question is how well each of the ranked lists performs compared to a randomly-sorted list. This is not an easy task, because it is not known in advance what the correct feature order is. However, if one assumes that most of the ranked lists are useful, and only a few are similar to the null (randomly-sorted) list, some progress can be made. The starting point of the Robust Rank Aggregation algorithm is examining how high a given feature scored on the various ranked lists. These values are collected into a so-called *rank order*, ordered from best to worst. If a feature is particularly useful, the predominance of these values should be towards the smaller (better) end, while only those ranked lists which are similar to the null list will give values that are randomly distributed along the range. Given each of these values, the algorithm finds the probability that if all

Figure 2.4: Data Sampling Approaches



the ranked lists were random, the values would be smaller (better) than is actually seen in the real ranking. It is expected that this probability will be small for good features, and so smaller values of this metric are better [77].

2.2.3 Feature Subset Size

After the rankings, whether they use an ensemble approach or not, the next step is to choose a subset of these features. In this case, twelve subsets are chosen per feature ranking. The sizes of the twelve subsets are as follows: 5, 10, 15, 20, 25, 50, 75, 100, 200, 350, 500, and 1000. These sizes are appropriate according to previous research [114]. It should be noted that subsets of feature subset sizes may be used and are discussed within the relevant chapters.

2.3 DATA SAMPLING

Data sampling is a data preprocessing that can be used to combat class imbalance. The process seeks to modify the dataset so as to have a more balanced class distribu-

tion. This is achieved by either the removal of instances from the majority class or the addition of instances to the minority class. Additionally, the modification process can be conducted randomly and the addition of instance can be duplicates of existing instances or synthetically created instances based on the existing ones. In this work we focus on three aspects of data sampling: the data sampling technique, the data sampling approach, and the post-sampling class distribution level.

2.3.1 Data Sampling Technique

In this work, we use three different data sampling techniques: random undersampling, random oversampling, and Synthetic Minority Oversampling TEchnique, or SMOTE [3, 42, 30]. Random undersampling (RUS) seeks to create balance between the two classes by reducing the size of the majority class. This is accomplished by randomly removing instances from the majority class until the desired class ratio has been achieved. Alternatively, random oversampling (ROS) seeks to improve the class balance by increasing the size of the minority class. The increase is performed through randomly duplicating instances from the minority class until the desired class ratio is achieved.

SMOTE is another form of oversampling which seeks to improve the balance between the two classes through the increasing the size of the minority class. However, unlike random oversampling, SMOTE does not duplicate instances. Instead SMOTE creates new minority instances using the original ones as a basis. It starts with an instance from the minority class and looks at a collection of its nearest neighbors (we use 5 neighbors in this work) and selects one at random. Once the neighbor has been selected, the differences between the two instances in terms of each feature is calculated. Finally a new instance is created by adding the product of the differences calculated and a random number between 0 and 1 to the original instance.

2.3.2 Data Sampling Approach

After selecting a data sampling technique the question remains: when should it be performed? Due to the high-dimensional nature of most bioinformtics datasets, feature selection is recommended to reduce the dimensionality of the dataset (for more information please see subsection 2.2.1). Therefore, there are two possible locations in which to perform data sampling: before or after the feature selection step. Additionally, if the data sampling is performed before the feature selection step, we have to choose whether to use the unsampled or sampled data to build the classification model. Therefore, there are three approaches to implementing data sampling. Figure 2.4 contains a visual description of how the three approaches are created. In the first approach (denoted as DS-FS:UnSam), data sampling is performed prior to feature selection and the unsampled data and selected features are used to build the classification models. The second approach (denoted as DS-FS:Sam) is similar to DS-FS:UnSam, with the difference that the sampled data is used for building classification models instead of the unsampled data. Lastly, the third approach (denoted as FS-DS) performs feature selection on the unsampled data and the selected features along with the sampled data are used to build the classification model [4].

2.3.3 Post-Sampling Class Distribution Ratio

As mentioned above, data sampling seeks to create a more balanced class distribution. While, logically a perfectly balanced class distribution seems ideal, for extremely imbalanced datasets you run the risk of too much data loss when using RUS because of the removal of instance from consideration for the model or overfitting with ROS and, to a lesser extent, SMOTE because of the addition of the duplicated or synthetic minority instances. Thus, a less aggressive post-sampling class distribution ratio may be beneficial to implement. As a result we decided to use two different final class ratios: 50:50 and 35:65. The two class ratios were chosen because 50:50 is the most

common final class ratio for data sampling and previous research has shown that 35:65 is an appropriate final class ratio [54, 72].

2.4 MEASURING STABILITY AND SIMILARITY

2.4.1 Dataset Perturbation

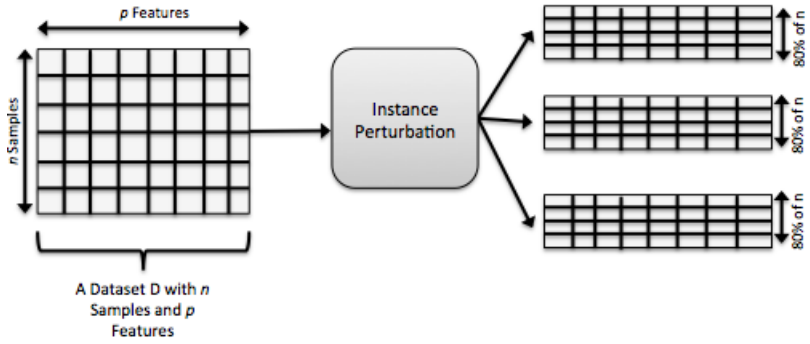
One of the key components in our research is to assess the stability of the feature rankers when changes are introduced to the dataset. In order to test this we removed a portion of the total number of instances and created a new data set. The process was the same for all of the datasets: we chose a fraction c of instances to keep and randomly removed $1 - c$ of the instances from both the majority and minority classes separately. Naturally, c was greater than 0 and less than 1. We removed from each class instead of just from the dataset as a whole in order to maintain the original level of class balance/imbalance for each dataset. For each c this process was repeated thirty times giving us thirty new datasets for each original data set and level of c. Following the creation of new datasets, feature selection was applied to both the original and the new datasets, and the chosen features were compared between each new dataset and the original dataset using our stability metric [44, 39].

The overall architecture of this approach is presented in Figure 2.5. Note that this figure illustrates three perturbed datasets, each with a c of 0.8. For our actual experiments, we chose four different levels of c, 0.95 (95%), 0.9 (90%), 0.8 (80%), and 2/3 (66.67%), and built 30 perturbed datasets for each level of perturbation.

2.4.2 Stability and Similarity Metric

We decided to use consistency index [80] to measure stability and similaritybecause it takes into consideration bias due to chance. First, we assume that a given dataset has n features. Let T_i and T_j be subsets of features, where $|T_i| = |T_j| = k$. The

Figure 2.5: Dataset Perturbation



consistency index [80] is obtained as follows:

$$I_C\left(T_i, T_j\right) = \frac{dn - k^2}{k\left(n - k\right)}, \tag{2.1}$$

where d is the cardinality of the intersection between subsets T_i and T_j, and $-1 < I_C\left(T_i, T_j\right) \leq +1$. The greater the consistency index, the more similar/stable the subsets are. However, in order to determine the similarity of the subsets we must determine which values of k (the feature subset size of both subsets) to use.

2.5 LEARNING ALGORITHMS

2.5.1 Classifiers

In this work, we use six diverse classifiers: 5-Nearest Neighbor (5-NN), Logistic Regression (LR), Naïve Bayes (NB), Multilayer Perceptron (MLP), Random Forest with 100 trees (RF100), and Support Vector Machine (SVM). All of the learners were chosen because they represent a variety of different classifiers. We provide only a brief discussion of these learners here because they are all well-understood classifiers. However, an interested reader may consult the provided references for more information. All models in this paper were built using the Weka data mining open-source software [58] with default parameter values unless otherwise specified. Note that any changes to default parameter values were applied when experimentation showed an overall improvement of the classification performance based on preliminary analysis.

5-Nearest Neighbor [118] is an implementation of the instance-based classifier K-Nearest Neighbor with the value of 5 for K. The classifier is used to predict the class of an unseen (new) instance by finding the five training-set instances closest to the test instance and having them vote on the class. The *"weight by 1/Distance"* parameter was used for this voting process. The prediciton of the new test instance will be the class with the largest total weight.

Logistic Regression [81] generates a simple logistic model using the training set which the model in turn uses to predict the class of new instances.

Naïve Bayes [118] is a Bayesian learner using Bayes's Theorem to approximate the posterior probability of an instance that belongs to a particular class, given its values for the different features. By using Bayes's rule and making the naïve assumption of conditional independence, this can be computed based on the individual probabilities of each feature value given each class.

Multilayer Perceptron [63] is a neural-network-based classifier which uses a simple feed-forward network along with back-propagation to train the network. In these experiments, we used a single hidden layer with three nodes (the $hiddenLayers$ parameter of the model was set to 3), and 10% (the $validationSetSize$ parameter of the model was set to 10) of the training data was held aside as a validation set so the classifier would know when to stop the iterative training process [104].

Random Forest [19] is an ensemble learner which builds a set of unpruned decision trees and then uses majority voting on the resulting trees to perform prediction. The trees themselves are built by bootstrapping (sampling with replacement) the training dataset and using an algorithm similar to C4.5 (J48 in Weka). In addition to using bootstrapping to randomize the training data, the features are also randomized by allowing only a small, randomly-determined subset to be used in the construction of each node of each tree. The number of trees used is determined by the user. Previous research [71] shows that the optimum number of trees is 100, so that is the number

used in this study.

Finally, Support Vector Machines [118] are linear discriminant classifiers using linear kernels. They are based on the idea that for binary classification, there should exist a hyperplane through feature-space which divides the instances into two sections, one for each class. The best such hyperplane would be the one which maximizes the distance between the hyperplane and members of each class. For our models, the complexity constant *"c"* was set to 5.0 and the *"buildLogisticModels"* parameter set to "true."

2.5.2 Ensemble Learning

In this work we utilize two different ensemble learning approaches: Select-Bagging and Select-Boosting. Both of these techniques incorporate the feature selection process into their respective algorithms. This is an important distinction because both Bagging and Boosting (when using Boosting by resampling discussed later in this section) creates new training datasets with each iteration of their algorithms. Therefore, any feature selection performed before the ensemble approach will not be as valid with the new training datasets. Despite this, studies have performed the feature selection process either before the ensemble methods [23] or even before the cross-validation process (if one is applied) [102]. As a result we developed Select-Bagging and Select-Boosting to apply the feature selection process on each new training dataset generated by their algorithms. Both the Select-Bagging and Select-Boosting processes were implemented by our research group in the Weka data mining toolset [118]. Each ensemble approach uses 10 iterations.

Bagging was developed in 1996 [18] in order to improve the results of unstable single-run classifiers. The basic principle of bagging is to take a random sampling of N instances from the dataset with replacement. The N instances are then split into two datasets: the first is a reduced dataset which removes all of the duplicate

instances and the other contains the duplicates that were removed. The first dataset is used to train the classifier and the second is used to determine the classification performance. The process is repeated a number of times R and results in R models. The classification of a new instance is based on classifying the case using all R models and the majority vote is the final decision. Bagging can be applied to any single-run classifier though is frequently applied toward decision trees. Though the process of sampling with replacement is can be used as a step in creating ensemble classifiers [33, 122], primarily, bagging is used as a benchmark ensemble classifier when testing new techniques [24, 57, 91, 123].

Select-Bagging [31] (see Figure 2.6) incorporates feature selection into the process of Bagging by performing feature selection after the sampling with replacement for every iteration of the Bagging algorithm. After feature selection, a classifier is trained and the process is repeated the predetermined number of times. The final decision for Select-Bagging like with the Bagging algorithm is decided by taking the average of the posterior probabilities of the membership of the instance for the positive class from the collection of classifiers and using that average to make the final decision.

Boosting [52] is a method for improving the performance of weak classifiers. The general process of boosting consists of applying a weight to each instance ($1/n$ where n is the number of instances) and performing classification with the dataset. The weights are modified based on a pseudo-loss function, such that instances which are correctly classified will have their weight reduced while those which are incorrectly classified will have their weight increased. After all of the weights are normalized the process is repeated, each model being built using the weights developed throughout the course of the algorithm. This will occur a number of times, and the majority votes of the models from each iteration (weighted based on their overall performance) are used to find the final classification result. Although the weights are varied for each round, boosting does not guarantee the diversity as strongly as many other forms of

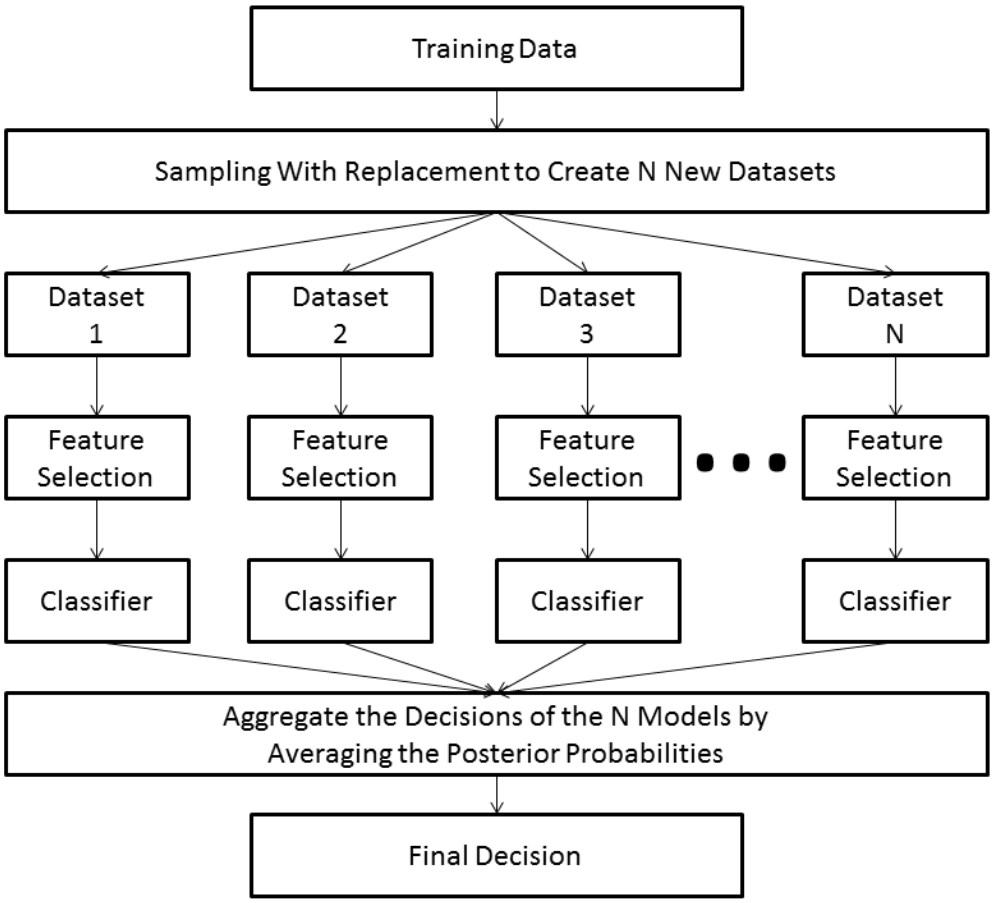
ensemble learning, and steps must be taken to ensure this diversity [57]

Select-Boosting [43] (see Figure 2.7), like Select-Bagging, incorporates the process of feature selection into the algorithm after the new training datasets are generated. However, in order to vary the training datasets for the feature selection process we used the Boosting by resampling option of the AdaBoost algorithm implemented in the WEKA data mining toolset. Boosting by resampling (activated by the "useResampling" option in WEKA being set to true), as opposed to Boosting by reweighting (as mentioned in the previous paragraph), resamples the training data based on the instance weights generated by that iteration. It should be noted, that the first iteration resamples based on the initial weights. As a result, a new training dataset is generated that is the same size as the original training dataset, with instances with high weights occurring more frequently than those with low weights. It is through this overrepresentation of the high weight instances and under representation of the low weight instances that the new weights are reflected. After feature selection is performed, a classifier is trained and is given a weight parameter and the process repeats for the predetermined number of iterations. The final decision of the Select-Boosting algorithm is the same as that of the AdaBoost algorithm: a weighted average of the posterior probabilities.

2.6 CROSS VALIDATION AND PERFORMANCE METRIC

Cross-validation [76] is the process of dividing the original dataset into N approximately equal-size partitions (folds), building the model using $(N-1)$ of these folds, then testing the built model using the Nth fold. This process is repeated N times so that each fold is used $(N-1)$ times to build the models and used only once to test the built model. The advantage of N-fold cross-validation over random sub-sampling is that all instances are used for both training and testing, and each instance is used only once per run for evaluating purposes. In this study, we used four runs of five-fold

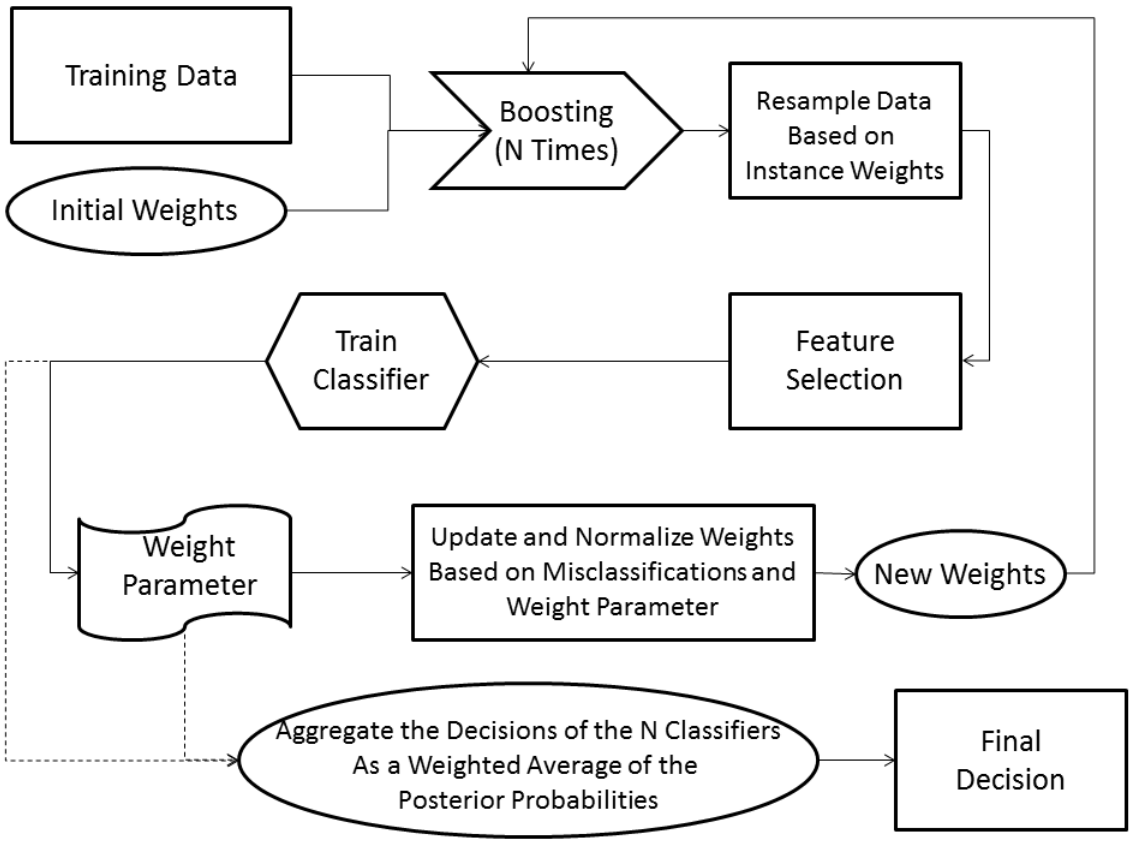
Figure 2.6: Select-Bagging



cross-validation to reduce any bias due to a chance split. In addition, we performed data sampling and feature selection for every training dataset generated by the cross validation process.

Due to the imbalanced nature of some of the datasets, we use the Area Under the Receive Operating Characteristic Curve (AUC) [49] to assess the performance of all classification models. The AUC plots the curve of the True Positive Rate (TPR) versus the False Positive Rate (FPR) across all decision boundaries. The area under the curve represents the quality of the model. It should be noted that the AUC described here is different from the ROC feature selection technique mentioned in

Figure 2.7: Select-Boosting



Section 2.2.1. To prevent any confusion, we use the notation AUC for the classification metric and ROC for the feature ranking technique.

CHAPTER 3
EFFECTS OF DATA CHARACTERISTICS ON THE STABILITY OF GENE SELECTION IN BIOINFORMATICS

3.1 INTRODUCTION

Identifying important genes in the human body has long been a major goal of biology and genomics. One of the more common methods of finding these important genes is through studying the levels of the messenger RNA (mRNA) that are derived from these genes. This is due to the fact that the mRNA that is present is what determines how all of the proteins are created within the cell. A recent chemical and technological advancement has created a method to test the levels of thousands of different mRNA sequences simultaneously: DNA microarrays. The DNA microarray is a grid of gene probes that are made from the complementary DNA (cDNA) sequences of the different mRNAs. Due to the ability of mRNA to readily bind to its corresponding cDNA, it is relatively easy to measure the amount of mRNA that is attached to the gene probes. The more mRNA attached to the probe, the stronger the influence of the gene the probed was designed from. Unfortunately, while the ability to test a sample for thousands of genes is an important advancement, the sheer amount of data created does give rise to the problem of high dimensionality [88].

High dimensionality is, at it's core, a problem of having too much data to easily work with. Specifically high dimensionality is when there are a very large number of attributes attached to each sample. With so much data to work with it can be very difficult and time consuming to distinguish what is important and what is not. A common method of dealing with this problem is a series of techniques called feature

selection. Feature selection is a method of selecting a smaller subset of the attributes (features) available and analyzing only those attributes. The use of feature selection can greatly reduce computation time, identify and remove irrelevant and redundant features, and possibly improve classification performance. Though data is lost during feature selection, it can assist with creating more efficient and accurate classifiers.

Normally when one wants to describe the effectiveness of a particular feature selection technique in comparison to other techniques one compares the performance of the classifier built using the subsets chosen by the techniques [45, 119]. However, another method of describing how well a feature selection technique performs is though studying its robustness or stability. By using these robust feature selection methods, feature subsets remain relatively unchanged even when there are changes in the data and the researcher can be more confident in the importance of any features chosen by the technique [108].

In order to measure the stability of a feature selection technique, we compare the feature subsets chosen by that technique for a dataset to the feature subsets chosen by the same technique on a version of the same dataset which has had instances removed. Alternatively, this can also be viewed as before and after instances are added. This is unlike the typical approach of comparing the selected subsets from independent random subsamples of the original dataset, and is thus unique to this paper. This comparison is repeated using twenty-six different real world datasets and eighteen feature rankers including six commonly used feature selection techniques, eleven Threshold Based Feature Selection (TBFS) techniques, and a relatively new technique called Signal to Noise.

The stability of a feature selection technique (and by extension its chosen feature subset) can be affected by factors other than changes to the dataset. We decided to test the effects of a number of factors inherent to the data itself: balance of the classes, quality of the data, and the number of instances and attributes. The balance of the

classes is percentage of how many instance belong to each of the classes. Difficulty-of-learning is a measurement of how difficult it is for a classifier to effectively classify the instances [39]. Lastly, the number of instances and attributes in a dataset are the total number of instances contained in a dataset and how many attributes each instance has. The twenty-six datasets represent a wide range for each of these factors. Additionally, we also examined the effects of the feature subset size has on stability.

3.2 CONTRIBUTIONS

The primary contribution of this paper is an in-depth look at how different data factors can affect the stability of feature ranking techniques within the domain of bioinformatics. Although previous work exists for some of these factors, no study has examined how each can influence the stability of feature ranking, using 18 feature ranking techniques and 26 real-world datasets to validate the results. This large number of data factors, feature selection techniques, and datasets is not found in any other single study. This, work is split into two sase studies: one which looks at a large number of data factors and the other which focuses on the factor of difficulty-of-learning.

The rest of the paper is organized as follows: Section 3.3 contains discussions of previous research that are relevant to our work. In section 3.4, all the details involving our experimental proceedure. In Section 3.5, we present our results of the two case studies along with discussions of our findings. Finally, in Section 3.6, we present our conclusions and potential avenues of future study.

3.3 RELATED WORKS

The use of DNA microarrays has gained a large amount of attention within the domains of biology, genetics, and bioinformatics. There are a number of different applications of this technology within these domains. One of the more promising

Table 3.1: Dataset List

Difficulty of Learning	Name	Level of Balance
Easy	DLBCL	Balanced
	BCancer50k	Balanced
	ovarian-cancer	Slightly Imbalanced
	Prostate MAT	Slightly Imbalanced
	MLL Leukemia	Slightly Imbalanced
	ALL AML Leukemia	Slightly Imbalanced
	Colon50k	Slightly Imbalanced
	CNS MAT	Slightly Imbalanced
	Lung	Slightly Imbalanced
	Lung Michigan	Imbalanced
	Lung Cancer	Imbalanced
	Lymphoma MAT	Imbalanced
	Lymphoma	Imbalanced
	Lung50k	Imbalanced
	Acute Lymphoblastic Leukemia	Imbalanced
Moderate	Prostate	Balanced
	Colon	Slightly Imbalanced
	Lung Cancer Ontario	Slightly Imbalanced
	Ovarian MAT	Imbalanced
	Brain Tumor	Imbalanced
Hard	Breast Cancer	Balanced
	Mulligan-R-NR	Balanced
	DLBCL NIH	Balanced
	Mulligan-R-PD	Slightly Imbalanced
	Central Nervous System	Slightly Imbalanced
	ECML Pancreas	Imbalanced

applications of microarrays is disease prediction. One example is a paper written by Abeel et al. [1] which uses DNA microarrays to identify biomarkers to be used in cancer diagnosis. Disease prognosis is another possible application of the DNA microarray in recent years. The papers written by Fan et al. [48] and Abraham et al. [2] are but two papers in which DNA microarray data has shown promise in predicting patient prognosis with cancer.

One of the drawbacks of using DNA microarrays for creating gene expression profiles is that the original dataset contains a large number of genes, many of which will be irrelevant to the problem at hand. One solution to this is feature selection. Feature selection is a process which reduces the number of features and only retains the top features. The effectiveness of feature selection has been studied across many

domains, including gene microarray analysis. One study looking at this application was done by Liu et al. [82], which tested whether twenty features selected through various feature selection methods would outperform twenty randomly selected features when used to create a model. In every test the features chosen by the feature selection methods outperformed the randomly-selected features. An excellent review of feature selection within the domain of bioinformatics was performed by Saeys et al. [96]. This review focuses on the various applications feature selection can be applied to, including sequence analysis, protein prediction, signal analysis, mass spectrometry, single nucleotide polymorphism analysis, etc.

We decided to include a number of datasets with varying levels of class imbalance in order to properly test the stability of the rankers. Class imbalance is a problem where the numbers of instances in all of the classes are not equal. This is a problem that appears frequently when using DNA microarray datasets due to there being few instances of the class of interest. One study examining the effects of imbalanced data on bioinformatics was done by Blagus et al. [15]. They found that using classification on imbalanced datasets creates bias towards the majority class. They also state that while minor cases of class imbalance can be corrected using methods like undersampling (removal of samples) or oversampling (duplication of samples), more complex approaches must be taken for more severe cases of class imbalance. Kamal et al. [68] performed another study looking at how class imbalance affects classifier performance. They found that gene selection (feature selection) with certain classifiers will increase the bias towards the majority class, even as it increases overall accuracy. Their recommendation is to work with measures that select genes useful to the minority class (class of interest) instead of focusing on increasing overall accuracy.

Difficulty-of-learning [40] is a measurement of how challenging a dataset is for inductive models (that is, how well models tend to perform on the dataset). By taking the average classification performance across a series of learners, we are able

to determine if a dataset is particularly easy or hard to learn from. This measurement is important because the difficulty of a dataset will directly affect the performance of any experiment performed on the dataset. For example, if a dataset is easy to learn from, then the classification performance will be skewed higher regardless of the other factors involved (ranker, learner, etc.). Alternatively, if a dataset is particularly difficult then there is more room for improvement, and the experimental decisions will have a larger effect on the performance. Thus, difficult-to-learn datasets are excellent for comparing different techniques.

One of the more commonly used methods of measuring the robustness of a feature selection technique is to calculate the similarity of multiple feature subsets derived from the same technique using randomly selected instance subsets of the original dataset as the input [80, 83]. While this provides the general outline for evaluating stability, a metric is still needed to compare the resulting feature sets. A work written by Saeys et al. [95] used the Spearman rank correlation coefficient. A paper by Abeel et al. [1] focused on selecting biomarkers from DNA microarray data and creating a framework of stability analysis. In their work Lustgarten et al. [84] devised a new stability measure called Adjusted Stability Measure (ASM) that can be applied to classifier based feature selecting methods. The paper by Kalousis et al. [67] used different measures of correlation to measure the stability of the feature ranker. Stiglic et al. [100] wrote a paper comparing the stability of ranked gene lists, and found that univariate selection methods create more stable feature subsets.

To evaluate the stability of feature selection, a metric is needed to define the similarity of two feature subsets. In 2007, Kuncheva et al. [80] devised a framework to study the stability of feature selection methods by building randomly-selected instance subsets of the original data and comparing feature subsets chosen from these (using the same feature selection technique throughout). To perform this comparison, the authors developed the consistency index, a measure of similarity between two

different feature subsets. They use this measure as a way to choose the best set of features for an experiment. In the present work, the consistency index is used to determine which feature rankers are the most stable, comparing the chosen features from both the original unmodified dataset and a reduced dataset which has had some instances randomly removed.

3.4 METHODOLOGY

In this chapter, we use a collection of 18 feature rankers. The feature rankers used are: Dev, F, GM, GI, GR, IG, KS, MI, OR, Pow, PR, PRC, RF, RFW, ROC, S2N, SU, and CS. The All Factors study uses all 18 feature rankers and the Difficulty-of-Learning Study uses Dev, IG, Pow, PR, ROC, and S2N. All feature rankers and their feature subset sizes are discussed in Section 2.2.1. In order to measure stability, we use the dataset perturbation process along with consistency index outlined in Section 2.4. Lastly we use a series of 26 bioinformatics (see Table 3.1) to test the stability of the feature rankers. The particulars of all of the datasets can be found in Tables 2.1 and 2.2. As difficulty-of-learning and class balance are factor we examine in this work we created categories for both factors. We created three groupings for difficulty-of-learning based on the Average AUC value of each dataset: Easy ($\geq$ 0.8, fifteen datasets), Moderate ($<$ 0.8 and $\geq$ 0.7, five datasets), and Hard ($<$ 0.7, six datasets), based on their average AUC value as described in Section 2.1. We used three different levels of class balance: balanced ($>$ 40% minority class), slightly imbalanced ($\geq$ 26% and $\leq$ 40% minority class), and imbalanced ($<$ 26% minority class) based on the minority class percentage described in Section 2.1.

3.5 RESULTS

In this section we will be presenting two case studies regarding feature selection stability. The first looks at all of the data characteristics and their effect on the

stability of generated feature subsets. The second focuses specifically on how the difficulty-of-learning affects stability.

3.5.1 All Factors Study

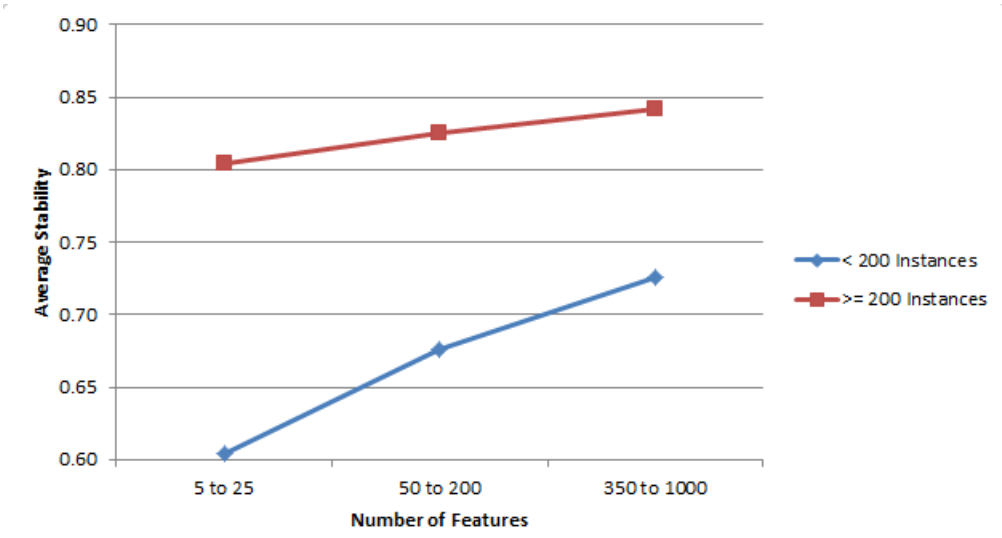
The experiment was conducted using eighteen feature selection techniques on twenty-six DNA microarray datasets. There are a number of factors of the data and aspects of the feature selection which can be examined to observe their effects on stability: degree of perturbation, class balance of dataset, number of features used, choice of ranker, number of instances, number of attributes, and difficulty-of-learning. To demonstrate how stability changes over the ranges for each factor, thresholds were chosen and the results summarized for each grouping. These thresholds (which will be discussed in detail for each factor) were chosen based on preliminary work and inspection of our datasets, and may not be appropriate for all collections of datasets.

Tables 3.2 through 3.7 and Figures 3.1 through 3.5 contain the results of our stability experiments. Each table focuses on one of the following six factors: degree of perturbation, class balance of dataset, number of features used, number of instances, number of attributes, and difficulty-of-learning. The tables will present the results across three different ranges of feature subset sizes as well as the view across all feature subset sizes used. The three different ranges for the number of features used are: 5 to 25, 50 to 200, and 350 to 1000. We chose these ranges because 50 to 100 is a recommended range for the number of features used in data mining in bioinformatics [114]. We decided to add 200 as a feature set size in order to have a more thorough range. We also use the ranges 5 to 25 and 350 to 1000, to demonstrate the effects of higher and lower numbers of features on stability. Each table shows the results of each of the eighteen feature selection techniques. Within each column, the feature ranker which had the highest stability value is printed in bold. The figures show the overall trends that occur with each data factor. In order to show this, we calculated

the average stability across all eighteen feature rankers at each combination of level of data factor and size of the feature subset. We begin the results with those factors that are inherent to the datasets themselves.

The number of instances in a dataset not only improves the statistical reliability of the data but also the stability of the feature selection techniques. Table 3.2 contains the average results of two different groups of datasets: datasets with less than 200 instances and datasets with more then 200 instances. Among the twenty-six datasets used in this paper, nineteen of them had less than 200 instances and had an average of 92.37 instances across all of the datasets. The remaining seven datasets contain more than 200 instances and the group has an average of 317.57 instances. Figure 3.1 shows the overall trend averaged across all eighteen rankers. The table and the figure show that as the number of instances in the dataset increases, stability increases.

Figure 3.1: Stability: Number of Instances



The number of attributes, like the number of instances in the data, affects how stable a feature selection method is. Table 3.3 shows the average stability across all datasets with greater than or equal to 10,000 attributes and the average stability across all datasets with fewer than 10,000 features. We chose this split to maximize

Table 3.2: Average Stability For Number of Instances

Ranker	< 200 Instances				≥ 200 Instances			
	Feature Subset Sizes				Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall
CS	0.55841	0.65395	0.72181	0.63110	0.81584	0.82631	0.87194	0.83336
Dev	0.62613	0.70337	0.74006	0.68036	**0.89076**	**0.90696**	**0.89993**	**0.89845**
F	0.62135	0.69846	0.74304	0.67748	0.85909	0.86443	0.85683	0.86031
GI	0.55741	0.60707	0.65604	0.59862	0.82143	0.80146	0.77751	0.80380
GM	0.63931	0.71129	0.75252	0.69161	0.84205	0.85488	0.85638	0.84991
GR	0.54344	0.58981	0.69101	0.59579	0.75905	0.73772	0.79188	0.76015
IG	0.54896	0.65493	0.72602	0.62855	0.79348	0.81792	0.86719	0.82006
KS	0.63682	0.71069	0.75011	0.68976	0.84206	0.85441	0.85586	0.84963
MI	0.59388	0.69180	0.73538	0.66189	0.84521	0.85202	0.84885	0.84839
OR	0.56980	0.64326	0.69105	0.62460	0.70213	0.76455	0.78784	0.74437
Pow	0.57599	0.65199	0.70091	0.63255	0.76742	0.81640	0.84627	0.80346
PR	0.55497	0.61179	0.66400	0.60117	0.74771	0.75574	0.75371	0.75188
PRC	0.66760	0.73700	0.77197	0.71683	0.83759	0.87275	0.87628	0.85898
RF	0.63103	0.69419	0.72491	0.67556	0.74405	0.80301	0.83304	0.78595
RFW	0.61869	0.67896	0.71418	0.66265	0.74283	0.79263	0.81344	0.77708
ROC	**0.66978**	**0.74192**	**0.77911**	**0.72116**	0.83108	0.86143	0.87004	0.85094
S2N	0.66826	0.73983	0.77761	0.71946	0.84165	0.86314	0.87792	0.85788
SU	0.58593	0.65112	0.71739	0.64052	0.79147	0.81752	0.87086	0.82000
Average	0.60376	0.67619	0.72540	0.65831	0.80416	0.82574	0.84199	0.82081

the separability between the twenty-six datasets. There are fourteen datasets with greater the or equal to 10,000 attributes per instance, and combined these datasets have an average of 24,383.21 attributes per instances. The group with less than 10,000 attributes per instances has 12 datasets, with a combined average of 5,665.67. Figure 3.2 summarizes the same information in a simpler-to-read format. The table and figure show that, with few exceptions, the more features a datasets has, the more stable the feature selection methods are.

The balance of the classes is an important factor to consider when discussing a dataset. The distribution of the instances between the two classes affects the stability

Table 3.3: Average Stability For Number of Attributes

Ranker	$< 10{,}000$ Attributes				$\geq 10{,}000$ Attributes			
	Feature Subset Sizes				Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall
CS	0.55482	0.64527	0.76747	0.63813	0.69020	0.74757	0.75878	0.72621
Dev	0.64474	0.72443	0.77123	0.70293	0.74249	0.78711	0.79294	0.77006
F	0.61984	0.70158	0.75172	0.68006	0.74151	0.77877	0.79039	0.76668
GI	0.56634	0.63345	0.68913	0.61941	0.68177	0.68165	0.68561	0.68339
GM	0.63148	0.70880	0.75388	0.68785	0.74739	0.78522	0.79974	0.77397
GR	0.53775	0.60765	0.75199	0.61461	0.65612	0.64847	0.67908	0.66183
IG	0.54725	0.65486	0.77241	0.63941	0.67269	0.73649	0.75547	0.71499
KS	0.62810	0.70783	0.75259	0.68580	0.74691	0.78500	0.79813	0.77310
MI	0.58160	0.68169	0.73857	0.65421	0.73007	0.78058	0.78853	0.76173
OR	0.55778	0.64440	0.70185	0.62267	0.64627	0.70293	0.72613	0.68614
Pow	0.57031	0.65118	0.70799	0.63169	0.67658	0.73489	0.76437	0.71875
PR	0.56916	0.63089	0.68988	0.61992	0.63918	0.66739	0.68310	0.66046
PRC	0.65772	0.73595	0.77626	0.71343	**0.76106**	**0.80578**	0.81877	**0.79082**
RF	0.63816	0.69731	0.72562	0.67974	0.68143	0.74592	0.77481	0.72716
RFW	0.63273	0.68433	0.70817	0.66879	0.66872	0.73119	0.76656	0.71460
ROC	0.66657	0.74156	0.78331	0.72075	0.75319	0.80198	**0.81927**	0.78640
S2N	**0.66951**	**0.75929**	**0.80028**	**0.73213**	0.75388	0.78481	0.80599	0.77780
SU	0.57901	0.65220	0.76609	0.65018	0.69463	0.73339	0.74973	0.72199
Average	0.60294	0.68126	0.74491	0.66454	0.70467	0.74662	0.76430	0.73423

of the feature rankers. Table 3.4 and Figure 3.3 show that with few exceptions, as the balance between the classes decreases, stability increases.

There is one final factor that is inherent in the datasets themselves, difficulty-of-learning. Table 3.5 shows that, with few exceptions, as the difficulty-of-learning increases (or alternatively the quality of data decreases), stability decreases. Even the few exceptions only reverse this trend by less than 0.015. The overall effect follows this trend and can be seen in Figure 3.4.

Feature subset size, dataset perturbation, and choice of ranker are not factors that are inherent to the dataset itself, but do have an effect on stability. Feature subset

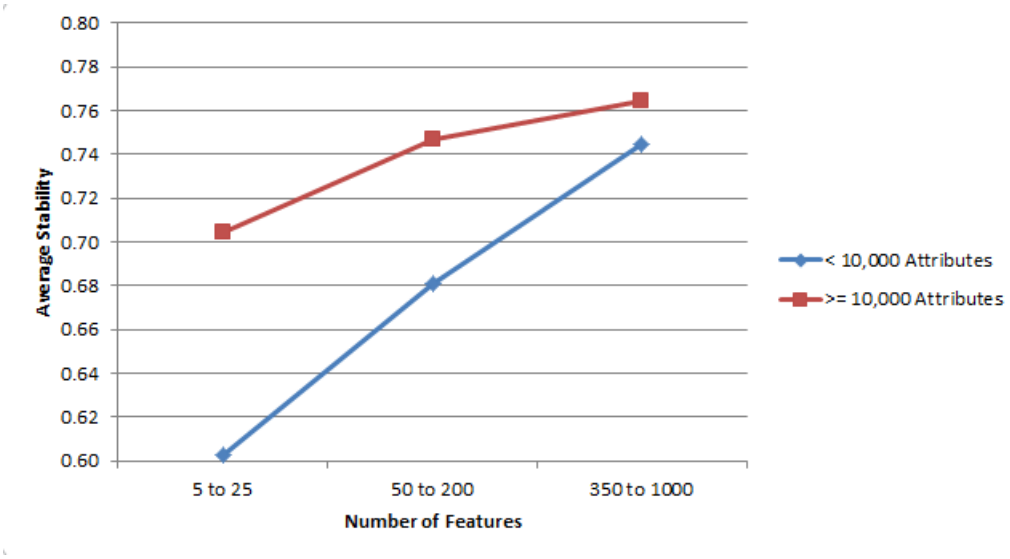
Table 3.4: Average Stability For Class Balance

Ranker	Balanced				Slightly Imbalanced				Imbalanced			
	Feature Subset Sizes				Feature Subset Sizes				Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall
CS	0.57286	0.64497	0.73502	0.63744	0.61388	0.68599	0.77545	0.67831	0.68120	0.75483	0.76421	0.72650
Dev	**0.70018**	**0.76397**	**0.78308**	**0.74217**	0.68132	0.72931	0.77341	0.72034	0.71513	0.78962	0.79498	0.75992
F	0.66729	0.70160	0.72676	0.69359	0.66833	0.72716	0.77363	0.71426	0.71821	0.79037	0.80502	0.76397
GI	0.58824	0.61643	0.65391	0.61406	0.62124	0.67364	0.71846	0.66301	0.66418	0.67065	0.67566	0.66921
GM	0.68170	0.70804	0.74669	0.70673	0.68622	0.74245	0.77573	0.72734	0.71140	0.78706	0.80882	0.76097
GR	0.54251	0.57322	0.69853	0.59175	0.60247	0.63724	0.72686	0.64516	0.63960	0.65794	0.72063	0.66597
IG	0.54699	0.63492	0.73412	0.62308	0.60658	0.68500	0.77442	0.67468	0.67004	0.75829	0.77125	0.72476
KS	0.67111	0.70529	0.74247	0.70034	0.68693	0.74242	0.77429	0.72727	0.71235	0.78728	0.80791	0.76122
MI	0.64503	0.70079	0.72161	0.68276	0.67090	0.71908	0.76574	0.71067	0.66113	0.77708	0.79571	0.73342
OR	0.51330	0.58667	0.65644	0.57354	0.59851	0.67128	0.72538	0.65449	0.67530	0.74107	0.74745	0.71526
Pow	0.59194	0.65169	0.69983	0.63883	0.61351	0.69222	0.74585	0.67283	0.66839	0.73090	0.75975	0.71207
PR	0.56098	0.60312	0.66099	0.60003	0.59691	0.66220	0.71557	0.64834	0.64962	0.66791	0.67276	0.66150
PRC	0.67460	0.73267	0.76355	0.71619	0.71329	0.76408	0.80417	0.75294	0.73931	0.81238	0.81936	0.78368
RF	0.57671	0.64640	0.70213	0.63129	0.67347	0.74378	0.76897	0.72078	0.70328	0.75008	0.77036	0.73565
RFW	0.58452	0.63391	0.68570	0.62628	0.65937	0.71660	0.74283	0.69931	0.68829	0.75139	0.77535	0.73109
ROC	0.67679	0.72561	0.76908	0.71614	0.70438	0.76815	0.80177	0.74999	**0.74828**	**0.81367**	**0.82881**	**0.79021**
S2N	0.66517	0.73556	0.77532	0.71617	**0.71837**	**0.77796**	**0.81141**	**0.76149**	0.74393	0.79199	0.81584	0.77793
SU	0.57090	0.62732	0.73014	0.62952	0.63820	0.68505	0.77170	0.68719	0.69193	0.75492	0.76188	0.73041
Average	0.61282	0.66623	0.72141	0.65777	0.65299	0.71242	0.76365	0.70047	0.69342	0.75486	0.77199	0.73354

Table 3.5: Average Stability For Difficulty-of-Learning

Ranker	Easy				Moderate				Hard			
	Feature Subset Sizes				Feature Subset Sizes				Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall
CS	0.72036	0.78461	0.80539	0.76303	0.53700	0.62345	0.70191	0.60704	0.45865	0.53988	**0.70512**	0.54734
Dev	0.76342	0.81567	0.83641	0.79908	0.63650	0.70336	0.72515	0.68095	**0.57228**	**0.65151**	0.69275	**0.62881**
F	0.76964	0.81841	0.84099	0.80373	0.62302	0.70173	0.73387	0.67697	0.50730	0.56705	0.61952	0.55527
GI	0.72663	0.74296	0.73949	0.73529	0.51644	0.55787	0.63694	0.56037	0.46854	0.53059	0.59871	0.52177
GM	0.76948	0.82480	0.84100	0.80580	0.64926	0.70890	0.74082	0.69203	0.52068	0.57467	0.64653	0.57014
GR	0.70163	0.71326	0.74505	0.71636	0.51530	0.54411	0.67545	0.56494	0.40449	0.48135	0.68880	0.50119
IG	0.70490	0.78591	0.80814	0.75771	0.53261	0.62093	0.70301	0.60465	0.44308	0.53097	0.70489	0.53783
KS	0.76980	0.82460	0.84094	0.80585	0.64739	0.70844	0.73864	0.69055	0.51254	0.57285	0.63947	0.56438
MI	0.74044	0.80832	0.83485	0.78667	0.60425	0.69163	0.71759	0.66171	0.49363	0.56676	0.61716	0.54889
OR	0.69068	0.75953	0.78500	0.73721	0.54174	0.61662	0.68162	0.60167	0.42611	0.49622	0.55603	0.48196
Pow	0.69309	0.76825	0.79942	0.74473	0.54131	0.61907	0.67458	0.60055	0.53433	0.57289	0.64045	0.57371
PR	0.67949	0.72493	0.73450	0.70839	0.52363	0.56532	0.63997	0.56661	0.48888	0.52964	0.60693	0.53198
PRC	0.79581	0.83784	0.85487	0.82458	0.64783	**0.73628**	0.76036	0.70544	0.54468	0.62540	0.68326	0.60623
RF	0.75742	0.80596	0.82248	0.78987	0.55353	0.62548	0.67361	0.60753	0.50310	0.59366	0.64516	0.56880
RFW	0.71965	0.77071	0.79552	0.75564	0.60867	0.67578	0.70508	0.65514	0.50162	0.56664	0.62005	0.55290
ROC	0.79356	**0.84847**	**0.86021**	**0.82853**	**0.67759**	0.73572	**0.77310**	**0.72085**	0.51490	0.59701	0.67030	0.58112
S2N	**0.79869**	0.84113	0.85878	0.82786	0.66593	0.72309	0.76571	0.70993	0.52251	0.62866	0.68882	0.59947
SU	0.74061	0.78933	0.80311	0.77247	0.55928	0.61361	0.69997	0.61256	0.44163	0.51444	0.69600	0.52949
Average	0.74085	0.79248	0.81145	0.77571	0.58785	0.65397	0.70819	0.63997	0.49216	0.56334	0.65111	0.55563

Figure 3.2: Stability: Number of Attributes

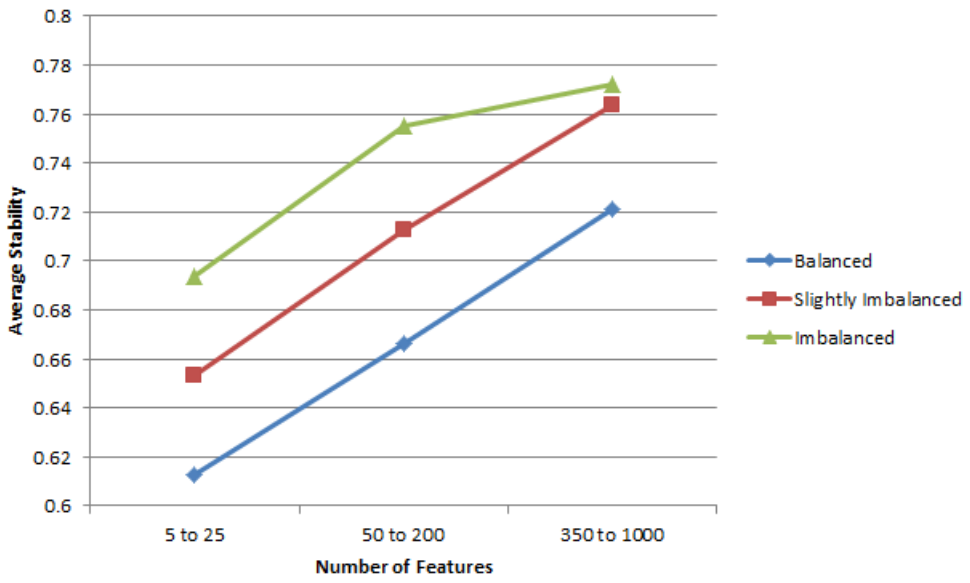


size is an essential factor when it comes to feature selection as it is the framework of the final subset. As in the previous tables, Table 3.6 is split into columns containing the three feature subset ranges (5 to 25, 50 to 200, and 300 to 1000). The results show that for every ranker, as the feature subset size increases, so does stability.

Dataset Perturbation is used to test how stable the feature ranker is with respect to changes in the dataset. Table 3.7 and Figure 3.5 show that as perturbation levels increase, stability decreases. While this may be intuitive, our results do show this trend. This leads us to state that, after enough change, any feature selection method becomes unstable.

In terms of choice of ranker an interesting trend appeared upon looking at Tables 3.2 through 3.7. The trend is that with the exception of one case (Table 3.5 Hard datasets, using 300 to 1000 features) the most stable rankers in each column are either ROC, PRC, S2N, or Dev. This consistency provides evidence that these four rankers are very robust and stable.

Figure 3.3: Stability: Class Balance

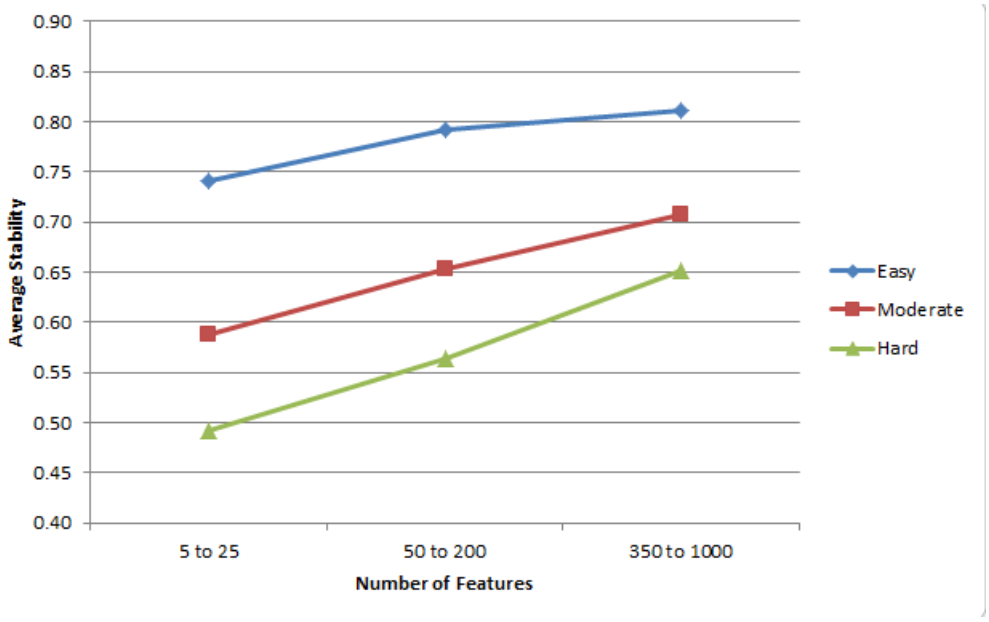


3.5.2 Difficulty-of-Learning Study

In this case study we examine the effects of difficulty-of-learning on the stability of six feature selection techniques of various levels of relative stability. We tested these six rankers on a set of twenty-six DNA microarray datasets with varying levels of difficulty. Additionally we used four levels of dataset perturbation and twelve feature subset sizes. The results of our experiments can be found in Tables 3.8 through 3.19. Each table presents the average results for every pairwise combination of ranker and subset size while keeping the level of difficulty-of-learning and level of dataset perturbation static.

The first trend to notice is that the stability of a feature selection technique is directly affected by the level of difficulty. In general, as the difficulty-of-learning increases, the stability of the resulting feature subsets decreases. With few exceptions, this is true for each ranker at all levels of perturbation and all feature subset sizes. Interestingly, the relative stability between the rankers seems to remain constant at all difficulty-of-learning levels. This leads us to state that even a relatively stable

Figure 3.4: Stability: Difficulty-of-Learning



feature selection technique such as ROC will produce unstable feature subsets if the dataset is difficult to learn from. However, a stable feature selection technique will still produce more stable feature subsets than unstable techniques.

Another trend is how the level of dataset perturbation affects the stability at different levels of difficulty-of-learning. It has been shown in the past that the larger the level of perturbation, the more unstable the feature subsets become [46]. However, our research shows that the amount of difference in stability between the difficulty-of-learning levels changes based on the levels of dataset perturbation. In general, the larger the level of perturbation, the larger the difference between the levels of difficulty. This trend is found for all of the rankers. This trend means that the level of difficulty-of-learning matters even more as the amount of change increases.

Considering the feature selection techniques themselves, we do see an interesting trend when comparing the levels of difficulty-of-learning. In general, we see that the relatively unstable feature selection techniques will have a larger change in stability when moving from Easy datasets to Moderate datasets than from Moderate to Hard

Table 3.6: Average Stability For Feature Subset Sizes

Ranker	Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall
CS	0.62772	0.70035	0.76223	0.68556
Dev	0.69738	0.75818	0.78310	0.73908
F	0.68536	0.74314	0.77368	0.72670
GI	0.62849	0.65940	0.68875	0.65386
GM	0.69389	0.74995	0.78048	0.73423
GR	0.60149	0.62963	0.71817	0.64004
IG	0.61479	0.69881	0.76402	0.68011
KS	0.69208	0.74938	0.77858	0.73281
MI	0.66155	0.73494	0.76593	0.71211
OR	0.60543	0.67592	0.71711	0.65685
Pow	0.62753	0.69625	0.74004	0.67857
PR	0.60686	0.65054	0.68815	0.64175
PRC	0.71337	0.77355	0.80005	0.75510
RF	0.66146	0.72349	0.75402	0.70528
RFW	0.65211	0.70956	0.74090	0.69346
ROC	0.71321	**0.77409**	0.80359	0.75610
S2N	**0.71494**	0.77303	**0.80462**	**0.75672**
SU	0.64127	0.69592	0.75871	0.68884
Average	0.65772	0.71645	0.75679	0.70207

datasets. The opposite is true for the stable techniques in that they will have a larger change moving from the Moderate to Hard datasets than from the Easy to Moderate datasets. This may indicate that on the Moderate datasets, the difference in stability between the stable and unstable techniques is most pronounced, because it is here where the stable techniques are still able to produce consistent results while the greater amount of difficulty (compared to the Easy datasets) is enough to stymie the less-stable feature selection techniques.

Lastly, we look at the feature subset sizes. We see two trends that arise from the feature subset sizes: in general, as the feature subset size increases, the stability increases and the difference between the levels of difficulty-of-learning decreases. The is found to be true for all of the rankers and all levels of perturbation. The only main

Table 3.7: Average Stability For Dataset Perturbaiton

Ranker	66.67%				80.00%				90.00%				95.00%			
	Feature Subset Sizes				Feature Subset Sizes				Feature Subset Sizes				Feature Subset Sizes			
	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall	5 to 25	50 to 200	300 to 1000	Overall
CS	0.47035	0.56446	0.65246	0.54725	0.57360	0.65510	0.72940	0.63971	0.68667	0.75398	0.80495	0.73868	0.78026	0.82787	0.86210	0.81659
Dev	0.55708	0.63658	0.67111	0.61209	0.65137	0.72279	0.75122	0.70014	0.75384	0.80590	0.82776	0.78968	0.82721	0.86747	0.88232	0.85441
F	0.54335	0.61733	0.65671	0.59635	0.63848	0.70536	0.74050	0.68628	0.74187	0.79303	0.82127	0.77877	0.81772	0.85685	0.87624	0.84539
GI	0.46027	0.48926	0.52696	0.48661	0.57363	0.61060	0.64981	0.60500	0.69902	0.73102	0.75719	0.72423	0.78105	0.80673	0.82103	0.79961
GM	0.54478	0.62395	0.66659	0.60162	0.64862	0.71199	0.74899	0.69484	0.75451	0.80037	0.82604	0.78768	0.82765	0.86349	0.88031	0.85276
GR	0.43137	0.45913	0.58489	0.47900	0.53960	0.57104	0.67515	0.58397	0.67189	0.69665	0.77058	0.70482	0.76308	0.79170	0.84206	0.79236
IG	0.44929	0.55612	0.65198	0.53558	0.55747	0.65289	0.73049	0.63253	0.67946	0.75370	0.80768	0.73626	0.77295	0.83254	0.86594	0.81606
KS	0.54158	0.62216	0.66404	0.59905	0.64646	0.71137	0.74661	0.69313	0.75280	0.80003	0.82400	0.78634	0.82747	0.86398	0.87969	0.85269
MI	0.51070	0.59935	0.64425	0.57364	0.61307	0.69541	0.73172	0.67018	0.72071	0.78752	0.81372	0.76623	0.80170	0.85746	0.87402	0.83837
OR	0.42178	0.50268	0.55502	0.48206	0.53832	0.61940	0.66790	0.59775	0.67683	0.74646	0.78457	0.72697	0.78479	0.83512	0.86096	0.82061
Pow	0.45230	0.53872	0.60630	0.51961	0.57118	0.64927	0.69974	0.62935	0.69765	0.75751	0.79282	0.74140	0.78900	0.83952	0.86131	0.82392
PR	0.41252	0.46595	0.51993	0.45718	0.54723	0.59955	0.64737	0.58970	0.68828	0.72815	0.75994	0.71948	0.77943	0.80853	0.82538	0.80062
PRC	**0.56397**	0.65053	0.69077	0.62452	0.66857	0.73946	0.77049	0.71768	0.77337	0.82252	0.84402	0.80742	0.84755	0.88169	0.89494	0.87078
RF	0.49494	0.57013	0.61515	0.55006	0.61321	0.68030	0.71449	0.66089	0.72278	0.78417	0.81151	0.76543	0.81491	0.85934	0.87495	0.84473
RFW	0.46904	0.54699	0.59415	0.52630	0.60092	0.66215	0.69909	0.64587	0.72554	0.77776	0.80258	0.76220	0.81294	0.85135	0.86780	0.83946
ROC	0.56355	**0.65182**	**0.69644**	**0.62619**	0.67127	**0.74073**	**0.77503**	**0.72036**	0.77310	0.82250	0.84623	0.80785	0.84492	0.88132	0.89666	0.86999
S2N	0.56272	0.63896	0.68684	0.61916	**0.67179**	0.73819	0.77459	0.71962	**0.77596**	**0.82656**	**0.85275**	**0.81202**	**0.84931**	**0.88842**	**0.90429**	**0.87609**
SU	0.48056	0.55242	0.64424	0.54543	0.58594	0.64910	0.72358	0.64140	0.70453	0.75224	0.80394	0.74529	0.79406	0.82990	0.86307	0.82326
Average	0.49612	0.57147	0.62932	0.55454	0.60615	0.67304	0.72090	0.65713	0.72216	0.77445	0.80842	0.76115	0.80644	0.84685	0.86850	0.83543

Table 3.8: Average Stability for the Easy Datasets using 95% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.83410	0.86292	0.89131	0.87475	0.86966	0.88866	0.89426	0.88977	0.90654	0.91076	0.90939	0.90875
IG	0.77230	0.80661	0.82865	0.83541	0.84432	0.86664	0.87614	0.87922	0.89225	0.89837	0.89417	0.88525
Pow	0.78253	0.82667	0.84593	0.84616	0.84975	0.87135	0.87222	0.88024	0.89138	0.88585	0.89521	0.89459
PR	0.79497	0.81488	0.84950	0.84985	0.85644	0.85691	0.85893	0.85621	0.85351	0.84578	0.84674	0.85503
ROC	0.85501	0.89586	0.90039	0.90786	0.90690	0.91771	0.91548	0.92025	0.92466	0.92489	0.92456	0.92549
S2N	0.86791	0.88495	0.89683	0.91198	0.90689	0.91354	0.91706	0.91516	0.92523	0.92888	0.92683	0.92482

exceptions to this rule is between feature subset sizes 5 and 10 between the Easy and Moderate datasets, which will in general show an increase in stability as the level of difficulty-of-learning goes up. This final observation may result from unusual properties of these extremely small feature subset sizes, for example only a very small number of features containing useful information on the difficult datasets which could lead to greater stability (at subset sizes 5 and 10) compared to easier datasets which have more relevant features to choose from.

Table 3.9: Average Stability for the Easy Datasets using 90% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.76962	0.81263	0.84217	0.82539	0.82605	0.84547	0.85363	0.85015	0.86873	0.87327	0.87241	0.87067
IG	0.68869	0.75610	0.77448	0.77836	0.79178	0.80946	0.82821	0.82802	0.84932	0.85669	0.85351	0.83773
Pow	0.69226	0.74056	0.76904	0.78600	0.78375	0.80140	0.81476	0.81982	0.83801	0.83550	0.84309	0.84625
PR	0.70070	0.73100	0.77038	0.77856	0.78885	0.78933	0.79383	0.79313	0.79329	0.79240	0.79447	0.80322
ROC	0.80920	0.84246	0.85245	0.86342	0.86570	0.88057	0.87935	0.88591	0.89148	0.89282	0.89238	0.89418
S2N	0.82789	0.84133	0.84724	0.86787	0.86757	0.87777	0.87903	0.87709	0.89053	0.89446	0.89317	0.89305

Table 3.10: Average Stability for the Easy Datasets using 80% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.65264	0.72339	0.75636	0.75564	0.75673	0.77805	0.78939	0.78790	0.80788	0.81369	0.81446	0.81266
IG	0.57884	0.65819	0.69252	0.69680	0.71088	0.73489	0.75236	0.75373	0.78106	0.79046	0.78582	0.76433
Pow	0.56819	0.62842	0.67122	0.68485	0.68807	0.71296	0.72391	0.73618	0.75534	0.76165	0.76932	0.78011
PR	0.57352	0.59885	0.63726	0.65671	0.67215	0.67770	0.68193	0.68399	0.69244	0.69231	0.69774	0.71199
ROC	0.71047	0.75545	0.77704	0.78533	0.79463	0.82160	0.82281	0.83016	0.83728	0.83932	0.83961	0.84309
S2N	0.72471	0.76656	0.77614	0.79222	0.79738	0.81232	0.81592	0.81674	0.83280	0.83896	0.83808	0.84069

Table 3.11: Average Stability for the Easy Datasets using 66.67% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.55703	0.63082	0.66803	0.67786	0.68136	0.70838	0.71850	0.72165	0.74176	0.74974	0.75210	0.74897
IG	0.47521	0.56563	0.60643	0.61170	0.62518	0.65646	0.67716	0.68067	0.70896	0.72126	0.71634	0.69377
Pow	0.45123	0.51917	0.55782	0.58226	0.58783	0.62255	0.63386	0.64446	0.67362	0.68323	0.69112	0.70714
PR	0.44234	0.47404	0.51320	0.53554	0.55099	0.56164	0.56249	0.56811	0.57551	0.58125	0.58728	0.60578
ROC	0.59306	0.65376	0.68604	0.70377	0.71239	0.74953	0.75717	0.76466	0.77687	0.77924	0.78141	0.78558
S2N	0.62509	0.67555	0.68648	0.70176	0.70749	0.73434	0.74185	0.74495	0.76381	0.77348	0.77520	0.77775

Table 3.12: Average Stability for the Moderate Datasets using 95% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.73836	0.78951	0.81401	0.83252	0.83552	0.84131	0.84802	0.83696	0.85194	0.85195	0.86153	0.84606
IG	0.63560	0.66720	0.72795	0.73609	0.74542	0.78249	0.79262	0.78810	0.78723	0.78144	0.80763	0.86686
Pow	0.71833	0.76008	0.75639	0.74498	0.73134	0.76286	0.80743	0.79038	0.82247	0.81279	0.80660	0.82880
PR	0.74635	0.74471	0.68766	0.71314	0.72149	0.74355	0.76096	0.73675	0.71798	0.80637	0.78756	0.81016
ROC	0.78509	0.80417	0.84068	0.84255	0.84441	0.85458	0.86103	0.86717	0.87069	0.88399	0.88453	0.87070
S2N	0.82511	0.80222	0.81847	0.84097	0.83940	0.85926	0.87625	0.87297	0.88601	0.89640	0.89091	0.89865

Table 3.13: Average Stability for the Moderate Datasets using 90% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.65024	0.71332	0.74754	0.76721	0.76465	0.77158	0.77706	0.76844	0.78289	0.78374	0.79405	0.78252
IG	0.55151	0.58637	0.63520	0.66209	0.67088	0.69944	0.70225	0.69502	0.69408	0.69460	0.74329	0.82122
Pow	0.65560	0.63978	0.61816	0.61676	0.63047	0.69216	0.71804	0.71027	0.72696	0.72962	0.73902	0.76231
PR	0.65559	0.64312	0.60652	0.62581	0.62926	0.65055	0.66792	0.65336	0.65352	0.72075	0.71492	0.75128
ROC	0.71834	0.73733	0.76571	0.77384	0.77300	0.78813	0.79803	0.80466	0.81451	0.83075	0.83451	0.81810
S2N	0.77706	0.72805	0.74710	0.77430	0.77175	0.78697	0.80637	0.81126	0.82637	0.84109	0.83454	0.84363

Table 3.14: Average Stability for the Moderate Datasets using 80% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.52609	0.59033	0.62796	0.64259	0.64898	0.67222	0.67076	0.67282	0.69204	0.69155	0.69689	0.68657
IG	0.38331	0.47012	0.50989	0.52350	0.52744	0.56026	0.56374	0.56361	0.59077	0.60825	0.66132	0.76063
Pow	0.49544	0.49143	0.48969	0.47547	0.49480	0.54859	0.56964	0.57621	0.59955	0.59907	0.62197	0.64792
PR	0.50077	0.49012	0.46341	0.47185	0.48288	0.49819	0.52073	0.52794	0.52815	0.57922	0.59730	0.64365
ROC	0.61153	0.62705	0.66220	0.66493	0.67395	0.68794	0.70292	0.71368	0.73236	0.75246	0.75648	0.73880
S2N	0.63154	0.61442	0.63738	0.64667	0.64837	0.67727	0.70079	0.70592	0.72623	0.73842	0.73584	0.75553

Table 3.15: Average Stability for the Moderate Datasets using 66.67% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.43266	0.48813	0.53208	0.54413	0.55058	0.56022	0.56975	0.57589	0.58680	0.58856	0.59522	0.58144
IG	0.28722	0.34786	0.39440	0.40502	0.40762	0.43639	0.43601	0.43712	0.47497	0.50274	0.57913	0.69211
Pow	0.34859	0.38188	0.38671	0.37471	0.36691	0.39345	0.41934	0.43676	0.45607	0.49293	0.51268	0.52861
PR	0.33390	0.34715	0.32828	0.32955	0.33000	0.34245	0.35629	0.36500	0.39039	0.42854	0.45044	0.49888
ROC	0.51274	0.51413	0.55475	0.56915	0.57629	0.58675	0.59886	0.61330	0.63866	0.65991	0.66810	0.65140
S2N	0.51808	0.52090	0.52503	0.52683	0.53440	0.56133	0.57773	0.58743	0.61269	0.62737	0.62557	0.65806

Table 3.16: Average Stability for the Hard Datasets using 95% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.73880	0.72482	0.74493	0.77469	0.76628	0.81326	0.81404	0.81792	0.82764	0.83040	0.82922	0.85534
IG	0.70210	0.71533	0.72376	0.72988	0.73105	0.73961	0.75191	0.73707	0.79104	0.83605	0.84890	0.83123
Pow	0.69545	0.72259	0.72157	0.73653	0.74857	0.76445	0.77709	0.77079	0.79886	0.81285	0.82061	0.83425
PR	0.66878	0.67312	0.67893	0.70346	0.73818	0.70622	0.74433	0.78123	0.75274	0.80235	0.77948	0.77581
ROC	0.68988	0.72256	0.75267	0.76965	0.77602	0.79484	0.79761	0.79787	0.81271	0.82947	0.83821	0.85220
S2N	0.66987	0.77370	0.76824	0.79246	0.78760	0.81006	0.82848	0.83322	0.83792	0.85085	0.85166	0.86365

Table 3.17: Average Stability for the Hard Datasets using 90% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.59876	0.60753	0.61333	0.63343	0.65478	0.69313	0.70509	0.71224	0.73021	0.74026	0.74239	0.77050
IG	0.49315	0.52077	0.54173	0.55047	0.55364	0.57761	0.59782	0.59874	0.67677	0.73712	0.75231	0.75803
Pow	0.59208	0.61694	0.62516	0.60662	0.61192	0.61695	0.63013	0.65080	0.67445	0.69481	0.70132	0.73937
PR	0.55763	0.55303	0.56955	0.58080	0.59439	0.59611	0.62902	0.63883	0.64556	0.70439	0.68111	0.69604
ROC	0.54650	0.58302	0.62105	0.63668	0.64001	0.66966	0.67856	0.68824	0.70585	0.72959	0.73940	0.76414
S2N	0.52315	0.59302	0.62142	0.63891	0.64091	0.67473	0.70216	0.71479	0.73188	0.74734	0.75986	0.77745

Figure 3.5: Stability: Dataset Perturbation

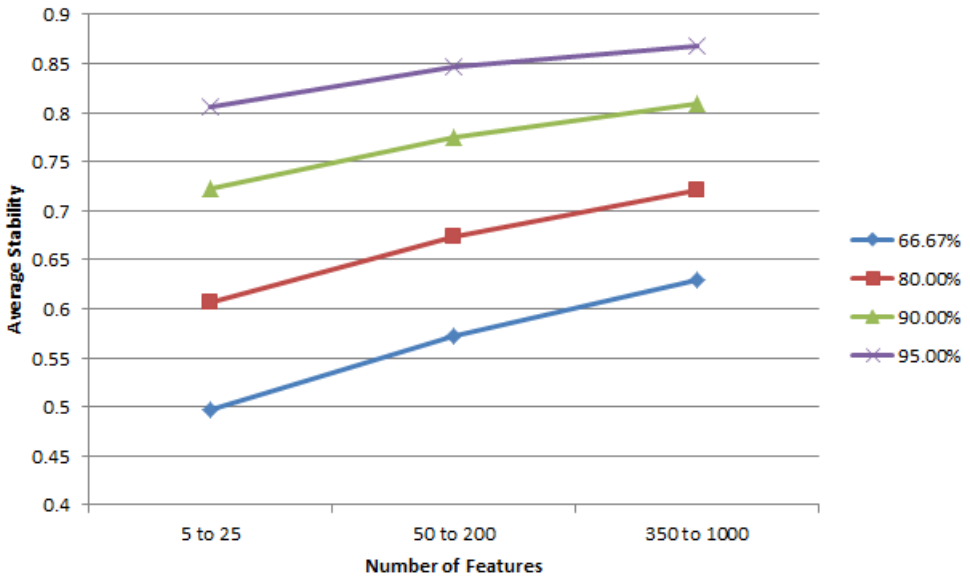


Table 3.18: Average Stability for the Hard Datasets using 80% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.46539	0.47411	0.49322	0.51271	0.52569	0.56828	0.58553	0.59211	0.61779	0.62870	0.64018	0.66574
IG	0.29530	0.32453	0.34449	0.37383	0.38534	0.41791	0.44021	0.45521	0.54966	0.62293	0.64345	0.68664
Pow	0.42091	0.43960	0.45608	0.45421	0.46391	0.48052	0.48944	0.50847	0.54292	0.55579	0.58663	0.61897
PR	0.38423	0.39680	0.39974	0.40667	0.41542	0.43186	0.44717	0.46042	0.48336	0.54062	0.53360	0.56965
ROC	0.40978	0.44070	0.46125	0.48200	0.49349	0.52945	0.53700	0.54839	0.58084	0.61176	0.62786	0.65757
S2N	0.38532	0.45569	0.46754	0.47394	0.48171	0.52983	0.55729	0.57362	0.59832	0.61991	0.63563	0.66172

3.6 CONCLUSIONS

Feature selection is an effective method of improving the process of classification. In the case of bioinformatics datasets, the process is not only useful but even necessary to bring the datasets to a more manageable size by removing the irrelevant and redundant features. In this study we examined the stability of eighteen different feature selection techniques using twenty-six different datasets by comparing the subsets created from reduced datasets with the subset created from the original dataset. This

Table 3.19: Average Stability for the Hard Datasets using 66.67% Perturbation

Ranker	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Dev	0.34979	0.36737	0.38310	0.39950	0.40949	0.44397	0.46675	0.48136	0.50563	0.52491	0.53805	0.56347
IG	0.13302	0.17218	0.20691	0.22915	0.24795	0.27884	0.29982	0.32702	0.43855	0.51213	0.53943	0.61751
Pow	0.26752	0.29839	0.30889	0.30733	0.32298	0.34736	0.35957	0.37109	0.40215	0.43593	0.46468	0.49898
PR	0.22640	0.24725	0.25329	0.26146	0.26849	0.27361	0.29089	0.30308	0.32768	0.37693	0.37790	0.42019
ROC	0.26529	0.30727	0.31405	0.33679	0.34167	0.38021	0.39858	0.41367	0.45378	0.48940	0.50927	0.53994
S2N	0.24749	0.29447	0.30848	0.32594	0.33722	0.37939	0.40347	0.42348	0.45723	0.48261	0.50014	0.53751

is different from previous studies which compare the performance of feature subsets from a number of reduced datasets to each other. In this chapter we presented two case studies regarding feature selection stability.

Our results in All Factors Study, which focuses on dataset characteristics and their effect on feature selection stability, show a number of trends when it comes to the effects of various data factors within this domain. One of our more interesting trends is that as the class balance level decreases, stability increases. We believe this is due to the fact that while we take the same fraction of instances from both classes, in the larger levels of imbalance, fewer instances are taken from the minority class (the class of interest), and therefore it is less likely that an important instance will be taken away in the reduction of the datasets. We suggest further investigation into this trend in order to better understand the underlying cause of this phenomenon, as this is the first paper to report this result.

The other trends found with the data factors can be split into two categories: trends from factors inherent to the data and trend from factors chosen during experimentation. The first trend is that as the difficulty-of-learning increases, stability increases. We believe this is because datasets that are easy to learn from have much stronger features that those that are more difficult and the more likely those features will be chosen for feature subsets. The next trend from a factor inherent of the data

is that as the number of instances increases, so does stability. We believe that as the number of instances increase the support for the stronger features will increase with more data. The number of attributes plays a role in the stability of the feature rankers in that as the number of attributes increases, the stability of the feature rankers will also increase. As for the trends found within factors chosen due to experiment design, the number of features used in the feature subsets follows the trend of the datasets with more features producing higher stability values. We found that as the amount of perturbation increases, stability decreases. While intuitive, the results supports the statement that with enough change any feature selection method can be considered unstable. Lastly, in terms of the choice of ranker we believe the most stable rankers are ROC, PRC, S2N, and Dev.

In the Difficulty-of-Learning Study, which focuses on how difficulty-of-learning affects feature selection stability, the results show that there is a clear connection between difficulty-of-learning and feature list stability. As with the All Factors Study we find that in general, as the difficulty of the dataset increases, the stability decreases. This was found to be true for all rankers and levels of dataset perturbation. Additionally, it was found that the level of perturbation has a larger effect on stability as the difficulty-of-learning increases. Our results also indicated that the unstable feature selection techniques (IG, PR, and Pow) will have a more drastic change between the Easy and Moderate datasets than between the Moderate and Hard datasets. The opposite has been found to be true for the more stable feature selection techniques (Dev, ROC, and S2N). Lastly, we found that as the feature subset size increases, the stability of the feature subset increases and the distance between the levels of difficulty decreases; though it should be noted that the very small subset sizes (5 features) may show the opposite effect.

In conclusion, we find that the difficulty-of-learning is an important factor to consider. The more difficult datasets can produce unstable feature subsets even with

relatively stable rankers. Additionally, changes to the dataset are going to affect the more difficult dataset more than the easier ones.

CHAPTER 4
ENSEMBLE GENE SELECTION AND FEATURE RANK AGGREGATION

4.1 INTRODUCTION

Gene identification has long been a prevalent and challenging goal of the field of bioinformatics. By selecting the correct genes we can determine the current state or predict potential future states of a number of problems including patient response to a drug treatment, diagnosis of certain cancers and genetic diseases, or distinguishing between sub-types of a single disease. The challenge that arises with the task of gene identification is that the domain contains vast amounts of data which is quite difficult and even impossible to process without assistance. One possible solution to this dilemma may be found through the use of feature selection, a data pre-processing technique from the field of data mining.

The goal of feature selection is to choose an optimum subset of features (genes) and use only those features in subsequent analysis, instead of using the entire feature set. The feature selection identifies features that can be considered irrelevant or redundant and removing them from future analysis. What remains is an optimized feature subset which not only will reduce the computational costs of running the experiment when compared to using the entire dataset, but has the potential to even improve the performance of inductive models built from this data. The dimension reducing properties of these techniques makes them ideal for application in field like bioinformatics as a majority of the genes in the genome will not be useful to the problem at hand.

While there are a number of different types of feature selection techniques, univariate feature rankers (which determine the importance of each attribute to the classes seperately) have a number of benefits which makes them a popular choice for work in bioinformatics even over other other techniques such as multivariate subset evaluation (determining the importance of a subset of attributes in terms of the classes). These benefits include a relatively small computation time even in high dimensional (many attributes per instance) datasets and their output of a ranked list of the features is intuitive to interpret even for non-practitioners of data mining.

However, recent studies have shown that the feature subsets chosen by these techniques can be unstable (i.e., the features chosen for the subset differ from dataset to dataset) [17]. The confidence in the decisions of the feature ranker are severely reduced as the results will not be similar to a feature subset derived from similar data. As a result of this instability, there have been a number of studies both measuring the stability of feature ranking techniques [46] and creating new methods and procedures to measure and improve the stability of these feature subsets [84].

With the instability of these feature subsets in mind, potential ways of improving the stability of the subsets has been a popular topic of research. One of the more promising methods of improving the stability of feature selection is ensemble feature selection. The concept of ensemble (the aggregation of results from multiple techniques into a single result) has primarily been applied toward the building of inductive models and only recently been applied toward feature selection. The idea behind ensemble feature selection is to aggregate the results of multiple runs of feature selection and aggregate them into a single optimized feature subset. Ensemble feature selection has been shown to create more stable or robust feature subsets and will generally have comparable or even at times super inductive models built from their feature subsets, as opposed to models built from a single instance of feature selection. As the use of ensemble feature selection techniques is relatively new, a ma-

jority of the research has focused on a single approach to the ensemble concept: data diversity [95]. Data diversity consists of performing a single feature selection method on multiple datasets (or sampled data from the same dataset) and aggregating the results into a single feature subset. However, there are more approaches available to using ensemble feature selection.

However, data diversity is not the only approach to ensemble feature selection. Therefore, we present in this work two new approache to ensemble feature selection for the domain of bioinformatics. The two new approachs are functional diversity (using multiple feature selection methods on a single dataset), and a hybrid approach (a combination of functional and data diversity). In order to present the abilities of these new approaches we perform a similarity experiment which compares the new approach not only to each other but towards data diversity. Through observing the similarity between the approaches we can determine if the new approach are sufficiently distinct from data diversity to warrant further testing. We test the techniques across twenty-six DNA microarray datasets form a number of medical, genomic, and bioinformatics studies, and 10 univariate feature rankers (or an ensmble of those rankers as required). Additionally, we also compare the classification results of these three approaches as well as a set of feature rankers which have had no approach of ensemble selection applied.

4.2 CONTRIBUTIONS

The main contribution of this paper is a thorough analysis of the ensemble gene selection on bioinformatics data. There are three main aspects of ensemble gene selection process discussed in this work. The first focuses on the application of two relatively new ensemble approaches, Select-Bagging and Select-Boosting (the Bagging and Boosting algorithms with feature selection incorporated into each iteration of their respective algorithms), on balanced (no dataset has less than a 43.50% mi-

nority class distribution) bioinformatics datasets. We test these two approaches using a series of seven balanced bioinformatics datasets, three feature rankers, four subset sizes, and two classifiers. Additionally, to better observe the absolute effect of ensemble learning, we also observed the results when no ensemble approach is applied (denoted as No-Ensemble in this work).

The second aspect is a thorough study of nine rank aggregation techniques and their classification performance within the domain of bioinformatics. To compare these, we use an ensemble of twenty-five feature selection techniques on fifty iterations (each technique is used twice) with eleven bioinformatics datasets (specifically, gene microarray datasets used for distinguishing between either cancerous and non-cancerous cells or response and non-response to cancer treatment). Additionally, we also use five learners for building classification models.

Lastly, the final aspect is an in-depth analysis of how the number of iterations employed within ensemble feature selection can effect the classification results of models created using the resulting feature subsets. This work looks at three levels of iterations (10, 20, and 50) across a number of different scenarios. In particular, we tested across 11 datasets, 2 ensemble feature selection approaches (Data Diversity which uses a single feature selection algorithm on multiple sampled datasets derived from an original and a Hybrid approach which uses a collection of different feature selection algorithms against multiple sampled datasets derived from an original one), 5 learners, 10 feature selection algorithms (3 for Data Diversity and a collection of 10 for the Hybrid approach which contains the three from Data Diversity), and 4 feature subset sizes.

The rest of the paper is organized as follows: Section 4.3 contains discussions of previous research that are relevant to our work. Section 4.4 outlines methods used to conduct our empirical study. In Section 4.5, we present our results of each of the ensemble gene selection aspects along with discussions of our findings. Finally, in

Section 4.6, we present our conclusions and potential avenues of future study.

4.3 RELATED WORKS

One of the biggest issues regarding DNA microarray datasets is their inherent high dimensionality. The cause for the high dimensionality is that DNA microarrays test each sample for thousands of genes simultaneously. This, combined with the relatively low sample size commonly found in medical and bioinformatics datasets, makes the process of data mining much more difficult [83]. Therefore, it has become necessary to include dimension reducing techniques such as feature selection. It has been shown in a number of cases that classification models built with reduced feature sets outperformed classification based on models built using the entire feature set, and in all cases building models with fewer features will reduce the computation time [65].

Unfortunately, the level of high dimensionality found in the DNA microarray datasets excludes certain feature selection techniques due to computational time. For example, wrapper-based feature selection techniques employ a classifier when making a decision instead of relying only upon statistical measures. However, building a classifier can be very involved even for only one model, and is compounded when multiple models are being built [60]. This level of high dimensionality also excludes filter-based subset evaluation. While there are more efficient ways to find an optimal subset, these algorithms may find only a local optima, and even in the best-case scenario will take significantly more time than feature ranking. Therefore not only are univariate filter-based feature rankers the most appropriate techniques for feature selection but in some cases are the only feasible option.

However, univariate feature ranking techniques are very well suited for work in bioinformatics. There are a number of reasons why these techniques are ideal for this problem, including: the output of the techniques (a ranked list of features) is intuitive and easy to understand; the ranking of genes makes it easy for researchers

to further validate the results through laboratory techniques; and the relatively small computational time when compared to other types of feature selection techniques (filter-based subset evaluation, wrapper, etc.) [96].

The use of ensembles has been most frequently applied to the creation of learners for building inductive models. It has been shown that these ensemble learners are competitive with other learners and in some cases are superior. This has been found to be true within the domain of bioinformatics [33]. Recently, there has been studies in applying the ensemble concept to the process of feature selection [95, 62]. In 2012, our research group performed a survey of current methods of improving the stability of feature selection in bioinformatics data. Current research has shown that not only do models built with feature subsets created using ensemble methods have comparable (or better) classification performance (when compared to models built using a single feature selection method), but the feature subsets themselves are more robust and can be appropriately applied to other data from the same problem [9].

However, with ensemble feature selection comes a decision of how to aggregate the results. A number of different rank aggregation techniques have been proposed in the literature: some are simple (Mean, Median, Highest Rank, Lowest Rank), and some are less so. Recent work in the area of rank aggregation techniques has centered around developing unique and innovative approaches. These new techniques can focus on different aspects of the ranking process, including comparing results to randomly generated results [77], giving more weight to top ranking features [62], or combining two known techniques to enhance each other [7]. While there has been work focusing on comparing a large number of rank aggregation techniques [110], the focus of that work was on the similarity of the selected feature subsets, not the classification results. Additionally, previous research has shown that the choice of aggregation technique can affect classification results [107].

One aspect of data which can potentially affect the similarity between the tech-

niques as well as the classification performance is how difficult it is to learn from a dataset. Our research group has been one of the first to consider the effect of the inherent Difficulty of Learning of a dataset can affect results. Difficulty of Learning refers to a measurement of how difficult a data set is to learn from by performing classification experiments with a few learners without feature selection. This measurement give us an indication of what to expect in terms of classification performance and whether or not feature selection will have a significant effect on the analysis. For example, if the dataset is particularly easy to learn from (achieving >0.8 AUC on average with no feature selection) then feature selection will not be able to improve on the results as one can only improve up to 1 [34].

For some ensemble, there is the factor of the number of iterations of gene selection that is performed in the ensemble. This number changes the number of ranked feature lists to aggregate. In 2012, Awada et al. [8] looked at the similarity between ensemble ranked lists generated by differing numbers of iterations. They found that the final ranked feature list do change the most when comparing 10 iterations with either 20 or 50 iterations, but there is little difference between the final ranked feature lists between 20 iterations and 50 iterations. This indicates that the final feature rank lists seems to stabilize around 20 iterations.

In order to study the similarity between these ensemble approaches, we must use a method of measuring the similarity. Unfortunately, while the study of similarity between feature selection techniques has garnered some recent attention [36], a majority of the measures were developed to measure stability. However, since one of the more common methods of measuring the stability of a feature ranker involves calculating the similarity of different feature subsets [80], these techniques can be applied toward the study of similarity. As a result, there are a number of stability measures that are used to compare feature subsets and therefore are appropriate to use to measure similarity [84].

In order to better understand our methods, some background information on our chosen method of measuring similarity is necessary. In 2007, Kuncheva et al. [80] devised a framework to study the stability of feature selection methods by calculating the similarity of multiple feature subsets derived from the same technique using randomly sampled data of the original dataset as the input. This study defined the term consistency index. The consistency index is a measure of similarity between two different feature subsets. They devised this measure as a way to choose the best set of features for an experiment. As this method derives the commonality between the two feature subsets, we use it to determine the similarity of the feature selection techniques.

4.4 METHODOLOGY

In this chapter, we use three approaches towards ensemble gene selection: Data Diversity, Functional Diversity, and Hybrid Diversity which are discussed in Section 2.2.2. In terms of feature selection, for Data Diversity, we use four different feature selection algorithms: Information Gain (IG), ReliefF (RF), Area Under the Receiver Operator Characteristic (ROC) Curve, and Signal-to-Noise (S2N). It should be noted that the four feature selection techniques mentioned are also presented without an ensemble approach and RF was not used in the number of iterations experiment. For the Functional Diversity and Hybrid Diversity approaches, we used an ensemble of ten feature selection algorithms. We start with the four feature selection algorithms from Data Diversity (IG, RF, ROC, and S2N) and we add Chi-Squared, Mutual Information, Kolmogorov-Smirnov statistic, Deviance, Geometric Mean, and Area Under the Precision Recall Curve. All feature rankers are discussed in Section 2.2.1. In terms of subset sizes we use the twelve feature subset sizes described in Section 2.2.3 though only subset size 10, 25, 50, and 100 were used in the feature rank aggregation and the number of iterations experiments as they are an appropriate collection of feature

Table 4.1: Dataset List

Difficulty of Learning	Name
Easy	Ovarian Cancer
	Lung Michigan
	Lung Cancer
	ALL AML Leukemia
	Prostate MAT
	MLL Leukemia
	Lung
	DLBCL
	BCancer50k
	Colon50k
	Lymphoma
	Acute Lymphoblastic Leukemia
	Lymphoma MAT
	CNS MAT
	Lung50k
Moderate	Colon
	Ovarian MAT
	Prostate
	Brain Tumor
	Lung Cancer Ontario
Hard	ECML Pancreas
	Mulligan-R-PD
	Breast Cancer
	Mulligan-R-NR
	DLBCL NIH
	Central Nervous System

subset sizes based on preliminary research. Additionally, in this work we use the following classifiers: Naïve Bayes, MLP, 5-NN, SVM, and Logistic Regression all of which are described in Section 2.5.1. For the Similarity experiments we use the consistency index outlined in Section 2.4.2. To conduct the classification experiments, we use four runs of five-fold cross-validation and the Area Under the Receiver Operating Characteristic curve as the performance metric as detailed in Section 2.6. Lastly we use a series of 26 bioinformatics (see Table 4.1) to test the ensemble approaches. The particulars of all of the datasets including the Average AUC Values for each dataset can be found in Tables 2.1 and 2.2. We created three groupings based on these average AUC values: Easy (≥ 0.8, fifteen datasets), Moderate (< 0.8 and ≥ 0.7, five datasets), and Hard (< 0.7, six datasets) based on their average AUC value

as described in Section 2.1.

4.5 RESULTS

4.5.1 Ensemble Feature Selection Approaches

Similarity Results

In this study, we used ten different rankers to create ranked feature lists from twenty-six bioinformatics datasets, using three forms of ensemble feature ranking: data diversity (where bootstrapping is used to create ten different sets of sampled data from the original dataset, a single ranker is used to create a feature list from each of these, and mean aggregation is used to combine these ten lists into one), functional diversity (where ten different feature rankers are applied directly to the original dataset, and the resulting gene lists are combined with mean aggregation), and hybrid (where each of the ten different rankers are applied to a different random sampled data from the original data, with the resulting feature lists combined using mean aggregation). We then used pairwise comparison for each pair of ensemble types: holding the dataset and feature subset size constant, we compared all the ranked lists produced by the first ensemble with all the ranked lists from the second ensemble. These similarity values were then averaged across all the datasets, and the results are presented for each feature subset size. In total, because for each combination of dataset and subset size we had 10 ranked lists for data diversity and one ranked list each for functional and hybrid diversity, there are 10 such pairwise comparisons for the data-functional and data-hybrid comparisons, and one comparison for the functional-hybrid comparison. These values, averaged across all twenty-six datasets, are presented in the Table 4.2 below.

When we look at the table we see that at all feature subset sizes, the functional diversity and hybrid approaches has the largest similarity of the three comparisons.

Table 4.2: Average Similarity Between The Ensemble Feature Selection Approaches Using All 26 Datasets

Comparison	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Funct.-Hybrid	0.67670	0.73428	0.70970	0.73007	0.75310	0.78156	0.79639	0.79727	0.80262	0.81255	0.81467	0.81623
Funct.-Data	0.45735	0.52247	0.55816	0.57163	0.58609	0.62374	0.63506	0.63996	0.65565	0.67203	0.67596	0.67536
Data-Hybrid	0.47198	0.53981	0.56253	0.58762	0.59830	0.64398	0.66023	0.66944	0.68966	0.70198	0.70445	0.70738

Table 4.3: Average Similarity Between The Ensemble Feature Selection Approaches Using The 5 "Moderate" Datasets

Comparison	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Funct-Hybrid	0.63948	0.67932	0.66566	0.73877	0.76683	0.74894	0.76129	0.76254	0.74802	0.76395	0.77113	0.77408
Funct-Data	0.37535	0.42684	0.46645	0.48096	0.49755	0.51470	0.55029	0.57053	0.58132	0.59720	0.60306	0.60381
Data-Hybrid	0.38337	0.48093	0.52132	0.50608	0.50403	0.55066	0.58705	0.60699	0.62534	0.63333	0.63293	0.63643

Table 4.4: Average Similarity Between The Ensemble Feature Selection Approaches Using The 6 "Hard" Datasets

Comparison	Feature Subset Size											
	5	10	15	20	25	50	75	100	200	350	500	1000
Funct-Hybrid	0.39975	0.56634	0.52169	0.57435	0.58591	0.63183	0.61304	0.63183	0.65325	0.66463	0.66923	0.68797
Funct-Data	0.30638	0.37116	0.38926	0.40739	0.42420	0.47922	0.47472	0.46926	0.47793	0.49859	0.49934	0.50216
Data-Hybrid	0.35308	0.40620	0.41931	0.44662	0.46493	0.50162	0.51406	0.52102	0.53423	0.55120	0.55217	0.55576

Table 4.5: Average Classification Results Using Naïve Bayes and the 5 "Moderate" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.78828	0.81771	0.83068	0.82888	0.82324	0.80097	0.79329	0.78352	0.77762	0.76355	0.75247	0.73667
	RF Ranker	0.75458	0.78246	0.78290	0.77328	0.76470	0.74974	0.74901	0.73331	0.72386	0.71198	0.70539	0.69397
	MI Ranker	0.79369	0.81787	0.81944	0.82189	0.82952	0.79054	0.77573	0.78138	0.77520	0.76933	0.76742	0.74144
	ROC Ranker	0.81167	0.82056	0.82391	0.82332	0.81981	0.77850	0.78323	0.78121	0.77678	0.77637	0.76461	0.73959
Hybrid	Ensemble	0.80769	0.81986	0.82340	0.82070	0.81619	0.78214	0.77372	0.77179	0.77245	0.76704	0.76135	0.73962
Functional	Ensemble	0.81371	0.82643	0.82880	0.82896	0.83097	0.78551	0.76570	0.75209	0.75226	0.75977	0.75000	0.73384
Single	IG Ranker	0.71011	0.77134	0.79313	0.79178	0.79534	0.81209	0.79314	0.78763	0.76725	0.74855	0.73656	0.71615
	RF Ranker	0.75678	0.77726	0.79342	0.79391	0.78111	0.74970	0.74428	0.73902	0.73330	0.71104	0.69324	0.67962
	MI Ranker	0.79798	0.81781	0.83291	0.83898	0.84067	0.79238	0.77981	0.77816	0.77390	0.76296	0.75534	0.73471
	ROC Ranker	0.81540	0.82843	0.83733	0.83533	0.83139	0.79678	0.78819	0.77999	0.78649	0.77781	0.76666	0.74415

Table 4.6: Average Classification Results Using MLP and the 5 "Moderate" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.82661	0.85830	0.86117	0.86188	0.86530	0.86379	0.85848	0.85093	0.86082	0.84895	0.84312	0.83045
	RF Ranker	0.78619	0.79947	0.80786	0.80586	0.81625	0.81829	0.82275	0.82371	0.82779	0.83706	0.82509	0.73871
	MI Ranker	0.82581	0.84063	0.83466	0.84371	0.84731	0.85382	0.84626	0.84374	0.84788	0.83401	0.84067	0.83433
	ROC Ranker	0.83069	0.84609	0.85552	0.86256	0.85617	0.83612	0.82609	0.83308	0.83124	0.83348	0.84092	0.83123
Hybrid	Ensemble	0.82935	0.84177	0.84343	0.83147	0.82471	0.82282	0.81747	0.82580	0.83109	0.83066	0.83804	0.82340
Functional	Ensemble	0.84036	0.83915	0.82857	0.81768	0.82629	0.83210	0.81957	0.82677	0.82073	0.83395	0.83147	0.82907
Single	IG Ranker	0.80807	0.85113	0.86123	0.87030	0.87250	0.86555	0.86254	0.85972	0.86613	0.87198	0.85597	0.83910
	RF Ranker	0.77868	0.79764	0.80541	0.81144	0.81755	0.81998	0.81019	0.81638	0.82481	0.82684	0.81461	0.72986
	MI Ranker	0.82321	0.84450	0.83891	0.85224	0.86456	0.86087	0.85694	0.85853	0.85422	0.84647	0.84147	0.83520
	ROC Ranker	0.84069	0.85386	0.84782	0.85028	0.85041	0.84309	0.84954	0.84246	0.83295	0.83861	0.83337	0.84116

Table 4.7: Average Classification Results Using 5-NN and the 5 "Moderate" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.79005	0.82249	0.83710	0.82614	0.83023	0.82479	0.82521	0.83202	0.82961	0.83572	0.83316	0.83407
	RF Ranker	0.76072	0.80464	0.80622	0.82305	0.82137	0.82378	0.82151	0.81901	0.81524	0.81397	0.81217	0.80620
	MI Ranker	0.79183	0.82897	0.82700	0.82204	0.82559	0.83569	0.83660	0.83122	0.82569	0.82713	0.83244	0.83524
	ROC Ranker	0.80805	0.83463	0.83412	0.83778	0.84500	0.84296	0.83712	0.83395	0.82013	0.82483	0.82556	0.82913
Hybrid	Ensemble	0.79430	0.82612	0.83905	0.83595	0.83840	0.83360	0.82891	0.83475	0.82112	0.82702	0.82695	0.83118
Functional	Ensemble	0.81305	0.82739	0.82140	0.82200	0.83830	0.84330	0.84016	0.83531	0.83019	0.82263	0.82500	0.83626
Single	IG Ranker	0.77330	0.81817	0.83189	0.83673	0.83909	0.84727	0.83227	0.84107	0.82969	0.82882	0.82589	0.82665
	RF Ranker	0.76396	0.79479	0.79828	0.81880	0.81811	0.81284	0.81276	0.81539	0.80634	0.80160	0.80227	0.79761
	MI Ranker	0.79722	0.82503	0.83668	0.82089	0.83347	0.83745	0.83830	0.83786	0.83444	0.82405	0.82474	0.82753
	ROC Ranker	0.81448	0.83355	0.83734	0.83782	0.83927	0.84166	0.83975	0.83663	0.81398	0.81683	0.82599	0.82766

Table 4.8: Average Classification Results Using SVM and the 5 "Moderate" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.84034	0.85687	0.88114	0.87838	0.87271	0.86582	0.86136	0.85250	0.87106	0.86811	0.86509	0.87274
	RF Ranker	0.79627	0.81859	0.82870	0.82371	0.81843	0.82655	0.82403	0.82282	0.83748	0.85597	0.85689	0.87539
	MI Ranker	0.83042	0.84812	0.85521	0.84743	0.85844	0.85555	0.84478	0.83880	0.84927	0.85658	0.86146	0.87124
	ROC Ranker	0.84230	0.85722	0.86318	0.86327	0.85507	0.84200	0.83878	0.84180	0.83716	0.85149	0.86051	0.86342
Hybrid	Ensemble	0.83054	0.84867	0.85357	0.85139	0.85275	0.85496	0.84351	0.83547	0.84628	0.85046	0.86008	0.86101
Functional	Ensemble	0.84194	0.85853	0.85541	0.84969	0.84175	0.83794	0.83610	0.84509	0.84295	0.85301	0.85493	0.86704
Single	IG Ranker	0.81876	0.85920	0.87584	0.87853	0.87829	0.85721	0.85278	0.83654	0.85430	0.87360	0.87640	0.87613
	RF Ranker	0.79488	0.81321	0.81980	0.82128	0.81484	0.81920	0.82860	0.82814	0.83530	0.85336	0.86555	0.87801
	MI Ranker	0.82877	0.85166	0.85885	0.86766	0.86568	0.85867	0.84828	0.84227	0.85375	0.86090	0.85944	0.87405
	ROC Ranker	0.84535	0.85824	0.86024	0.85513	0.85211	0.84008	0.83959	0.83764	0.84237	0.85311	0.86237	0.86728

Table 4.9: Average Classification Results Using Logistic Regression and the 5 "Moderate" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.80161	0.80453	0.80434	0.79532	0.78578	0.80711	0.78849	0.78511	0.80462	0.77910	0.78010	0.77817
	RF Ranker	0.79382	0.79589	0.77664	0.77137	0.78360	0.77920	0.75739	0.78369	0.76866	0.79056	0.78272	0.80159
	MI Ranker	0.81028	0.80226	0.79596	0.77631	0.78371	0.79620	0.76714	0.76905	0.77086	0.77735	0.78105	0.79968
	ROC Ranker	0.82413	0.81460	0.80052	0.80261	0.78334	0.77798	0.77027	0.78155	0.76967	0.77443	0.77275	0.79782
Hybrid	Ensemble	0.82427	0.81970	0.81026	0.78138	0.78006	0.80297	0.78779	0.77167	0.78122	0.79697	0.79262	0.78402
Functional	Ensemble	0.83839	0.82032	0.80611	0.78130	0.77232	0.78913	0.78656	0.79111	0.76867	0.80302	0.78961	0.79838
Single	IG Ranker	0.80620	0.80384	0.81189	0.78705	0.80835	0.79278	0.77557	0.77711	0.78229	0.80428	0.80483	0.80376
	RF Ranker	0.78359	0.77047	0.76938	0.78038	0.78368	0.77588	0.75856	0.76794	0.78158	0.77551	0.78694	0.80474
	MI Ranker	0.81460	0.80405	0.78950	0.80360	0.78718	0.79778	0.77160	0.76703	0.77559	0.77842	0.78146	0.80172
	ROC Ranker	0.84572	0.83852	0.81520	0.78680	0.76508	0.78474	0.78415	0.79115	0.77750	0.78614	0.77392	0.80008

Table 4.10: Average Classification Results Using Naïve Bayes and the 6 "Hard" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.64877	0.66653	0.68634	0.69818	0.70801	0.70871	0.70751	0.70388	0.70000	0.69825	0.68459	0.64284
	RF Ranker	0.67820	0.69130	0.69181	0.69700	0.69800	0.69626	0.69254	0.69323	0.69566	0.70130	0.70084	0.69137
	MI Ranker	0.65234	0.66780	0.67274	0.67498	0.67889	0.69216	0.70901	0.71249	0.71270	0.71644	0.71127	0.67581
	ROC Ranker	0.65883	0.67925	0.68769	0.69519	0.69891	0.70814	0.70802	0.70766	0.68452	0.67471	0.65881	0.63368
Hybrid	Ensemble	0.66730	0.68977	0.69012	0.68838	0.69518	0.70205	0.70317	0.70737	0.71957	0.71212	0.71044	0.68820
Functional	Ensemble	0.68184	0.68797	0.69882	0.70071	0.70127	0.69324	0.70243	0.70452	0.70933	0.70642	0.70402	0.69672
Single	IG Ranker	0.64294	0.65708	0.68059	0.68713	0.69583	0.71080	0.71588	0.71436	0.70856	0.69925	0.68794	0.64732
	RF Ranker	0.69553	0.69930	0.69975	0.70016	0.70384	0.70025	0.69667	0.69910	0.70935	0.69814	0.69866	0.68616
	MI Ranker	0.64683	0.66255	0.67092	0.67831	0.68634	0.69586	0.69638	0.70356	0.71163	0.71217	0.71530	0.68278
	ROC Ranker	0.64998	0.68327	0.69544	0.69748	0.70265	0.70683	0.71049	0.70845	0.70082	0.68489	0.67853	0.63829

Table 4.11: Average Classification Results Using MLP and the 6 "Hard" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.65154	0.66604	0.67543	0.68387	0.68106	0.68711	0.69491	0.68466	0.70160	0.70540	0.71215	0.71106
	RF Ranker	0.67621	0.67946	0.67913	0.68259	0.68014	0.68941	0.69046	0.69370	0.70663	0.71812	0.71811	0.72664
	MI Ranker	0.63982	0.64987	0.65424	0.65917	0.66439	0.66432	0.67534	0.68392	0.70425	0.71751	0.72363	0.72530
	ROC Ranker	0.65753	0.67050	0.67908	0.67937	0.67849	0.69005	0.68906	0.68938	0.70749	0.71139	0.71430	0.72285
Hybrid	Ensemble	0.64705	0.66822	0.67950	0.68062	0.66636	0.68247	0.68143	0.68979	0.71435	0.71720	0.71563	0.71727
Functional	Ensemble	0.66497	0.67582	0.68173	0.67850	0.66826	0.66383	0.68491	0.68456	0.69073	0.70751	0.71325	0.71599
Single	IG Ranker	0.64094	0.66366	0.67978	0.67860	0.68119	0.68133	0.68656	0.68868	0.69788	0.70513	0.69725	0.70404
	RF Ranker	0.67107	0.68972	0.68850	0.69323	0.68823	0.67394	0.68746	0.69160	0.70415	0.70755	0.71386	0.71607
	MI Ranker	0.64461	0.65551	0.65552	0.65906	0.66267	0.65996	0.67082	0.67970	0.70250	0.71219	0.71499	0.72283
	ROC Ranker	0.64729	0.66853	0.68261	0.67943	0.68079	0.68230	0.69319	0.69068	0.69631	0.71180	0.71504	0.72409

Table 4.12: Average Classification Results Using 5-NN and the 6 "Hard" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.63287	0.66168	0.67594	0.68676	0.68285	0.70137	0.70559	0.70147	0.70426	0.69960	0.70277	0.70065
	RF Ranker	0.66030	0.66046	0.68082	0.68609	0.68819	0.70376	0.70434	0.70427	0.69965	0.69698	0.69461	0.68706
	MI Ranker	0.63922	0.65775	0.66210	0.66923	0.67551	0.68168	0.68960	0.68889	0.69768	0.70326	0.71044	0.71579
	ROC Ranker	0.62971	0.65685	0.67061	0.67943	0.69162	0.69922	0.69338	0.70297	0.71163	0.71804	0.72675	0.71813
Hybrid	Ensemble	0.65049	0.68611	0.69165	0.69738	0.70116	0.71022	0.71717	0.70652	0.71922	0.71975	0.71653	0.72186
Functional	Ensemble	0.67917	0.68539	0.69922	0.69895	0.69314	0.70409	0.71739	0.71100	0.71267	0.71064	0.71151	0.71067
Single	IG Ranker	0.62701	0.65779	0.66481	0.67064	0.67901	0.69382	0.69937	0.70272	0.69825	0.69427	0.70036	0.69471
	RF Ranker	0.66852	0.67747	0.68031	0.68073	0.67799	0.69223	0.68904	0.69236	0.69489	0.69269	0.68844	0.67916
	MI Ranker	0.64033	0.65543	0.66555	0.66542	0.66859	0.68505	0.68519	0.68716	0.69860	0.70580	0.70817	0.71514
	ROC Ranker	0.63343	0.67121	0.67824	0.68685	0.68586	0.69491	0.71180	0.70539	0.71078	0.71035	0.71239	0.72218

Table 4.13: Average Classification Results Using SVM and the 6 "Hard" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.65859	0.67592	0.68528	0.68475	0.69493	0.69329	0.68815	0.67924	0.68799	0.70004	0.70821	0.71024
	RF Ranker	0.67852	0.68542	0.68552	0.69366	0.68221	0.68854	0.68339	0.68578	0.69516	0.71105	0.71552	0.73147
	MI Ranker	0.65236	0.65283	0.64633	0.66248	0.67101	0.67026	0.68310	0.68224	0.69956	0.71319	0.71772	0.72370
	ROC Ranker	0.66553	0.68297	0.68243	0.68499	0.69311	0.69807	0.69513	0.68541	0.70937	0.70963	0.71019	0.72598
Hybrid	Ensemble	0.65392	0.67704	0.69268	0.69109	0.68903	0.68624	0.69253	0.69347	0.70903	0.71390	0.71798	0.72500
Functional	Ensemble	0.67897	0.69364	0.68715	0.68774	0.68290	0.66770	0.68520	0.69126	0.69287	0.70693	0.71507	0.72199
Single	IG Ranker	0.64914	0.66837	0.68565	0.69152	0.68522	0.68653	0.68319	0.69656	0.69183	0.70022	0.69460	0.70527
	RF Ranker	0.68637	0.69507	0.68960	0.69309	0.68846	0.68525	0.69028	0.69184	0.70545	0.71330	0.71954	0.72671
	MI Ranker	0.65090	0.65624	0.65780	0.66495	0.66822	0.66603	0.66634	0.67435	0.68940	0.70723	0.71133	0.72016
	ROC Ranker	0.65033	0.68018	0.68591	0.68972	0.69177	0.69383	0.69678	0.69132	0.69383	0.70394	0.71117	0.72426

Table 4.14: Average Classification Results Using Logistic Regression and the 6 "Hard" datasets

Technique	Filter	Feature Subset Size											
		5	10	15	20	25	50	75	100	200	350	500	1000
Data	IG Ranker	0.64624	0.66171	0.65243	0.64034	0.63478	0.61472	0.59690	0.60030	0.63553	0.62760	0.63831	0.67765
	RF Ranker	0.67682	0.64260	0.65805	0.65715	0.63590	0.61652	0.60048	0.59568	0.60625	0.65372	0.64830	0.65882
	MI Ranker	0.65064	0.62746	0.60360	0.61930	0.60852	0.60591	0.61592	0.61330	0.60396	0.65261	0.64519	0.68429
	ROC Ranker	0.66267	0.65980	0.65695	0.65379	0.65727	0.64955	0.63390	0.62925	0.63343	0.65974	0.67615	0.70356
Hybrid	Ensemble	0.64917	0.65016	0.66745	0.64760	0.62625	0.62254	0.60866	0.61914	0.64334	0.63129	0.66563	0.67661
Functional	Ensemble	0.67065	0.66765	0.66931	0.65117	0.62625	0.61550	0.60886	0.61091	0.59765	0.63104	0.66625	0.68958
Single	IG Ranker	0.63574	0.64518	0.65884	0.64984	0.63900	0.60361	0.61918	0.61426	0.60277	0.59971	0.63799	0.66320
	RF Ranker	0.67741	0.68191	0.66928	0.64667	0.64768	0.63235	0.61414	0.62450	0.63846	0.66352	0.64647	0.65560
	MI Ranker	0.64369	0.62256	0.64380	0.65358	0.63455	0.60387	0.58991	0.58236	0.60923	0.63433	0.63398	0.67054
	ROC Ranker	0.64548	0.66652	0.67703	0.66227	0.65113	0.63413	0.63147	0.61313	0.61313	0.64376	0.67408	0.69733

It should be noted however at each of the feature subset sizes that the data diversity and hybrid techniques will be slightly more similar to each other than the data diversity and the functional diversity techniques. We believe that the cause for the lack of similarity between the functional diversity and data diversity lists is partially because one is comparing the results from an ensemble of filter-based feature selection techniques and therefore makes its decisions from a variety of rankers, while the other uses a single feature selection technique and so bases all of its decisions on a single ranker. Alternatively, we believe that the fact that we are comparing two techniques which employ an ensemble of filter-based features selection techniques accounts for the increased similarity between the functional diversity and hybrid techniques. While this comparison between an ensemble of filter-based feature selection techniques and a single filter-based feature selection technique is also true for the data diversity and hybrid comparison, in this case both of the techniques employ the bagging technique which accounts for the slight increase in similarity.

In addition to observing how the three comparisons (functional-hybrid, functional-data, data-hybrid) fare with respect to each other, it is important to observe the effect of feature subset size on similarity. In particular, we note that similarity increases as subset size increases. This makes sense because even if the different approaches disagree about which are the most important features, as they select more and more features it is likely that one list's top-ranked features will appear on the other list, and vice versa. All three comparisons show this improvement, each increasing their similarity by approximately 0.13-0.14 as the subset size changes from 5 to 1000. The functional-hybrid similarity shows the greatest improvement, perhaps because the increased subset size negates some of the randomness of the bagging step in the hybrid ensemble, which is the only step which distinguishes hybrid from functional.

Table 4.2 shows the results across all twenty-six of the datasets. However, these results do not take into account how difficult it is to learn from these datasets. There-

fore, we decided to narrow down our list of datasets to those datasets which are more difficult to learn from. Tables 4.3 and 4.4 contain the average similarity between the ensemble feature selection methods when considering only the Moderate and Hard datasets respectively (as described in Section 4.4). What we see is that a majority of the same patterns arise even when isolating these Moderate and Hard datasets. Once again, the functional diversity and hybrid approaches have the largest similarity. Additionally, the data diversity and hybrid approaches have more similarity than the data diversity and functional diversity approaches. In general, as the feature subset size increases, similarity increases, though there are more deviations from these patterns when isolating the Hard datasets. Notably, we see that the functional-hybrid pair shows unusually low similarity when using only 5 features on the Hard datasets, presumably because selecting so few features from such datasets is an inherently noisy process which is compounded with the increased difficulty-of-learning found in the Hard datasets. In general, the difficult-to-learn datasets have much less connection between their independent values (features) and the class, which results in both the reduced capacity of classification models (because there is less information which may be used to predict a class) and reduced stability (since without solid information on which features are most useful, rankers may find most features to be similar, with minor variations leading to different rankings). This explains the additional deviations we see in Table 4.4 which vary from the results seen in Tables 4.2 and 4.3.

Classification Results

While similarity is an important factor when comparing these ensemble techniques, it is not necessarily an indicator of their performance in terms of classification. Therefore we decided to perform a classification experiment using these techniques. To provide an even more exhaustive comparison, we also present how feature selection

performs without using any ensemble method at all by choosing a set of rankers to perform feature selection. For the rankers we chose the Information Gain (IG), ReliefF (RF), Mutual Information (MI), and ROC rankers as these rankers have shown good classification results and strong stability [46]. We chose only four rankers due to the computational cost of using more rankers. We performed the classification experiment using the same five learners discussed previously, using four runs of five-fold cross-validation and the AUC performance metric to evaluate the models. In each case, we selected the top features using the given ensemble technique (or no ensemble, in the case of Single), and built the models using only the selected features. Additionally, as data diversity only uses a single ranker at any given time we chose to present the same four rankers used in the non-ensemble feature selection for proper comparison. The results (Tables 4.5 through 4.14) contains the results from the classification experiments.

Looking at the classification results we see that in general the ensemble techniques do perform well in terms of classification but do not always outperform the results from the single run of feature ranking. In terms of the Moderate datasets, this trend is especially true. The number of cases in which the single run of feature ranking outperforms the ensemble techniques is greater than the number of cases in which the ensemble techniques are the top performer. Conversely, we start to see the benefit of the ensemble technique more clearly in the Hard datasets. With these datasets the number of cases in which the ensemble techniques are the top performer are greater than the number of cases where the single techniques are the top performer. What we do notice is that in the Moderate datasets the results are in general very close to one another and in the Hard datasets there are larger differences found between the techniques. We believe that the ensembles are better suited for discovering the harder to learn patterns found in the Hard datasets compared to the single run of feature ranking, but the patterns in the Moderate datasets are easier to discover and

place the single run of feature ranking in a more even scenario.

Delving deeper into the specific factors of the experiment we see that the results depend on the learner involved in the classification. When we look at the individual learners we see that three of the learners (MLP, SVM, and Logistic Regression) follow similar patterns in that in the Moderate datasets the single run of feature ranking outperforms the ensemble techniques and in the Hard datasets the ensemble techniques outperform the single run of feature ranking. The remaining two learners (5-NN and Naïve Bayes) will favor one type of technique or the other. 5-NN in both the Moderate and the Hard datasets will have the ensemble techniques outperform the single run of feature ranking, though in the Hard datasets this trend is more prevalent. Naïve Bayes will favor the single run of feature ranking in both scenarios over the ensemble techniques.

The ensemble approach data diversity uses only a single ranker to assist in making the feature subsets. Therefore, we look at how data diversity compares to the feature ranking when no ensemble approach is applied when both are using the same ranker. We notice that each ranker reacts differently toward applying it toward data diversity rather than by itself. Overall, we see that Information Gain and ReliefF show that they can benefit from the use of data diversity (though this is not always the case) and Mutual Information and ROC will perform better as single run of feature ranking. When we look at the Moderate datasets only ReliefF is improved by applying it to data diversity in a majority of the cases; Mutual Information and ROC will have superior performance when using them as a single run of feature ranking; and Information Gain is improved by the use of the data diversity approach exactly 50% of the cases. In terms of the Hard datasets, Information Gain and Mutual Information will generally be improved by the use of data diversity whereas ReliefF and ROC perform better as single run of feature ranking.

Table 4.15: ANOVA Results: Moderate Datasets

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	7.4680	9	0.8298	31.8713	0
X2	3.2426	11	0.2948	11.3224	0
X3	43.0412	4	10.7603	413.297	0
X1*X2	2.2984	99	0.0232	0.89174	0.7712
X1*X3	2.8078	36	0.0780	2.9957	0
X2*X3	11.9421	44	0.2714	10.4247	0
Error	1556.8032	59796	0.0260		
Total	1627.6032	59999			

Table 4.16: ANOVA Results: Hard Datasets

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
X1	2.0677	9	0.2298	8.8869	0
X2	9.5034	11	0.8639	33.4187	0
X3	28.6742	4	7.1686	277.2914	0
X1*X2	3.4352	99	0.0347	1.3422	0.0132
X1*X3	1.4772	36	0.0410	1.5872	0.0140
X2*X3	12.9052	44	0.2933	11.3453	0
Error	1856.0749	71796	0.0259		
Total	1914.1378	71999			

Statistical Analysis

In order to further validate the results in our classification experiments, we performed a three way ANalysis Of VAriance (ANOVA) test [14] on both the Moderate and Hard datasets to determine the effect of the choice of feature selection approach (X1), feature subset size (X2), and learner (X3) have on the AUC levels. The ANOVA analysis was performed within MATLAB.

Looking at the Moderate dataset results (Table 4.15) we see that all three of the main factors have a Prob>F score less than 0.05 which indicates that two or more of the populations of each factor are significantly different from each other. Additionally we see that the interactions between feature selection technique and learner as well as the interactions between the feature subset size and the learner also have Prob>F

scores less than 0.05 which indicates at least two different combinations of each these interactions are significantly different.

The Hard dataset results (Table 4.16) show similar patterns with the ANOVA test when compared to the Moderate dataset results. As with the Moderate datasets, we see that all three of the main factors have a Prob>F score less than 0.05 which indicates that these factors can be considered significant. However, unlike the Moderate datasets, the Hard datasets have all three interactions with Prob>F scores less than 0.05 indicating that all three of the interactions are considered significant. The reuslts of the interactions state that decision of which feature selection technique to pair with which feature subset size is significant only in the Hard datasets while the decision on which learner to pair with which feature selection technique and which learner to pair with which feature subset size is significant in both the Moderate and Hard datasets.

As the ANOVA results only show that the factors have at least two significantly different options, we need to investigate further into which options are significantly different from any other options. To this end we performed a series of Tukey's Honestly Significant Different (HSD) test [14] on each individual factor. As with the ANOVA analysis we performed the Tukey's HSD tests in MATLAB. In the Tukey's HSD test results (Figures 4.1 through 4.6) the circles are the group means and the extending lines are the 95% confidence intervals around those means. We can consider two values consistently differentiable if the lines do not overlap and are distinguishable.

Looking at the Moderate results we find each of the factors have a group of options which are considered the statistically best performers in terms of AUC. In terms of the feature selection techniques two of the techniques (ReliefF both with data diversity and when no ensemble approach is applied) are significantly worse than the remaining eight techniques. The remaining techniques are not significantly different from one

Figure 4.1: Tukey's HSD Results: Feature Selection Techniques on Moderate Datasets

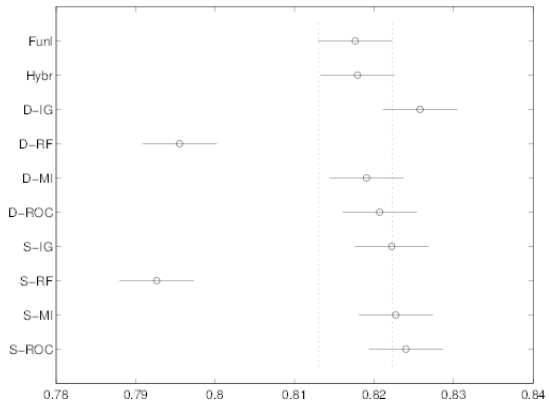
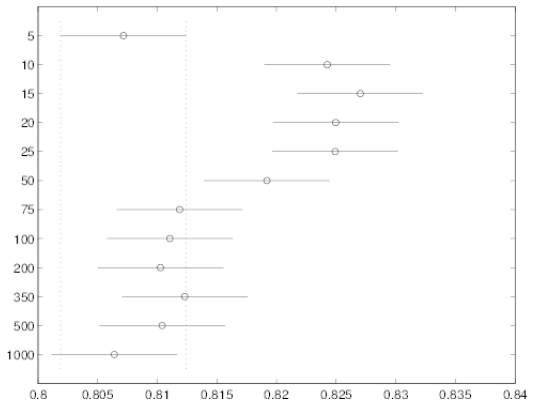


Figure 4.2: Tukey's HSD Results: Feature Subset Size on Moderate Datasets



another. This allows us to confidently state that for the Moderate datasets, the two newer techniques are comparable to both data diversity and the feature ranking techniques when no ensemble approach is applied in terms of classification results. For the feature subset sizes, the top performers were between 10 and 50 features inclusive. Lastly, of the learners, all of the learners are statistically different from each other and SVM is clearly the top performer.

The Hard datasets show different patterns in all three factors when compared to the Moderate datasets. In terms of the feature selection techniques the groups are much closer to each other with only three options (Mutual Information when no

Figure 4.3: Tukey's HSD Results: Learner on Moderate Datasets

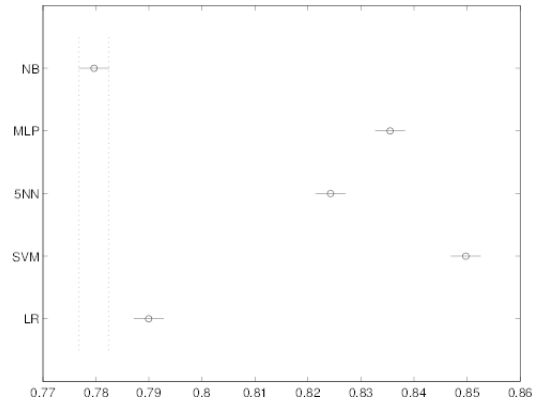
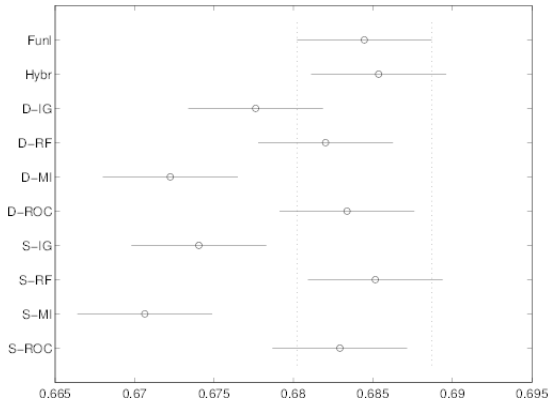


Figure 4.4: Tukey's HSD Results: Feature Selection Techniques on Hard Datasets



ensemble approach is applied or with data diversity and Information Gain when no ensemble approach is applied) being significantly worse than the top performer. As with the Moderate datasets the new techniques are comparable to data diversity and the single run of feature selection in terms of classification results. As for the feature subset sizes, there is a gradual trend that as feature subset size increases, classification performance increases. Additionally the range of 350 to 1000 is statistically indistinguishable from the top performer which is 1000 features. Finally, the learners are not significantly different from each other with the exception of Logistic Regression which is significantly worse then the other learners.

Figure 4.5: Tukey's HSD Results: Feature Subset Size on Hard Datasets

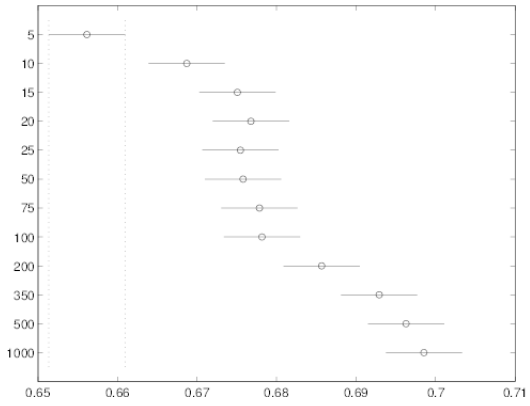
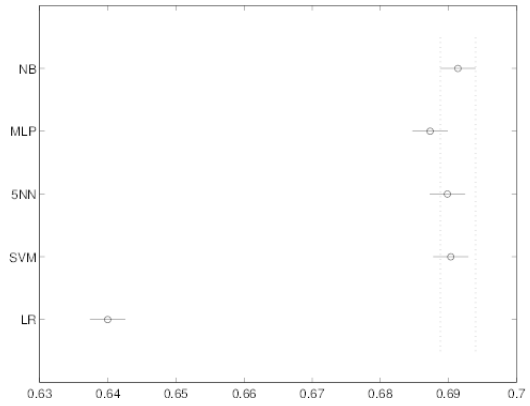


Figure 4.6: Tukey's HSD Results: Learner on Hard Datasets



4.5.2 Feature Rank Aggregation

This section contains the results from our experiment regarding the classification performance (in terms of AUC) of nine rank aggregation techniques. The performance was tested using an ensemble of twenty-five feature rankers along with the hybrid ensemble approach with 50 iterations (each ranker is used twice), applied toward eleven bioinformatics datasets from various genetics, biological, and biomedical experiments. Table 4.17 contains the results of our experiment, with all five learners presented in one table. The learner and subset size are shown in columns 1 and 2, and columns 3 through 11 contain the average classification results across the eleven datasets when keeping the learner, rank aggregation technique, and the feature subset size static.

Table 4.17: Average Classification Performance (in AUC) of the 9 Aggregation Techniques

Learner	Subset Size	Enhanced Borda	Exponential Weighting	Highest Rank	Lowest Rank	Mean	Median	Robust Rank	Round Robin	Stability Selection
5-NN	10	0.73907	**0.74439**	*0.70145*	0.70570	0.74358	0.74135	0.72014	0.71465	0.74039
	25	0.75324	0.75625	0.74384	*0.72097*	**0.76283**	0.74973	0.73133	0.74336	0.75163
	50	0.76149	0.76168	0.75345	*0.74189*	**0.76980**	0.76642	0.75899	0.75141	0.76080
	100	0.76537	0.76853	0.76508	*0.75707*	0.76446	0.76847	0.76804	**0.77371**	0.76491
LR	10	**0.72758**	0.72525	0.71930	0.71723	*0.71557*	0.72109	0.71952	0.72401	0.72341
	25	0.70646	0.71092	0.70706	*0.70236*	0.71081	0.71489	0.71003	**0.71605**	0.70481
	50	0.69938	**0.70871**	0.69220	0.70153	0.69756	0.69118	0.69924	*0.69098*	0.69267
	100	0.69279	0.69134	*0.68000*	0.68872	0.69102	**0.69330**	0.69328	0.68027	0.69268
MLP	10	**0.75647**	0.75117	0.74400	*0.73042*	0.74316	0.74636	0.74345	0.74144	0.75235
	25	0.75452	**0.76322**	0.75343	*0.73416*	0.74418	0.75776	0.75552	0.75935	0.76011
	50	0.76516	0.76832	0.76677	*0.73987*	0.74506	0.75219	0.75678	0.76117	**0.76936**
	100	0.76904	0.76616	0.77280	*0.74300*	0.75126	0.75968	0.76472	**0.77534**	0.76997
NB	10	**0.75618**	0.75219	0.73596	0.73395	0.74271	0.75183	*0.73217*	0.73304	0.75394
	25	0.75983	0.76167	0.75676	*0.73983*	0.75330	**0.76409**	0.75214	0.75157	0.76018
	50	0.74665	0.74874	**0.76670**	*0.72049*	0.74260	0.74391	0.74739	0.76591	0.74870
	100	0.74394	0.74516	0.75299	*0.72461*	0.74038	0.74503	0.73891	**0.75365**	0.74456
SVM	10	**0.76329**	0.75616	0.75059	0.74793	0.75576	0.75705	0.74793	*0.74233*	0.75776
	25	0.76905	**0.77089**	0.75691	*0.75570*	0.76421	0.76447	0.76178	0.75616	0.76616
	50	0.76528	0.76746	0.75507	*0.74975*	0.76320	0.76090	0.75544	0.75602	**0.76932**
	100	0.76131	0.76457	0.75895	*0.74090*	0.75657	0.76125	0.75956	0.76221	**0.76500**

The top performing value in each row is in boldface while the worst performing value in each row is in italics.

Looking at the top performers across the five learners we see that for seven of the rank aggregation techniques (Mean, Median, Highest Rank, Stability Selection, Exponential Weighting, Enhanced Borda, and Round Robin) there is at least one combination of feature subset size and learner which will have the rank aggregation technique be the top performer. Of these seven the most common top performer is a tie between Enhanced Borda, Exponential Weighting, and Round Robin with four combinations each. The least frequent top performer was Highest Rank with only a

single combination as top performer.

The remaining two rank aggregation techniques (Lowest Rank and Robust Rank) did not have a single combination of learner and feature subset size in which they were the top performing rank aggregation technique. While Robust Rank was never the top performer, it was only the worst performer once. However, Lowest Rank is by far the most frequent worst performer of the rank aggregation techniques. Lowest rank was the worst performer a total of fourteen out of twenty combinations. This leads us to say that of the nine rank aggregation techniques one should definitely avoid Lowest Rank.

Statistical analysis was performed and it was found that with the exception of Lowest Rank, all rank aggregation techniques perform similarly. First, a one-way ANOVA was performed, with the factor being the choice of rank aggregation and it was found to be statistically significant, meaning that at least one pair of values has different means. A Tukey's Honestly Significant Difference test performed on the factor confirms that the Lowest Rank technique differs from the rest, while all other techniques are statistically indistinguishable. We believe this due to Lowest Rank choosing the worst value for a feature; with 25 different rankers, there is a high chance of one randomly giving a bad result . Thus, as all other rankers are indistinguishable, we see that it is preferable to choose a simple and computationally inexpensive aggregation technique such as Mean over a more complex and computationally expensive aggregation technique.

4.5.3 Optimum Number of Iterations

In this paper we performed classification experiments using two different ensemble feature selection approaches: Data Diversity (using a single feature selection technique on multiple sampled datasets derived from a single one) and Hybrid (using a collection of feature selection technique on multiple sampled datasets derived from the original

Table 4.18: Average AUC: Data Diversity 10 Iterations

Subset	Naive Bayes			MLP			5-NN			SVM			Logistic Regression		
Size	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N
10	0.73525	0.74349	0.72568	0.75343	0.75032	0.73370	0.73477	0.73766	0.73180	0.75817	0.76217	0.74326	0.72663	0.73017	0.71740
25	0.76038	0.75386	0.72812	0.76481	0.75926	0.74390	0.74984	0.76134	0.74089	0.77574	0.76673	0.75391	0.70342	0.71458	0.69382
50	0.75065	0.74012	0.72482	0.76742	0.75644	0.75629	0.75747	0.76456	0.74041	0.77172	0.76349	0.75732	0.70217	0.70793	0.68334
100	0.74008	0.74109	0.69895	0.76023	0.75470	0.76024	0.76081	0.76251	0.74740	0.75799	0.75649	0.75976	0.68431	0.69848	0.70234

Table 4.19: Average AUC: Hybrid 10 Iterations

Subset Size	Naive Bayes	MLP	5-NN	SVM	Logistic Regression
10	0.74890	0.74711	0.74975	0.75505	0.72722
25	0.75018	0.73834	0.76354	0.76344	0.69617
50	0.73845	0.74627	0.76630	0.76293	0.70455
100	0.73665	0.75161	0.76480	0.75802	0.68847

dataset). To throughly test these ensemble approaches we use three different numbers of iterations of feature ranking (10, 20, and 50) to create ranked feature lists which were then applied to 5 different learners in order to perform classification experiments. In order to combine the resulting ranked feature lists created by each iteration we choose to use Mean Aggregation (calculating the average score of each feature based on the individual runs). In order to evaluate the classification performance of the models built from the gene lists created by the ensemble feature selection we use the Area Under the ROC Curve (AUC) metric. For each of the eleven datasets we performed four runs of five-fold cross-validation. In each case, we selected the top features using the given ensemble technique and built the models using only the selected features. We present average AUC values over all folds, runs, and datasets, for each of the level of number of iterations of feature selection are shown in Tables 4.18 through 4.23. It should be noted however that our experiments also found the main conclusion (that neither of 20 or 50 iterations reliably outperform the other) to be true when considered for each dataset individually.

In order to compare the different numbers of iterations of feature selection we

Table 4.20: Average AUC: Data Diversity 20 Iterations

Subset	Naive Bayes			MLP			5-NN			SVM			Logistic Regression		
Size	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N
10	0.74016	0.74950	0.72587	0.75428	0.75859	0.74382	0.73716	0.74611	0.72491	0.76705	0.75804	0.74877	0.73304	0.73766	0.71546
25	0.76363	0.75707	0.73419	0.76390	0.75837	0.75318	0.75541	0.75190	0.74061	0.77418	0.76696	0.75942	0.72091	0.71694	0.70957
50	0.75415	0.74540	0.71856	0.76940	0.75862	0.74825	0.76313	0.76387	0.74425	0.77509	0.76460	0.75858	0.68724	0.70690	0.69497
100	0.74012	0.74366	0.69669	0.76973	0.76054	0.75561	0.76181	0.76113	0.74758	0.76536	0.75316	0.75493	0.69460	0.69968	0.69750

Table 4.21: Average AUC: Hybrid 20 Iterations

Subset Size	Naive Bayes	MLP	5-NN	SVM	Logistic Regression
10	0.74611	0.74925	0.74798	0.75832	0.74520
25	0.75738	0.75046	0.76602	0.76687	0.72199
50	0.73907	0.74373	0.76869	0.75696	0.69439
100	0.73891	0.75229	0.77392	0.76088	0.69916

determine which numbers outperform the others. We calculate this by counting the number of combinations (consisting of learner, feature subset size, and feature selection technique in the case of Data Diversity) wherein each number of iterations is the top performer. When comparing 10 iterations and 20 iterations, 20 iterations outperforms 10 iterations 41 out of 60 cases for Data Diversity and 15 out of 20 for the Hybrid approach. The next combination is 10 iterations and 50 iterations. The results show that 50 iterations outperforms 10 iterations by 39 out of 60 cases and 17 out of 20 cases for Data Diversity and Hybrid approaches respectively. However, when we look at the comparison of 50 iterations and 20 iterations, we see that the two are pretty evenly matched. The results show that for Data Diversity 20 iterations outperforms 50 iterations 31 out of 60 cases, but 50 iterations outperforms 20 iterations 12 out of 20 cases for the Hybrid approach. This leads us to state that 10 iterations is not sufficient for ensemble feature selection in terms of classification performance. However, there is little distinction between 20 iterations and 50 iterations and therefore we recommend using 20 iterations due to the significantly smaller computational requirements of 20 iterations.

Table 4.22: Average AUC: Data Diversity 50 Iterations

Subset	Naive Bayes			MLP			5-NN			SVM			Logistic Regression		
Size	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N	IG	ROC	S2N
10	0.74199	0.74580	0.72708	0.75838	0.75155	0.73597	0.74263	0.73791	0.72398	0.76802	0.75857	0.74234	0.73252	0.74362	0.71380
25	0.76445	0.75467	0.72630	0.76995	0.76375	0.75211	0.76466	0.74676	0.72865	0.78027	0.77170	0.75914	0.72591	0.71080	0.71209
50	0.75280	0.74798	0.71832	0.76803	0.76318	0.75546	0.76649	0.76672	0.74016	0.77393	0.76756	0.75843	0.69573	0.69947	0.69675
100	0.74307	0.74397	0.69662	0.77038	0.75884	0.75649	0.76037	0.75872	0.74284	0.76296	0.75664	0.75635	0.68994	0.68938	0.69663

In addition to looking at the number of iterations across all of the other parameters, we also isolated each parameter and performed further analysis. Looking at the learners, there is no learner where 10 iterations outperforms 20 iterations for either Data Diversity or the hybrid approach. This holds true for 10 iterations and 50 iterations with one exception (5-NN using Data Diversity). When we look at 20 iterations and 50 iterations we see that the results depend on the learner being used. For Data Diversity, 20 iterations outperforms 50 iterations when using 5-NN or Logistic Regression and 50 iterations outperforms 20 iterations for MLP and SVM. In the case of Naive Bayes the two are tied. For the Hybrid approach, 20 iterations is the top performer for Logistic Regression or SVM while 50 iterations is the top performer when using Naive Bayes, MLP, or 5-NN.

In terms of feature subset sizes, there are only two exceptions in which ten is the top performer of the pairwise combinations: 10 outperforms 20 iterations using 50 features for the Hybrid approach and 10 outperforms 50 iterations when using 100 features and Data Diversity. For 20 iterations and 50 iterations the top performer depends on the which feature subset size is being used. For the Hybrid approach 20 iterations out performs 50 iterations when using either 25 features or 100 features while 50 iterations outperforms 20 iterations when using 10 or 50 features. In terms of Data Diversity, 20 iterations outperforms 50 iterations when using either 10 or 100 features and 50 iterations outperforms 20 iterations when using 25 or 50 features.

Lastly we must look at the individual rankers applied in the Data Diversity-based ensembles. When we look at 10 iterations and 20 iterations we see that there is no

Table 4.23: Average AUC: Hybrid 50 Iterations

Subset Size	Naive Bayes	MLP	5-NN	SVM	Logistic Regression
10	0.75030	0.74940	0.75643	0.75867	0.73596
25	0.76091	0.75006	0.76795	0.76347	0.70842
50	0.74693	0.74988	0.77182	0.75666	0.71243
100	0.74029	0.75879	0.76453	0.75895	0.68536

ranker/learner combination in which 10 iterations outperforms 20 iterations with one exception (using AUC with 5-NN). The same hold true for comparing 10 iterations and 50 iterations except there are three exceptions (AUC with Logistic Regression, and S2N with either 5-NN or Naive Bayes). When we look at 20 iterations and 50 iterations we have to go by a case by case basis. The results find that 20 iterations outperforms 50 iterations for the following ranker/learner combinations: S2N with Naive Bayes, 5-NN, or SVM; and AUC with 5-NN or Logistic Regression. 50 iterations will outperform 20 iterations for the following combinations: IG with Naive Bayes, MLP, or 5-NN; and AUC with SVM. The remaining combinations (AUC with Naive Bayes and MLP; S2N with MLP and Logistic Regression; and IG with SVM and Logistic Regression) they are tied.

4.6 CONCLUSIONS

One of the more promising approaches to achieving improved classification performance and more stable feature subsets or gene lists is the use of ensemble feature selection techniques. To date, all existing research explores the creation of the ensemble through the use of data diversity. In this paper we proposed two new approaches to ensemble feature selection in the domain of bioinformatics: functional diversity and a hybrid method which combines data and functional diversity. In addition to introducing these approaches, we performed analysis comparing these two approaches to the existing data diversity approach to determine if the two new techniques are

clearly distinct from the commonly used technique (data diversity) and whether they perform better in terms of classification.

In terms of similarity it was found that functional diversity and hybrid methods were the most similar to one another (versus the other two pairwise comparisons). We believe this is because both approaches utilize an ensemble of feature selection techniques rather than a single feature selection technique. Of the two remaining comparisons, it was found that the data diversity and hybrid methods are the next most similar. We believe this is because both data diversity and hybrid techniques utilize multiple datasets in their approach for ensembles. These observations remain true at all feature subset sizes and when using all twenty-six datasets or isolating the five Moderate datasets or the six Hard datasets. Additionally, it was observed that in general, as the feature subset increases, similarity increases. We believe that as the feature subset size increases the approaches have more room to achieve their similarity. As the results show that both functional diversity and hybrid approaches have little in common with data diversity, this leads us to state that further research into the abilities of these techniques would be of interest as they are clearly not similar to data diversity.

While similarity is an important measure when comparing these techniques to one another, it is not an indicator to how they will perform in terms of classification. Therefore, we performed a classification experiment to compare these techniques in terms of classification performance. We found that in general, when using the Moderate data sets the results show that the ensemble techniques are outperformed by the single run of feature selection in a majority of the cases. However, when looking at the Hard datasets the ensemble techniques outperform the single run of feature selection. We believe that the ensembles are better suited for discovering the harder to learn patterns found in the Hard datasets compared to the single run of feature selection, but the patterns in the Moderate datasets are easier to discover and place

the single run of feature selection in a more even scenario.

In addition, we discovered a trend found with the different learners. Two of the learners favored either the ensemble techniques or the single run of feature selection for both the Moderate and the Hard datasets. With the Naïve Bayes learner the single run of feature selection outperformed the ensemble techniques a majority of the cases. The 5-NN learner had the ensemble techniques outperforming the single run of feature selection in both the Moderate and Hard datasets. The remaining three learners (MLP, SVM, and Logistic Regression) follow the pattern in that the top performers with the Moderate datasets are the single run of feature selection and the top performers of the Hard datasets were the ensemble techniques.

With the ensemble approach data diversity there is another factor to consider, the feature rankers used. Therefore we compared the classification results of the data diversity techniques to when no ensemble approach is applied where the ranker used is the same. Information Gain will in general improve when applied toward the data diversity approach. ReliefF will in general improve with the ensemble approach data diversity except when using the Hard datasets. Mutual Information will improve in general when using the data diversity approach with the Hard datasets but not so with the Moderate datasets. Lastly, ROC will perform better when no ensemble approach is applied than when applied toward the data diversity approach.

We also performed statistical tests to further investigate the properties of the three factors: feature selection technique, feature subset size, and learner. We found that a majority of the feature rankers are statistically indistinguishable from each other. The few rankers which are significantly different from the other rankers are significantly worse, not better. This trend is true for both the Moderate and Hard datasets. This allows us to confidently state that the new techniques discussed in this paper (functional diversity and hybrid approaches) are comparable with the best performances of both data diversity and the feature selection techniques used by

themselves. In terms of the feature subset size we found that the top performances would be found between 10 and 50 features for the Moderate datasets and 350 and 1000 features for the Hard datasets. Lastly, we found that all of the learners are significantly different for the Moderate datasets with SVM being the top performer; but for the Hard datasets the learners are not significantly different from one another with the exception of logistic Regression which is significantly worse.

In terms of feature rank aggregation techniques, it was found that for seven of the techniques (Enhanced Borda, Exponential Weighting, Highest Rank, Mean, Median, Round Robin, and Stability Selection) there is at least one combination of learner and feature subset size which will have the rank aggregation technique as the top performer. Conversely, two of the techniques (Lowest Rank and Robust Rank) did not have a combination which will place the technique as the top performer. Additionally, Lowest Rank was by far the most frequent worst performer and it is recommended not to use this technique for rank aggregation.

The results between the rank aggregation techniques have very little variance. Statistical tests confirm that other than Lowest Rank (which performs worst), all rank aggregation techniques are statistically indistinguishable. This allow us to state that the effect of the choice of rank aggregation technique is minimal and it is recommended to choose a simple and efficient technique over a more complicated one.

In terms of the appropriate number of iterations, our results show that in general, 10 iterations of feature selection is not sufficient for ensemble feature selection. For both the Data Diversity and Hybrid approaches both 20 iterations and 50 iterations outperformed 10 iterations in a majority of the cases. However, in terms of comparing 20 iterations and 50 iterations, there is little distinction between the two in terms of classification performance. The classification performance shows that 20 iterations will outperform 50 iterations by a small margin with Data Diversity and 50 iterations will outperform 20 iterations by a similarly small margin when using the Hybrid

approach. This allows us to recommend that one should use 20 iterations over 50 iterations to maximize classification performance in general due to the much larger computational constraints required for 50 iterations.

In addition to looking at the three different numbers of iterations across all possible combinations, we also observed the performance when isolating each of the individual aspects (learner, subset size, and feature ranker for Data Diversity only). The results show very similar patterns for each of the aspects as with the experiment as a whole. With few exceptions 20 and 50 iterations will outperform 10 iterations even when isolating each aspect. Additionally, in the comparison between 20 and 50 iterations, we find that each is the top performer around 50% of the time as with the experiment as a whole. These trends are found in both the Data Diversity approach and the Hybrid approach..

Overall, this chapter has shown that the new new techniques are comparable to the best of the other feature selection approaches and are distinct enough from data diversity in terms of subset selection to warrant further research. It was also shown that the two new techniques were statistically indistinguishable from the top performers in the Moderate datasets and Hard datasets and the hybrid approach was the top performer for the Hard datasets. In addition to these findings, we observed that ensemble techniques show increased ability to correctly classify instances among the Hard datasets when compared the the feature rankers performed without ensemble approaches. We also found that the choice of feature rank aggregation technique is insignificant as long as one avoids using lowest rank and that the appropriate number of iterations is around 20 iterations and that 10 iterations is insufficient for performing ensemble gene selection.

CHAPTER 5
SIMPLIFYING THE UTILIZATION OF GENE SELECTION AND CLASSIFICATION FOR BIOINFORMATICS DATA

5.1 INTRODUCTION

Due to the large-scale nature of gene databases and modern technology such as DNA microarrays and mass spectrometry, both of which generate large amounts of data with each successful run, working with bioinformatics datasets is a challenging endeavour. Problems like large levels of high dimensionality (having a large number of features for each sample, many of which are irrelevant or redundant) and datasets which are particularly difficult to learn from make properly handling and analyzing such data almost impossible to do without the assistance of a computer.

However, in the domain of machine learning, there are a number of techniques which are designed for working with such challenges. Two technique types that are particularly useful are feature selection and classifiers. Dimensionality reduction techniques such as feature selection seek to reduce the feature set of the dataset by identifying and removing the redundant and irrelevant features (i.e. gene probes) and leaving only the features which contain the most information on the problem at hand. This process has a number of benefits, including reduced computational time, potentially improved classification performance, and creating a subset of features which have been identified as being particularly useful and are a direction for future research.

Classifiers are used to build inductive models for future analysis. The classifier takes known labeled samples and uses the information contained in them to build a

model which can be used to identify properties from unknown samples. Additionally, if a particular model shows good classification performance, then the features used to build said model are extremely important to the problem being modeled.

However, applying machine learning techniques can be a daunting task, especially for researchers who are not practitioners of the domain. There are a large number of decisions to make (i.e. choice of classifier, feature selection technique, and number of features to use) and each decision has many options. If it is possible to either identify an optimum choice ahead of time or make the practitioner's decision matter less to the overall process, then these useful techniques would be easier to utilize in research [41, 111].

5.2 CONTRIBUTIONS

This work is a comprehensive analysis of the classification performance of six classifiers and twenty-four feature selection techniques on twenty-five bioinformatics datasets. We specifically focus on simplifying these choices for four features subset sizes, 50, 75, 100, and 200, as these represent the best balance between producing reliable models (which cannot be easily done with fewer than 50 features) and creating feature sets small enough to be interpreted by humans (demonstrated by few studies considering more than 200 features).

The rest of the paper is organized as follows: Section 5.3 contains discussions of previous research that are relevant to our work. Section 5.4 outlines methods used to conduct our empirical study. In Section 5.5, we present our results with discussions of our findings. Finally, in Section 5.6, we present our conclusions and potential avenues of future study.

5.3 RELATED WORKS

Due to the high dimensionality of bioinformatics datasets, it has become necessary to employ dimension reduction techniques in this domain. A study performed by Inza et al. [65] found that classification performed on reduced feature subsets derived from the original DNA microarray datasets outperformed classification using the whole feature set in a majority of cases and that in addition, feature selection drastically reduced computation time. In particular, univariate (filter-based) feature selection techniques are especially popular in bioinformatics research. There are a number of reasons why these techniques are ideal for this problem, including: the output of the techniques (a ranked list of features) is intuitive and easy to understand; the ranking of genes makes it easy for researchers to further validate the results through laboratory techniques; and the computational time is relatively small when compared to other types of feature selection techniques (filter-based subset evaluation, wrapper, etc.) [96].

The Random Forest algorithm was introduced by Leo Breiman in 2001. His chapter "Random Forest" in the Springer book "Machine Learning" [20] gave a description of the technique and its mechanics, as well as comparing it to other learners. Random Forest had a number of advantages over the other learners, such as robustness to outliers and noise, speed when compared with similar techniques like bagging and boosting, simplicity of implementation, and ability to provide internal information including error, strength, and correlation. In the end, when compared to other learners, Random Forest performed well and was an effective tool in prediction.

Random Forest is a robust and powerful classifier and has been applied to the domain of bioinformatics in the past. The first was a study performed by Diaz-Uriarte et al. [27] in which the Random Forest classifier was applied toward a series of ten DNA microarray datasets focusing on different areas of the body. Another is a 2011 study performed by Dittman et al. [33] which used Random Forest on a pair of

Table 5.1: Dataset List

Name
Ovarian Cancer
Lung Michigan
Lung Cancer
ALL AML Leukemia
Prostate MAT
MLL Leukemia
Lung
DLBCL
BCancer50k
Colon50k
Lymphoma
Acute Lymphoblastic Leukemia
Lymphoma MAT
CNS MAT
Lung50k
Colon
Ovarian MAT
Prostate
Brain Tumor
Lung Cancer Ontario
ECML Pancreas
Breast Cancer
Mulligan-R-NR
Central Nervous System
Spira 2007

DNA microarray datasets with the goal of predicting a patient's response to a drug treatment. Both studies agreed that compared to other classifiers, Random Forest is a powerful classifier which does not require as much parameter adjustment compared to other classifiers.

5.4 METHODOLOGY

In this chapter, we use 24 feature rankers. The feature rankers used are: CS, GR, IG, RF, RFW, SU, SVM-Att, F, OR, Pow, PR, GI, MI, KS, Dev, GM, ROC, PRC, S2N, FCD, WTS, WRS, FS, and SAM. All feature rankers are discussed in Section 2.2.1. In terms of subset sizes we use subset sizes 50, 75, 100, and 200as these represent the best balance between producing reliable models (which cannot be easily done with fewer than 50 features) and creating feature sets small enough to be interpreted by humans

(demonstrated by few studies considering more than 200 features). Additionally, in this work we use the following classifiers: Naïve Bayes, MLP, 5-NN, SVM, Random Forest 100 ,and Logistic Regression all of which are described in Section 2.5.1. To conduct the classification experiments, we use four runs of five-fold cross-validation and the Area Under the Receiver Operating Characteristic curve as the performance metric as detailed in Section 2.6. Lastly we use a series of 25 bioinformatics (see Table 5.1) to test the machine learning techniques. The particulars of all of the datasets can be found in Tables 2.1 and 2.2.

5.5 RESULTS

This section contains the results of our experiments. In this work we used a diverse set of twenty-four rankers and six classifiers to observe how the techniques perform at different feature subset sizes. Tables 5.2 and 5.3 detail the results of our experiment. Each individual value is the average AUC value across the twenty-five datasets where the ranker, classifier, and subset size are kept static. For every combination of subset size and classifier, the top performing value is in **boldface** and the worst is in *italics*. The final row, Range, represents the difference between the top-performing ranker and the worst ranker for that choice of learner and feature subset size.

As we can see from the classification results, we note that one ranker stands out as the most frequent top performing ranker, SVM-Att. For three of the classifiers (MLP, Random Forest 100, and Logistic Regression), SVM-Att is the top performer across all four subset sizes. In the case of 5-NN and SVM it is the top performer for three of the four subset sizes with the exceptions being subset size 200 for 5-NN in which S2N is the top performer and subset size 50 for SVM in which IG is the top performer. In the last classifier, Naïve Bayes, SVM-Att is only the top performer for subset size 200 with the remaining three subset sizes having IG as their top performer. In terms of the worst performing rankers, the most frequent worst performing ranker

Table 5.2: Average AUC for Naïve Bayes, MLP, 5-NN

Ranker	Classifiers											
	Naïve Bayes				MLP				5-NN			
	Feature Subset Size				Feature Subset Size				Feature Subset Size			
	50	75	100	200	50	75	100	200	50	75	100	200
CS	0.87816	0.88116	0.87900	0.87137	0.89419	0.89671	0.89669	0.90438	0.89365	0.89826	0.89663	0.89678
GR	0.86188	0.86058	0.85877	0.85081	0.88119	0.88731	0.88943	0.89785	0.87692	0.88230	0.88556	0.88893
IG	**0.88573**	**0.88359**	**0.88239**	0.87377	0.89830	0.89926	0.90052	0.90495	0.89610	0.89696	0.89912	0.89745
RF	0.86193	0.85867	0.85824	0.85897	0.88170	0.88570	0.88962	0.89522	0.88222	0.88277	0.88577	0.88355
RFW	0.86028	0.86327	0.86170	0.85628	0.88462	0.89146	0.89611	0.90470	0.87665	0.88383	0.88517	0.88784
SU	0.88202	0.88071	0.87912	0.87114	0.89376	0.89473	0.89746	0.90142	0.88862	0.89193	0.89618	0.89726
SVM-Att	0.87608	0.87835	0.87766	**0.88145**	**0.90217**	**0.90583**	**0.90751**	**0.91366**	**0.90098**	**0.89995**	**0.89998**	0.89889
F	0.87781	0.87700	0.87910	0.87712	0.88934	0.89196	0.89461	0.90484	0.88846	0.89387	0.89547	0.89817
OR	0.87363	0.87045	0.87030	0.87081	0.89063	0.89391	0.89542	0.89996	0.88838	0.88769	0.88918	0.89216
Pow	0.85439	0.84954	0.85137	0.85563	0.87752	0.88263	0.88874	0.89950	0.87724	0.87725	0.87656	0.87955
PR	*0.84638*	*0.83794*	*0.84153*	*0.83447*	0.87879	0.87619	0.88401	0.89531	0.86702	0.86911	0.87487	0.87533
GI	0.85770	0.84997	0.84964	0.84210	*0.87515*	0.87604	0.88113	0.89489	*0.86607*	*0.86685*	*0.87337*	*0.87356*
MI	0.87820	0.87594	0.87766	0.87636	0.89094	0.89330	0.89706	0.90396	0.88850	0.89321	0.89236	0.89664
KS	0.88450	0.87914	0.87982	0.87367	0.89732	0.89725	0.89935	0.90199	0.89025	0.89416	0.89755	0.89759
Dev	0.87817	0.87502	0.87323	0.86845	0.88980	0.89107	0.89432	0.89566	0.88752	0.88961	0.89290	0.8951
GM	0.88302	0.88044	0.87928	0.87541	0.89864	0.89940	0.89855	0.90140	0.89093	0.89547	0.89654	0.89844
ROC	0.88243	0.88281	0.87926	0.87604	0.89199	0.89868	0.89842	0.89962	0.89090	0.89715	0.89653	0.89539
PRC	0.87689	0.87734	0.87898	0.87800	0.89124	0.89147	0.89578	0.90069	0.89438	0.89209	0.89121	0.8906
S2N	0.87455	0.86891	0.86228	0.85394	0.89339	0.89951	0.89836	0.90025	0.88940	0.89227	0.89242	**0.89986**
FCD	0.87271	0.87127	0.86915	0.86786	0.88386	0.88856	0.89304	0.89701	0.89163	0.89275	0.89478	0.89449
WTS	0.85671	0.85587	0.85430	0.84763	0.88398	*0.87344*	*0.87029*	*0.85871*	0.88131	0.88418	0.88343	0.89027
WRS	0.88039	0.88152	0.87751	0.87081	0.89154	0.89614	0.89723	0.90075	0.89207	0.89521	0.89858	0.89671
FS	0.87126	0.86651	0.86658	0.85976	0.88337	0.88499	0.88764	0.89020	0.88415	0.88359	0.88525	0.89141
SAM	0.87355	0.87109	0.86926	0.86261	0.88869	0.89271	0.89300	0.89578	0.89955	0.89866	0.89542	0.89807
Range	0.03935	0.04564	0.04086	0.04698	0.02702	0.03238	0.03721	0.05495	0.03491	0.03310	0.02661	0.02630

is GI with just under 50% (11/24) of the cases having GI as the worst performing ranker. In fact, Naïve Bayes is the only classifier that does not have GI as the worst performing ranker for at least one of the four subset sizes. Other poor-performing rankers include: PR (9 cases), WTS (3 cases), and Dev (1 case).

Looking at the best and worst performances of each classifier we see that Random Forest is the top classifier. When we compare the top performances of each classifier, Random Forest has the best classification performance for each feature subset size. In terms of the worst performances for each learner, the worst performance for Random Forest is higher than the worst performances for the other five learners at each feature subset size. As for the worst classifier, it depends on if we look at the best or worst

Table 5.3: Average AUC for Suport Vector Machines, Random Forest 100, Logistic Regression

Ranker	Classifiers											
	SVM				Random Forest 100				Logistic Regression			
	Feature Subset Size				Feature Subset Size				Feature Subset Size			
	50	75	100	200	50	75	100	200	50	75	100	200
CS	0.89230	0.88991	0.88835	0.89741	0.90537	0.90629	0.90710	0.91211	0.82953	0.82872	0.83459	0.85336
GR	0.88322	0.88662	0.88760	0.89501	0.89554	0.90092	0.90481	0.90758	0.83275	0.83243	0.83754	0.84754
IG	**0.89756**	0.89532	0.89371	0.89755	0.90624	0.90762	0.90934	0.91260	0.84188	0.84169	0.84595	0.85
RF	0.88400	0.88708	0.89047	0.89466	0.89810	0.90165	0.90417	0.90669	0.83504	0.83581	0.84100	0.85372
RFW	0.88068	0.88724	0.89102	0.89879	0.89667	0.90406	0.90357	0.90701	0.84702	0.84397	0.85199	0.85993
SU	0.89354	0.89280	0.89381	0.89634	0.90323	0.90802	0.90967	0.91020	0.83679	0.83465	0.83838	0.8467
SVM-Att	0.89708	**0.90438**	**0.90602**	**0.91113**	**0.90856**	**0.91385**	**0.91601**	**0.91934**	**0.87656**	**0.88548**	**0.89277**	**0.89374**
F	0.88784	0.88835	0.89149	0.89677	0.90142	0.90637	0.90752	0.90862	0.83573	0.83262	0.83227	0.84635
OR	0.89329	0.89377	0.89204	0.90012	0.89969	0.90203	0.90390	0.90626	0.83658	0.83882	0.84448	0.85594
Pow	0.88092	0.88368	0.88960	0.89402	0.89433	0.89716	0.90129	0.90428	0.83293	0.83279	0.83202	0.83851
PR	*0.87547*	0.87873	0.88363	0.88934	*0.88855*	*0.89255*	0.89576	0.89874	0.83011	*0.82633*	0.83645	*0.83344*
GI	0.87577	*0.87632*	*0.88197*	*0.88784*	0.89168	0.89507	*0.89330*	*0.89828*	*0.82857*	0.83247	0.83575	0.8368
MI	0.88941	0.88782	0.88951	0.89781	0.90196	0.90487	0.90407	0.90740	0.83904	0.83344	0.83138	0.85317
KS	0.89521	0.89121	0.89224	0.89822	0.90711	0.90959	0.90764	0.90906	0.84286	0.84047	0.84443	0.85428
Dev	0.89172	0.88897	0.88886	0.89031	0.89741	0.90078	0.90145	0.90520	0.83593	0.83255	*0.83060*	0.8415
GM	0.89451	0.89369	0.89327	0.89831	0.90451	0.90465	0.90729	0.90983	0.84017	0.84032	0.84605	0.85436
ROC	0.89479	0.89656	0.89726	0.89765	0.90456	0.90537	0.90653	0.90727	0.84728	0.84726	0.84726	0.84989
PRC	0.89346	0.89640	0.89613	0.89567	0.90347	0.90498	0.90508	0.90780	0.84684	0.84536	0.84806	0.85574
S2N	0.89722	0.89627	0.89595	0.89833	0.90320	0.90399	0.90512	0.90799	0.84802	0.84790	0.85033	0.85691
FCD	0.88627	0.89032	0.89210	0.89664	0.90004	0.90255	0.90336	0.90516	0.84171	0.84245	0.84471	0.84449
WTS	0.89114	0.89238	0.88950	0.89646	0.89923	0.90108	0.90159	0.90645	0.84300	0.83999	0.84351	0.84878
WRS	0.89556	0.89440	0.89780	0.89773	0.90496	0.90729	0.91142	0.90805	0.84651	0.84479	0.84987	0.85316
FS	0.88912	0.88867	0.88747	0.89171	0.89750	0.89887	0.90222	0.90491	0.83940	0.84348	0.83903	0.84187
SAM	0.89282	0.89365	0.89480	0.89650	0.90279	0.90538	0.90882	0.90783	0.84874	0.84690	0.84884	0.85135
Range	0.02208	0.02806	0.02404	0.02329	0.02001	0.02130	0.02271	0.02106	0.04799	0.05914	0.06218	0.06029

performances of each classifier. When we look at the top performances for each classifier, we see that for three of the feature subset sizes (75, 100, and 200) Naïve Bayes has the lowest classification performance with Logistic Regression having the lowest classification performance at subset size 50. When we look at the worst performance for each classifier, Logistic Regression has the lowest classification performance for all four subset sizes.

One of the more interesting trends we found was that certain classifiers are very resilient to changes in the classification performance when using different rankers. Of the six classifiers, Random Forest has the smallest range of classification results for

Table 5.4: ANOVA Results: Classifiers

	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
50 Features	Classifiers	26.8	5	5.36029	223.93	5.33×10^{-238}
	Error	1723.31	71994	0.02394		
	Total	1750.11	71999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
75 Features	Classifiers	30.46	5	6.09174	254.77	7.52×10^{-271}
	Error	1721.42	71994	0.02391		
	Total	1751.88	71999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
100 Features	Classifiers	29.7	5	5.94031	251.18	5.01×10^{-267}
	Error	1702.63	71994	0.02365		
	Total	1732.33	71999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
200 Features	Classifiers	29.28	5	5.85617	251.43	2.72×10^{-267}
	Error	1676.85	71994	0.02329		
	Total	1706.13	71999			

all four subset sizes and never rises above a difference of 0.02271 between its best and worst rankers. In terms of the largest range, Logistic Regression has the largest range among the six classifiers for all four subset sizes which has a maximum of 0.06218.

5.5.1 Statistical Analysis

In order to further validate the results in our classification experiments, we performed a series of one factor ANalysis Of VAriance (ANOVA) tests [14]. Additionally we performed Tukey's Honestly Significant Different (HSD) test [14] for each of the factors that were determined to be significant from the ANOVA results. Both the ANOVA and Tukey's HSD tests were performed in MATLAB.

Looking first at the classifiers (Table 5.4), we performed the ANOVA analysis with the factor being the choice of classifier for each feature subset size. The ANOVA

Table 5.5: ANOVA Results: Rankers - RF100

	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
50 Features	Rankers	0.287	23	0.01247	0.62	0.9205
	Error	241.619	11976	0.02018		
	Total	241.906	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
75 Features	Rankers	0.252	23	0.01096	0.55	0.9602
	Error	239.639	11976	0.02001		
	Total	239.891	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
100 Features	Rankers	0.259	23	0.01127	0.57	0.9508
	Error	237.832	11976	0.01986		
	Total	238.091	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
200 Features	Rankers	0.199	23	0.00864	0.44	0.99
	Error	233.684	11976	0.01951		
	Total	233.882	11999			

results found that the factor was significant for all four feature subset sizes. This is indicated by the factor achieving a Prob>F score of below 0.05 for each feature subset size. As the factor was significant, we performed the Tuley's HSD test for the classifiers for each subset size (Figures 5.1 through 5.4). The results show that Random Forest is significantly better than the other five classifiers for each feature subset size. Classifiers MLP, SVM, and 5-NN are statistically indistinguishable for all of the subset sizes with the exception of size 200 where MLP is significantly better than 5-NN. Lastly for all of the subset sizes, Logistic Regression is significantly worse than the other five classifiers and Naïve Bayes is only significantly better than Logistic Regression.

When we look at the rankers, we performed the ANOVA test with the factor of the rankers when we keep the subset size and choice of classifier static. We are

Table 5.6: ANOVA Results: Rankers - SVM

	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
50 Features	Rankers	0.498	23	0.02165	0.96	0.5127
	Error	269.545	11976	0.02251		
	Total	270.043	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
75 Features	Rankers	0.409	23	0.01776	0.79	0.7491
	Error	269.561	11976	0.02251		
	Total	269.969	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
100 Features	Rankers	0.281	23	0.0122	0.55	0.9583
	Error	264.782	11976	0.02211		
	Total	265.062	11999			
	Source	Sum Sq.	d.f	Mean Sq.	F	Prob>F
200 Features	Rankers	0.224	23	0.00975	0.47	0.986
	Error	250.912	11976	0.02095		
	Total	251.136	11999			

presenting the results for Random Forest (the top performing classifier) and SVM (chosen because while not significantly different from MLP and 5-NN it shows slightly better classification performance for the preferred smaller subset sizes). We can see in Tables 5.5 and 5.6 that for both classifiers, the choice of ranker is not statistically significant for all four subset sizes. This is indicated by the Prob>F score of above 0.05 for all feature subset sizes and the two classifiers. These results allow us to state that the choice of ranker is of little consequence.

5.6 CONCLUSIONS

Machine learning is a valuable tool in working with large and high dimensional datasets prevalent in domains such as bioinformatics. However, choosing the appropriate techniques to utilize in research can be a daunting task, especially if the

Figure 5.1: Tukey's HSD Results: Classifiers - Subset Size 50

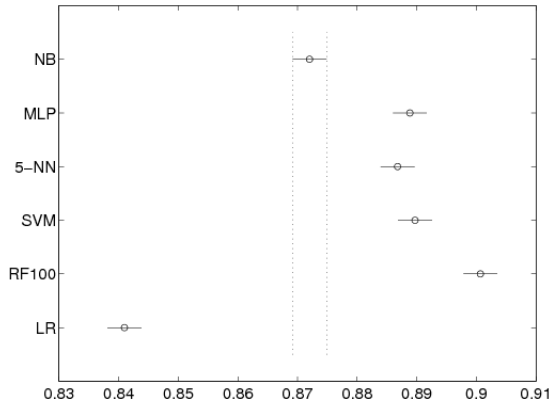
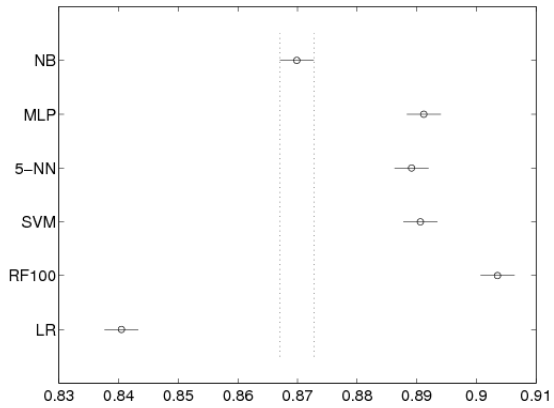


Figure 5.2: Tukey's HSD Results: Classifiers - Subset Size 75



researcher in not a practitioner of machine learning. If we can narrow the number of choices required for applying these techniques, we can make machine learning more accessible for bioinformatics studies. This study analyzes six classifiers, and twenty-four feature rankers on twenty-five bioinformatics datasets. We analyzed these factors individually for four feature subset sizes chosen for being appropriate for building the models and small enough to be used effectively for future research.

Our results show that of the six classifiers, Random Forest with 100 trees is the top performer for all four feature subset sizes. This is also confirmed through statistical analysis. In terms of the rankers, the most frequent top performer is SVM-Att. However, statistical analysis shows that when using Random Forest with 100 trees

Figure 5.3: Tukey's HSD Results: Classifiers - Subset Size 100

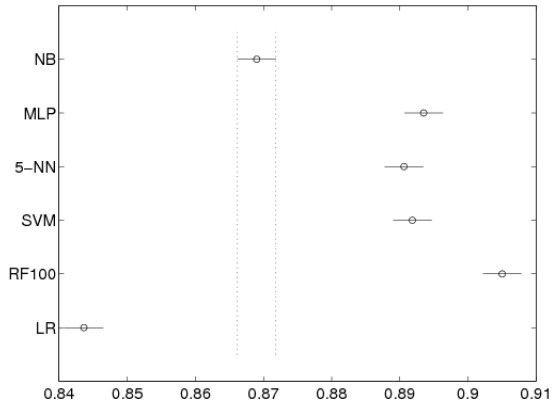
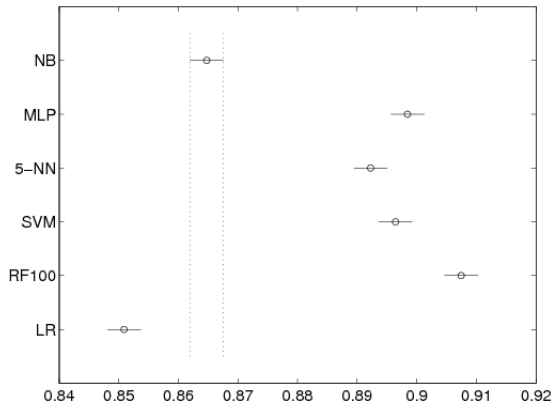


Figure 5.4: Tukey's HSD Results: Classifiers - Subset Size 200



(or when using another learner, SVM), no feature ranker is significantly better or worse than any other ranker. Thus, the choice of ranking is of little importance, so long as feature selection occurs to reduce the feature set to a more manageable number.

Future work in this area includes focusing on other applications of these datasets (i.e. patient response prediction) in order to see if the goal of the experiment changes the patterns discovered. Another area is applying these techniques on particularly difficult to learn from datasets to see if the same patterns emerge.

CHAPTER 6
COMBINING ENSEMBLE LEARNING AND FEATURE SELECTION TO IMPROVE CLASSIFICATION PERFORMANCE

6.1 INTRODUCTION

DNA microarray datasets contain many challenging properties, all of which increase the necessity of advanced data mining techniques such as ensemble classification. Perhaps the most prevalent problem associated with DNA microarray datasets is an overabundance of available data. The DNA microarray has allowed researchers to test for the reaction of tens of thousands of genes simultaneously. This results in the problem of high dimensionality. High dimensionality occurs when there are a large number of features (or genes) for each sample or instance. This fact makes these datasets particularly difficult to work with and can even prevent some methods of classification from working effectively due to the large computational costs. Also, many of the features are irrelevant or redundant to the problem being researched, making feature selection necessary even if an algorithm could handle the large quantity of data. In addition, many DNA microarray datasets are inherently noisy, with outliers and difficult-to-learn class boundaries which make any model-building more difficult. The presence of this noisy data can cause the results to become inaccurate and unreliable. Lastly, class imbalance (uneven distribution of samples across the classes) is a frequently occurring problem with DNA microarray datasets [96].

These problems make bioinformatics an ideal place to apply ensemble classification approaches. Ensemble classification is the process of generating multiple classification models and then aggregating their decisions into a single final decision. There are

a number of benefits of ensemble classification such as reduced overfitting, improved classification performance, and a reduction of bias. Additionally, ensemble learning approaches are versatile due to their ability to incorporate a variety of classifiers and data pre-processing techniques into their algorithms [69].

Two popular ensemble learning techniques are Bagging and Boosting. Bagging takes a random sample of instances with replacement from the training dataset so that it creates a new dataset made up from instances of the training dataset. This process is repeated multiple times and the classifiers are trained using the new datasets and the final result is made of a majority vote of the trained classifiers. Boosting begins with the training dataset and gives an initial identical weight to each instance. Upon the training and testing of the classifier built, the misclassified instances are given more weight and the correctly classified instances are given less weight. These new weights are used to directly give more weight to the instances in the new training data (Boosting by reweighting) or they are used in a weighted sampling with replacement process which creates a new training dataset where the misclassified instances are more likely to show in the new training dataset than the correctly classified ones (Boosting by resampling). The process repeats using the new training dataset and the overall process is repeated a predetermined number of times. The final decision is a weighted majority vote of all the trained classifiers. However, both the Bagging and Boosting algorithms do not take into account the inherent high-dimensionality commonly found in bioinformatics dataset or any data pre-processing techniques such as feature selection to combat said high-dimensionality.

6.2 CONTRIBUTIONS

Our work focuses on the application of two relatively new ensemble approaches, Select-Bagging and Select-Boosting (the Bagging and Boosting algorithms with feature selection incorporated into each iteration of their respective algorithms), on balanced (no

dataset has less than a 43.50% minority class distribution) bioinformatics datasets. We test these two approaches using a series of seven balanced bioinformatics datasets, three feature rankers, four subset sizes, and two classifiers. Additionally, to better observe the absolute effect of ensemble learning, we also observed the results when no ensemble approach is applied (denoted as No-Ensemble in this work). To our knowledge, this is the first empirical study which focuses on balanced datasets and utilizes Select-Bagging and introduces Select-Boosting.

The rest of the paper is organized as follows: Section 6.3 contains discussions of previous research that are relevant to our work. Section 6.4 outlines methods used to conduct our empirical study. In Section 6.5, we present our results with discussions of our findings. Finally, in Section 6.6, we present our conclusions and potential avenues of future study.

6.3 RELATED WORKS

Ensemble learning is the process of combining decisions of multiple classification models into a single final result [79]. The main objective of ensemble methods is not only improving overall classification performance [28] but also more accurate generalization capability in classifying unseen instances [121]. There are two key factors that affect ensemble method performance: the accuracy and the diversity of the base classifiers [28]. In this study, our focus is on the two most popular ensemble techniques: Bagging [18] and Boosting [53].

Several scholars have investigated both Bagging and Boosting in their works. For example, Nagi et al. [87] conducted an empirical study using nine high-dimensional cancer datasets and three classifiers. The researchers proposed a new ensemble method and compared class-specific accuracy of their method versus each single classifier as well as Bagging and Boosting. Another work by Tan et al. [102] used seven cancer gene expression datasets along with the C4.5 decision tree classifier, and

two ensemble methods: Bagging and Boosting with decision trees as the classifier. Chen [23] conducted an experiment using eight microarray datasets and one feature selection technique, Relief-F.

In 2014, our research group introduced the Select-Bagging ensemble method [31]. In this work we observed how Select-Bagging performed compared to when no ensemble approach is applied. We used a single classifier and two feature selection techniques in our case study. Our results showed that Select-Bagging performs significantly better than when no ensemble approach is applied.

However, there are a number of shortcomings found within these studies. Nagi et al. [87] did not employ feature selection and chose their three learners based on classification results from a series of datasets which are not high dimensional and are not representative of bioinformatics datasets. Tan et al. [102] applied feature selection but it was deployed outside 10-fold cross-validation. In addition, they applied one run of 10-fold cross-validation to some datasets but not all and they chose different feature subset sizes for different datasets. Chen [23] performed feature selection outside the ensemble methods (it causes overfitting of the classification models) and they did not provide any information on the class distribution of those datasets or how many features were selected for their experiment. As a result of these shortcomings, the results provided may be called into question. Even our own work can be considered preliminary, as it only discusses Select-Bagging for ensemble approaches.

Contrary to these studies, our current work addresses each of these concerns. We are comparing different ensemble approaches in addition to no ensemble. We also used seven high-dimensional and balanced datasets, three feature rankers from three different families of feature selection methods along with four feature subset sizes, and two classifiers using four runs of five-fold cross validation. In addition, we performed feature selection within each run and each fold of the cross-validation process (as well as within each iteration of the ensemble approaches) to avoid overfitting of the built

Table 6.1: Dataset List

Name
DLBCL
Prostate
Breast Cancer
DLBCL NIH
BCancer50k
Spira2007
SotiriouMatrixData-Grade

classification models. Lastly, all our results are validated by statistical analysis.

6.4 METHODOLOGY

In this chapter, we focus on two ensemble approaches, Select-Bagging and Select-Boosting, along with classification using no ensemble approach, denoted as No-Ensemble. The ensemble approaches are described in Section 2.5.2. The 3 feature rankers used in this chapter are: IG, ROC, and S2N. All feature rankers are discussed in Section 2.2.1. In terms of subset sizes we use subset sizes 25, 50, 100, and 200 as they are an appropriate collection of feature subset sizes. Based on preliminary research, we found that these sizes present the best balance of building usable models and producing a manageable feature subset. Additionally, in this work we use the following classifiers: 5-NN and LR all of which are described in Section 2.5.1. To conduct the classification experiments, we use four runs of five-fold cross-validation and the Area Under the Receiver Operating Characteristic curve as the performance metric as detailed in Section 2.6. Lastly we use a series of 7 bioinformatics (see Table 6.1) to test the ensemble approaches. The particulars of all of the datasets can be found in Tables 2.1 and 2.2.

Table 6.2: Classification Results - Ensemble Approaches

Classifier	Subset Size	IG			ROC			S2N		
		No-Ensemble	S.Boosting	S.Bagging	No-Ensemble	S.Boosting	S.Bagging	No-Ensemble	S.Boosting	S.Bagging
5-NN	25	*0.82121*	0.82375	**0.84697**	*0.80937*	0.82276	**0.83415**	*0.81288*	0.82486	**0.82842**
	50	0.82789	*0.82569*	**0.84477**	*0.81275*	0.82319	**0.82822**	0.81767	*0.81341*	**0.82512**
	100	0.83130	*0.83110*	**0.84523**	*0.81859*	0.82791	**0.83168**	*0.82446*	0.82518	**0.83114**
	200	*0.82857*	0.82939	**0.84125**	*0.81538*	0.82907	**0.83397**	0.82805	*0.82256*	**0.83175**
LR	25	*0.79156*	0.80923	**0.82071**	*0.79248*	0.81611	**0.83131**	*0.79787*	0.81141	**0.82803**
	50	*0.75411*	0.79538	**0.81517**	*0.76495*	0.80321	**0.82003**	*0.75487*	0.80897	**0.81702**
	100	*0.73920*	0.79938	**0.81375**	*0.75136*	0.80988	**0.81659**	*0.74286*	**0.81199**	0.80987
	200	*0.73927*	**0.81779**	0.81197	*0.75384*	0.82442	**0.82668**	*0.73874*	**0.82238**	0.80826

6.5 RESULTS

In this work, we seek to determine whether Select-Bagging or Select-Boosting is better suited for bioinformatics datasets using a series of seven balanced bioinformatics datasets. Additionally, we compare the two approaches to classification models built with no ensemble approach (No-Ensemble). In order to test these three approaches, we use three feature rankers, two classifiers, and four subset sizes. Table 6.2 contains the results of our experiment. Each entry in the table is the average AUC value across all seven datasets for every combination of ensemble approach, classifier, feature ranker, and feature subset size. The best ensemble approach for each combination of classifier, feature ranker, and feature subset size will be in **boldface** and the worst performing approach in *italics*.

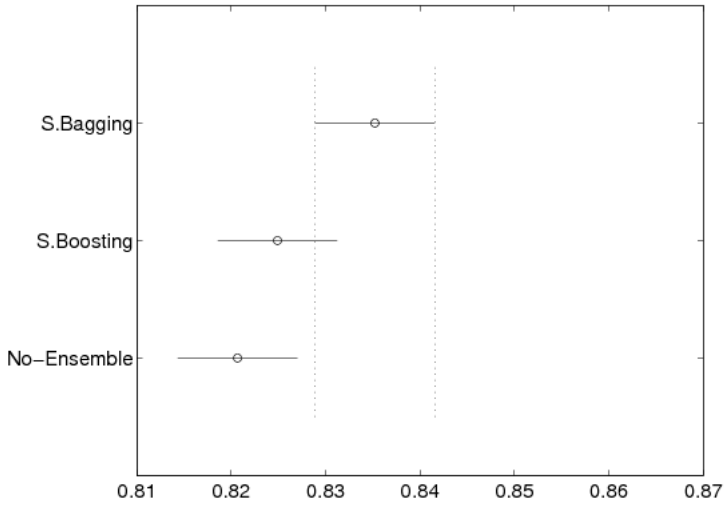
Looking at 5-NN (top portion of Table 6.2), we see that Select-Bagging is the top performing ensemble approach for 12 out of 12 scenarios. It should also be noted that the approach of No-Ensemble is the worst performing approach for the ROC feature ranker. In the case of Information Gain and Signal-to-Noise, No-Ensemble is the worst performing approach in 50% of the scenarios. Specifically, No-Ensemble is the worst performing approach for subset sizes 25 and 200 with Information Gain and subset sizes 50 and 200 with Signal-to-Noise.

For Logistic Regression (bottom portion of Table 6.2), we see that unlike 5-NN,

Table 6.3: ANOVA Results: Ensemble Approaches

Classifier	Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
5-NN	Approach	0.188	2	0.09407	3.84	0.0215
	Error	123.374	5037	0.02449		
	Total	123.563	5039			
LR	Approach	3.37	2	1.68492	64.39	2.43E-28
	Error	131.798	5037	0.02617		
	Total	135.168	5039			

Figure 6.1: Tukey's HSD Results: Ensemble Approaches For Each Classifier: 5-NN

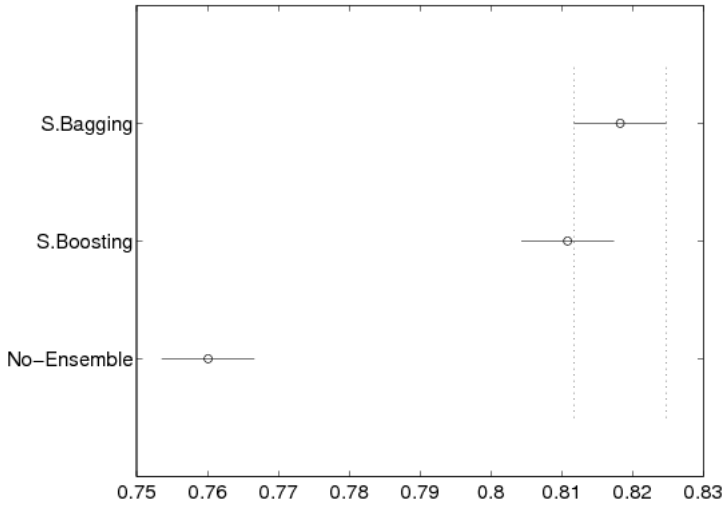


Select-Bagging is the top performing ensemble approach in 9 out of 12 scenarios. In terms of the three exceptions, Select-Boosting is the top performing approach, followed by Select-Bagging. Once again we see that No-Ensemble is the most frequent worst performing approach for the classifier for all 12 scenarios for Logistic Regression.

6.5.1 Statistical Analysis

In order to further validate the results in our classification experiments, we performed two one-factor ANalysis Of VAriance (ANOVA) tests [14] (one for each classifier) with the choice of ensemble learning approach being the factor, across the seven datasets to determine if it has any statistically significant effect on the AUC levels. The

Figure 6.2: Tukey's HSD Results: Ensemble Approaches For Each Classifier: LR



results of the ANOVA tests (as seen in Table 7.5) show that the choice of ensemble approach is a significant factor for both classifiers. This is indicated by the Prob>F value being less than 0.05. Additionally, we performed a multiple comparison test using Tukey's Honestly Significant Difference (HSD) test [14]. Figures 6.1 and 6.2 contain the results of the Tukey's HSD tests. The results show that for 5-Nearest Neighbor, Select-Bagging is significantly better than No-Ensemble and better than Select-Boosting (but not to a statistically significant degree). In terms of Logistic Regression, both Select-Boosting and Select-Bagging are significantly better than No-Ensemble but are not significantly better than each other, with Select-Bagging being the top performing approach. Thus, we can recommend that the inclusion of an ensemble approach is beneficial and that we recommend using Select-Bagging as it is always significantly better than No-Ensemble and better than Select-Boosting for both classifiers.

6.6 CONCLUSIONS

Ensemble learning combines the power of multiple models into a single decision. Benefits from ensemble learning can include reduced overfitting and increased classification performance which makes ensemble learning a potential useful tool for bioinformatics. However, many ensemble approaches, such as Bagging and Boosting, do not take into account the inherent high-dimensionality found in these datasets. Thus we developed two new ensemble approaches, Select-Bagging and Select-Boosting, which incorporate the feature selection process into each iteration of their algorithms. In this work, we seek to determine whether Select-Bagging or Select-Boosting is best suited for bioinformatics datasets. Additionally, we include the results of the same experiments but with no ensemble approach applied in order to determine if the utilization of the ensemble learning approach is beneficial. We test the techniques using a series of seven balanced bioinformatics datasets along with three feature rankers, two classifiers, and four subset sizes.

Our results show that Select-Bagging is the most frequent top performing ensemble approach for both classifiers. Of the possible 24 scenarios, only three do not have Select-Bagging as the top performing approach, with Select-Boosting being the top performing approach for those scenarios. Additionally, the most frequent worst performing approach is No-Ensemble producing the worst performance for 20 out of the 24 scenarios. Statistical analysis shows that Select-Bagging is significantly better than No-Ensemble and better (though not significantly better) than Select-Boosting for both classifiers. Thus, it is our recommendation that Select-Bagging significantly improves the classification performance for balanced bioinformatics datasets. Future work may include the inclusion of more datasets, especially those for more specific purposes (tumor classification, patient response prediction, etc), to see if our findings remain valid.

CHAPTER 7

DATA SAMPLING PROCESS FOR BIOINFORMATICS DATASETS

7.1 INTRODUCTION

Bioinformatics datasets have a number of challenges to overcome, such as high-dimensionality, difficult to learn class boundaries, and noisy data. However, a commonly ignored issue in this domain is that of class imbalance, where one class has more instances than the other class(es) in the dataset despite it being a frequent problem [11]. A possible reason why class imbalance tends to affect classification performance may be due to the fact that many classification algorithms assume that the classes will have an equal number of instances in the dataset [64]. This assumption can lead to some serious problems, including increased bias towards the majority class (whereas the class of interest is frequently the minority class) and an increased number of misclassifications [5]. One recommendation for combating some of these issues is applying data sampling methods.

Data sampling transforms the dataset into a more balanced dataset through the addition or removal of instances. There are a number of different forms that data sampling can take, each with their own benefits and detriments. The general data sampling process can be broken down into three aspects: data sampling technique, data sampling approach, and post-sampling class distribution ratio.

Data sampling technique refers to the specific algorithm which balances the dataset. In general, data sampling can be broken down into two different categories: undersampling and oversampling. Undersampling refers to removing instances from the majority class so as to achieve a more balanced class ratio for the dataset. Oversam-

pling occurs when instances are added to the minority class to create a more balanced class ratio. The addition to the minority class can be performed through the duplication of instances or the creation of synthetic instances based on the original minority class instances. Both undersampling and oversampling are valid options for balancing the datasets and should be considered.

Data sampling approach refers to the process of combining data sampling, feature selection, and classification. In particular it describes the order and to what extent the three pieces are utilized. The main component is the general order of the preprocessing techniques: either feature selection is performed first or data sampling is performed first. If data sampling is performed before feature selection, then one must decide whether to use the unsampled or the sampled data to build the classification model. These decisions must be decided in order to perform data sampling alongside feature selection.

Post-sampling class distribution ratio refers the the distribution of the two classes after data sampling occurs. While perfectly balanced is the logical choice for data sampling, similar results may be achieved with a less aggressive post-sampling class distribution ratio. The main benefit of the less aggressive post-sampling class distribution ratio is that the dataset is changed to a smaller degree when using that ratio. This means that there will be less data loss (when undersampling) or less overfitting (when using oversampling) and the model is more reliable.

7.2 CONTRIBUTIONS

In this study we focus on determining which options of the three aspects tend to be the best suited for bioinformatics data analysis. To this end, we test all combinations of three data sampling techniques (Random Undersampling, Random Oversampling, and Synthetic Minority Oversampling TEchnique or SMOTE), three data sampling approaches (data sampling performed before feature selection and using the unsam-

pled data to build the model, data sampling performed before feature selection and using the sampled data to build the model, and data sampling performed after feature selection and using the sampled data to build the model), and two post-sampling class distribution ratios (50:50 and 35:65 minority class:majority class) in terms of classification performance. In order to test these factors, we utilize a collection of fifteen imbalanced bioinformatics datasets along with a three diverse feature selection techniques, four feature subset sizes, and six classifiers. It should be noted that we are only discussing the three aspects of data sampling and factors such as feature selection technique, feature subset size, and classifier, are not being analyzed.

The rest of the paper is organized as follows: Section 7.3 contains discussions of previous research that are relevant to our work. Section 7.4 outlines methods used to conduct our empirical study. In Section 7.5, we present our results with discussions of our findings. Finally, in Section 7.6, we present our conclusions and potential avenues of future study.

7.3 RELATED WORKS

In 2005, Al-Shahib et al. [5] performed a study which utilized data sampling with multiple class ratios. The goal of the work was to see if the addition of a data sampling technique could improve the classification performance of protein function prediction. Their single dataset consisted of 1,151 proteins (instances) from thirteen different functional groups with each of the protein samples being represented by a feature space of 433 features. For the data sampling they used random undersampling and five different levels of undersampling which ranged from perfectly balanced to no data sampling being applied. Their findings were that the addition of the data sampling technique did improve classification performance, so as long as the data sampling was used to perfectly balance the two classes (a final class ratio of 50:50).

However, in that study there are a number of factors which indicated that it is a

very preliminary work and may not be applicable to other datasets. First, there is only a single dataset being used (though they did use multiple classification schemas) which makes it hard to claim that your findings will be valid when using other datasets. Additionally, the dataset has only 433 features and uses a wrapper-based feature selection technique. This is a problem because wrapper-based techniques are frequently computationally infeasible to use for very high-dimensional datasets. Another issue is that Al-Shahib et al. only uses a single data sampling technique, which may not be the most appropriate for the situation. Lastly, their only source of variability is applied to the test dataset, through sampling with replacement. This indicates that the model building process is not tested for variability and therefore is not throughly tested.

In 2012, Blagus et al. [16] performed a study which included utilizing data sampling techniques on high dimensional bioinformatics datasets. In the work, they used random undersampling and SMOTE as their data sampling techniques. Additionally, they used a collection of three breast cancer datasets, each with two binary classification schemas (two distinct binary-class attributes) for a total of six datasets. The balance levels of the datasets ranged from 14% to 45% and they used the t-statistic for feature selection, with the features ranked based on their values and the top 40 features used for classification. The results of the work were that only the k-NN classifier seemed to benefit significantly from SMOTE, and this benefit was larger as the number of neighbors used for the k-NN model was increased. However, for most of the other classifiers, it seemed that random undersampling was more useful than SMOTE.

However, that is also a relatively a preliminary work and may not be applicable to other datasets. First, some of their datasets are relatively balanced. Applying data sampling to balanced datasets will not change the datasets in any significant way as they are already almost balanced and therefore is not a good candidate for testing a

data sampling technique. Second, there are only three datasets being used (though they did use multiple classification schemas), which, while not as detrimental to the generalizability as only using a single dataset as Al-Shahib et al. [5] did,is still a very small collection of datasets to base general findings on.

In 2014, our research group performed a series of studies which focused on the data sampling process in their individual aspects. Dittman et al. [42] focused on using three data sampling techniques (the same three used in this work, see Section 2.3.1) and applied them to a collection of six imbalanced datasets, one feature selection technique, and two classifiers as a preliminary work to determine which technique is the most appropriate for bioinformatics data. The results showed that random undersampling was most frequently the top performing technique, but subsequent statistical analysis showed that the three techniques are statistically indistinguishable from each other.

Khoshgoftaar et al. [74] focused on the data sampling approaches, or where in the classification process should data sampling be applied, and the post-sampling class distribution ratio. The study used the same three data sampling approaches and two post-sampling class distribution ratios used in this work (see Section 2.3.2 and Section 2.3.3 for more details). As a preliminary test of these approaches and ratios, random undersampling along with four feature selection techniques and a single feature subset size were used. It was found that data sampling followed by feature selection and using the unsampled data to build the model outperformed the other two approaches and statistical analysis confirmed that this difference is significant.

Another paper by Dittman et al. [29] focused on the combination of data sampling technique and post-sampling class distribution ratio. That work looked at the same three data sampling techniques and post-sampling class distribution ratio used in this work (see Section 2.3.1 and Section 2.3.3 for more details). It used a collection of seven imbalanced bioinformatics datasets along with a single data sampling approach, three

feature selection techniques, one feature subset size, and six classifiers. Additionally, it included the results from when no data sampling was performed. It was found that random undersampling and 35:65 class ratio was the preferred combination, though statistical analysis showed that none of the combinations were significantly better than the others. Additionally, it was shown that at no point does no data sampling outperform the top performing data sampling technique for any classifier or feature selection technique, though it was not significantly worse.

In contrast to these previous preliminary works, this paper presents is an extensive study which focuses on all three aspects of the data sampling process (data sampling technique, data sampling approach, and post-sampling class distribution ratio) at the same time. Additionally, we use a diverse collection of fifteen imbalanced bioinformatics datasets, three feature selection techniques, and four feature subset sizes. We also use four runs of five-fold cross validation to throughly test the variability of the technique. It should be noted that we do perform both the data sampling and feature selection procedures on each training dataset generated by the cross validation process.

7.4 METHODOLOGY

In this chapter, we focus on three aspects of data sampling: data sampling technique, data sampling approach, and post-sampling class distribution ratio. We use three data sampling techniques (RUS, ROS, and SMOTE), three data sampling approaches (DS-FS:UnSam, DS:FS-Sam, and FS-DS) and two post-sampling class distribution ratio (50:50 and 35:65). The three aspects of data sampling are described in Section 2.3. The 3 feature rankers used in this chapter are: IG, ROC, and S2N. All feature rankers are discussed in Section 2.2.1. In terms of subset sizes we use subset sizes 25, 50, 100, and 200 as they are an appropriate collection of feature subset sizes. Based on preliminary research, we found that these sizes present the best balance of building

Table 7.1: Dataset List

Name
Brain Tumor
ECML Pancreas
GSE1456
GSE20271
GSE25055
GSE25065
GSE3494-GPL96-ER
GSE3494-GPL96-Grade
GSE3494-GPL97-ER
GSE3494-GPL97-Grade
Lung 50k
Ovarian MAT
Raponi 2007 No SD
Raponi 2007 R+SD
Watanabe 2006

usable models and producing a manageable feature subset. Additionally, in this work we use the following classifiers: NB, MLP, 5-NN, SVM, RF100, and LR, all of which are described in Section 2.5.1. To conduct the classification experiments, we use four runs of five-fold cross-validation and the Area Under the Receiver Operating Characteristic curve as the performance metric as detailed in Section 2.6. Lastly we use a series of 15 imbalanced bioinformatics (see Table 7.1) to test the data sampling process. The particulars of all of the datasets can be found in Tables 2.1 and 2.2.

7.5 RESULTS

In this work, we seek to identify the best practices for utilizing data sampling on imbalanced bioinformatics datasets. To this end, we use three data sampling techniques, three data sampling approaches, two post-sampling class distribution ratios, three feature selection techniques, four feature subset sizes and six classifiers. Tables 7.2 through 7.4 contain the results of our experiments. Each table utilizes a

Table 7.2: Classification Results: IG

Learner	DS Technique	DS-FS:Unsam				DS-FS:Sam				FS-DS			
		25	50	100	200	25	50	100	200	25	50	100	200
NB	RUS:35	0.7561	0.7602	**0.7653***	0.7521	0.7582	0.7596	0.7546	0.7545	0.7541	0.7609	0.7553	0.7552
	RUS:50	0.7516*	0.7514	0.7470	0.7366	0.7429	0.7420	0.7480	0.7449	0.7503	0.7516	0.7514	0.7438
	ROS:35	0.7569	0.7627*	0.7553	0.7503	0.7552	**0.7606**	0.7547	0.7514	0.7569	0.7627	0.7613	0.7588
	ROS:50	0.7521	0.7577	0.7609	**0.7584**	0.7519	0.7571	**0.7587**	**0.7576**	**0.7574**	**0.7673***	**0.7647**	**0.7598**
	SMOTE:35	**0.7679***	**0.7634**	0.7612	0.7472	**0.7627**	0.7541	0.7519	0.7489	0.7524	0.7600	0.7587	0.7544
	SMOTE:50	0.7521	0.7526	0.7564	0.7533	0.7506	0.7428	0.7508	0.7469	0.7549	0.7618*	0.7562	0.7504
MLP	RUS:35	0.7487	0.7443	0.7448	0.7452	**0.7462**	**0.7503**	0.7505	0.7531*	0.7434	0.7450	0.7503	0.7472
	RUS:50	0.7411	0.7409	0.7454	0.7474	0.7426	0.7426	**0.7595**	**0.7670***	**0.7506**	**0.7456**	**0.7564**	**0.7568**
	ROS:35	0.7460	0.7469	0.7454	0.7522*	0.7353	0.7362	0.7418	0.7481	0.7391	0.7305	0.7382	0.7387
	ROS:50	0.7300	0.7417	**0.7495**	**0.7581***	0.7281	0.7401	0.7430	0.7553	0.7361	0.7330	0.7386	0.7372
	SMOTE:35	0.7496	**0.7513***	0.7477	0.7504	0.7439	0.7398	0.7408	0.7447	0.7389	0.7326	0.7382	0.7384
	SMOTE:50	**0.7498***	0.7431	0.7435	0.7496	0.7391	0.7315	0.7389	0.7441	0.7321	0.7305	0.7350	0.7364
5-NN	RUS:35	0.7300	0.7452	0.7474	0.7390	0.7359	0.7481	0.7484*	0.7476	0.7296	0.7358	0.7432	0.7414
	RUS:50	0.7230	0.7408	0.7446	0.7537	0.7408	0.7495	**0.7624**	**0.7631***	**0.7351**	**0.7439**	**0.7570**	**0.7486**
	ROS:35	0.7263	0.7363	0.7423*	0.7403	0.7258	0.7306	0.7405	0.7404	0.7234	0.7316	0.7412	0.7377
	ROS:50	0.7145	0.7269	0.7282	0.7356	0.7060	0.7202	0.7223	0.7308	0.7214	0.7274	0.7398*	0.7345
	SMOTE:35	**0.7448**	**0.7511**	0.7557	0.7424	**0.7481**	**0.7569***	0.7535	0.7471	0.7286	0.7307	0.7436	0.7479
	SMOTE:50	0.7319	0.7495	**0.7562**	**0.7598***	0.7333	0.7512	0.7492	0.7529	0.7312	0.7340	0.7489	0.7480
SVM	RUS:35	0.7488	0.7449	0.7317	0.7340	0.7473	0.7411	0.7351	0.7428	**0.7524***	0.7366	0.7374	0.7391
	RUS:50	0.7431	0.7214	0.7297	0.7331	0.7384	0.7315	**0.7456**	**0.7586***	0.7511	**0.7393**	**0.7381**	**0.7456**
	ROS:35	0.7448	0.7371	0.7317	**0.7418**	0.7466	0.7353	0.7300	0.7398	0.7508*	0.7348	0.7224	0.7247
	ROS:50	0.7340	0.7428	**0.7351**	0.7397	0.7377	0.7384	0.7294	0.7395	0.7470*	0.7329	0.7208	0.7249
	SMOTE:35	**0.7498**	**0.7457**	0.7319	0.7388	**0.7528***	**0.7438**	0.7314	0.7385	0.7507	0.7335	0.7229	0.7246
	SMOTE:50	0.7473	0.7364	0.7309	0.7338	0.7524*	0.7332	0.7293	0.7335	0.7484	0.7304	0.7215	0.7249
RF100	RUS:35	0.7598	0.7607	0.7643	0.7679*	0.7543	0.7578	0.7669	0.7675	**0.7588**	0.7622	0.7660	0.7578
	RUS:50	0.7618	0.7659	0.7749	**0.7750***	0.7532	0.7643	0.7670	0.7709	0.7576	**0.7667**	0.7666	0.7671
	ROS:35	0.7550	0.7596	0.7689	0.7719*	0.7537	0.7640	0.7675	0.7709	0.7499	0.7585	0.7693	**0.7715**
	ROS:50	0.7468	0.7659	0.7702	0.7646	0.7564	0.7714	0.7736	**0.7797***	0.7522	0.7574	0.7643	0.7658
	SMOTE:35	0.7631	**0.7707**	**0.7784***	0.7749	0.7608	0.7649	0.7730	0.7694	0.7523	0.7619	**0.7704**	0.7703
	SMOTE:50	**0.7635**	0.7688	0.7764	0.7717	**0.7625**	**0.7767**	**0.7780***	0.7766	0.7522	0.7605	0.7635	0.7680
LR	RUS:35	**0.7194***	**0.6892**	0.6732	0.6654	0.6943	0.6748	0.6789	0.6806	0.6995	**0.6801**	0.6808	0.6839
	RUS:50	0.7046	0.6667	0.6767	0.6627	0.6892	0.6761	**0.6985**	**0.7067***	0.6793	0.6759	**0.6836**	**0.6942**
	ROS:35	0.7120	0.6752	0.6712	0.6621	0.7139	0.6794	0.6465	0.6567	**0.7166***	0.6690	0.6527	0.6368
	ROS:50	0.7100	0.6853	0.6646	**0.6738**	0.7050	0.6788	0.6526	0.6577	0.7146*	0.6627	0.6362	0.6361
	SMOTE:35	0.7083	0.6870	**0.6792**	0.6570	0.7141	**0.6900**	0.6680	0.6469	0.7154*	0.6715	0.6548	0.6356
	SMOTE:50	0.7168	0.6790	0.6713	0.6562	**0.7243***	0.6716	0.6577	0.6399	0.7113	0.6623	0.6412	0.6330

single feature selection technique and each individual value is the average AUC value across all fifteen datasets and four runs of five-fold cross-validation for every combination of data sampling technique, data sampling approach, post-sampling class distribution ratio, feature selection technique, feature subset size, and classifier. Additionally, all of the tables are split into six sections, one for each classifier. In order to improve visibility, in each section, we have put the top performing data sampling technique/post-sampling class distribution ratio combination for each feature subset size in each data sampling approach in **boldface**. Additionally, in each section, the top performing post sampling class ratio for every data sampling technique, data

Table 7.3: Classification Results: ROC

Learner	DS Technique	DS-FS:Unsam 25	DS-FS:Unsam 50	DS-FS:Unsam 100	DS-FS:Unsam 200	DS-FS:Sam 25	DS-FS:Sam 50	DS-FS:Sam 100	DS-FS:Sam 200	FS-DS 25	FS-DS 50	FS-DS 100	FS-DS 200
NB	RUS:35	0.7718	0.7667	0.7634	0.7460	0.7681	0.7630	0.7616	0.7519	0.7728*	0.7682	0.7593	0.7535
	RUS:50	0.7576	0.7675	0.7537	0.7433	0.7650	0.7624	0.7573	0.7482	0.7697*	0.7653	0.7582	0.7548
	ROS:35	0.7748	0.7746	0.7619	0.7557	0.7708	0.7718	0.7622	0.7517	0.7753	0.7773*	0.7616	0.7539
	ROS:50	0.7735	0.7746	0.7643	0.7567	**0.7726**	**0.7743**	**0.7642**	0.7560	**0.7754**	**0.7784***	**0.7623**	**0.7575**
	SMOTE:35	0.7731	0.7703	0.7672	**0.7675**	0.7686	0.7666	0.7621	**0.7600**	0.7750*	0.7747	0.7582	0.7471
	SMOTE:50	**0.7760**	**0.7792***	**0.7685**	0.7622	0.7695	0.7661	0.7606	0.7561	0.7716	0.7723	0.7573	0.7448
MLP	RUS:35	0.7458	0.7362	0.7367	0.7410	**0.7483**	0.7444	0.7487	0.7513	**0.7569***	**0.7415**	0.7427	0.7492
	RUS:50	0.7476	**0.7489**	**0.7419**	**0.7421**	0.7446	**0.7483**	**0.7542**	**0.7611***	0.7514	0.7379	**0.7453**	**0.7538**
	ROS:35	**0.7554***	0.7335	0.7288	0.7398	0.7472	0.7248	0.7232	0.7368	0.7380	0.7235	0.7280	0.7352
	ROS:50	0.7456*	0.7333	0.7337	0.7375	0.7354	0.7209	0.7272	0.7390	0.7348	0.7202	0.7275	0.7379
	SMOTE:35	0.7520*	0.7291	0.7281	0.7353	0.7421	0.7201	0.7245	0.7347	0.7367	0.7249	0.7252	0.7373
	SMOTE:50	0.7467*	0.7334	0.7308	0.7319	0.7328	0.7185	0.7234	0.7322	0.7347	0.7174	0.7266	0.7375
5-NN	RUS:35	0.7434	0.7346	0.7458	0.7440	0.7456	0.7490	0.7549*	0.7494	0.7439	0.7401	0.7456	0.7509
	RUS:50	0.7355	0.7458	0.7472	0.7479	**0.7501**	**0.7644**	**0.7623**	**0.7677***	**0.7485**	**0.7480**	**0.7464**	**0.7541**
	ROS:35	0.7399	0.7392	**0.7481***	0.7467	0.7358	0.7374	0.7450	0.7445	0.7358	0.7382	0.7387	0.7400
	ROS:50	0.7415	0.7394	0.7436	0.7444*	0.7332	0.7335	0.7374	0.7334	0.7346	0.7359	0.7339	0.7319
	SMOTE:35	0.7406	0.7429	0.7459	**0.7514**	0.7463	0.7462	0.7486	0.7523	0.7457	0.7414	0.7435	0.7535*
	SMOTE:50	**0.7444**	**0.7522***	0.7477	0.7476	0.7436	0.7514	0.7470	0.7496	0.7433	0.7393	0.7441	0.7438
SVM	RUS:35	0.7471	0.7322	0.7205	**0.7323**	0.7510	0.7336	0.7305	0.7388	**0.7603***	0.7297	0.7258	0.7332
	RUS:50	0.7450	**0.7381**	**0.7242**	0.7298	0.7422	**0.7385**	**0.7385**	**0.7509**	0.7543*	**0.7302**	**0.7263**	**0.7414**
	ROS:35	**0.7579**	0.7294	0.7139	0.7273	**0.7609***	0.7296	0.7129	0.7264	0.7531	0.7221	0.7112	0.7256
	ROS:50	0.7521	0.7258	0.7126	0.7232	0.7545*	0.7226	0.7100	0.7228	0.7516	0.7175	0.7111	0.7258
	SMOTE:35	0.7503	0.7241	0.7194	0.7260	0.7513*	0.7207	0.7188	0.7244	0.7511	0.7228	0.7116	0.7255
	SMOTE:50	0.7484	0.7242	0.7157	0.7182	0.7462	0.7184	0.7093	0.7159	0.7498*	0.7194	0.7109	0.7258
RF100	RUS:35	0.7629	0.7643	0.7633	**0.7718***	0.7613	0.7605	0.7620	0.7639	0.7589	0.7585	0.7600	0.7631
	RUS:50	0.7665	**0.7724**	**0.7738***	0.7716	0.7557	0.7614	0.7683	**0.7718**	0.7617	**0.7675**	**0.7655**	0.7622
	ROS:35	**0.7676**	0.7704*	0.7689	0.7654	**0.7658**	0.7672	0.7652	0.7671	0.7568	0.7632	0.7637	0.7651
	ROS:50	0.7637	0.7678	0.7686	0.7677	0.7589	0.7646	0.7606	0.7682	**0.7650**	0.7626	0.7643	0.7716*
	SMOTE:35	0.7584	0.7646	0.7733*	0.7632	0.7632	0.7664	0.7648	0.7704	0.7645	0.7643	0.7630	**0.7721**
	SMOTE:50	0.7535	0.7618	0.7664	0.7683	0.7627	**0.7677**	**0.7699***	0.7618	0.7604	0.7608	0.7629	0.7663
LR	RUS:35	0.7143*	0.6738	0.6772	0.6546	0.7013	0.6666	0.6837	0.6757	0.7013	0.6617	0.6701	0.6773
	RUS:50	0.7164*	**0.6865**	**0.6786**	**0.6742**	0.6884	**0.6855**	**0.6923**	**0.6984**	0.6841	**0.6644**	**0.6746**	**0.6937**
	ROS:35	**0.7217***	0.6782	0.6574	0.6524	**0.7192**	0.6633	0.6528	0.6370	0.7092	0.6602	0.6405	0.6417
	ROS:50	0.7081	0.6586	0.6485	0.6496	0.7143*	0.6512	0.6395	0.6364	**0.7109**	0.6531	0.6334	0.6326
	SMOTE:35	0.7129*	0.6580	0.6542	0.6542	0.7082	0.6506	0.6385	0.6376	0.7081	0.6593	0.6467	0.6427
	SMOTE:50	0.6956	0.6535	0.6443	0.6473	0.6979	0.6446	0.6369	0.6383	0.7079*	0.6522	0.6351	0.6362

sampling approach, and feature subset size is underlined. Lastly, the top performing data sampling approach for each feature subset size is marked with an asterisk.

Starting with IG, we see that RUS, with either 35:65 or 50:50 as the post-sampling class distribution ratio, is the most frequent top performing data sampling technique with 31 of the possible 72 (43.06%) data sampling approach, classifier, and feature subset size combinations, followed by SMOTE with 25 (34.72%) combinations and ROS with 16 (22.22%) combinations.We see that DS-FS:UnSam is the most frequent top performing data sampling approach with 14 out of 36 (38.89%) possible combinations of data sampling technique, post-sampling class distribution ratio, and clas-

Table 7.4: Classification Results: S2N

Learner	DS Technique	DS-FS:Unsam				DS-FS:Sam				FS-DS			
		25	50	100	200	25	50	100	200	25	50	100	200
NB	RUS:35	0.7527	0.7411	0.7358	0.7265	0.7542*	0.7483	0.7433	0.7343	0.7440	0.7432	0.7325	0.7255
	RUS:50	0.7580*	0.7505	0.7306	0.7278	**0.7576**	0.7525	0.7442	0.7383	0.7422	0.7436	**0.7412**	**0.7359**
	ROS:35	0.7534*	0.7528	0.7358	0.7259	0.7534	0.7522	0.7364	0.7285	0.7509	**0.7487**	0.7238	0.7203
	ROS:50	0.7533	0.7548*	0.7321	0.7215	0.7532	0.7540	0.7335	0.7221	**0.7515**	0.7474	0.7225	0.7196
	SMOTE:35	0.7564	**0.7600***	**0.7561**	**0.7447**	0.7531	**0.7557**	**0.7532**	**0.7493**	0.7495	0.7465	0.7275	0.7233
	SMOTE:50	**0.7605***	0.7559	0.7428	0.7406	0.7557	0.7450	0.7437	0.7407	0.7509	0.7460	0.7270	0.7207
MLP	RUS:35	**0.7485**	0.7410	0.7281	0.7462	**0.7543**	0.7437	0.7490	0.7562*	**0.7430**	**0.7367**	0.7447	0.7490
	RUS:50	0.7466	**0.7445**	**0.7399**	**0.7477**	0.7467	**0.7477**	**0.7521**	**0.7575***	0.7378	0.7339	**0.7492**	**0.7532**
	ROS:35	0.7457*	0.7242	0.7327	0.7385	0.7376	0.7201	0.7312	0.7345	0.7312	0.7278	0.7276	0.7326
	ROS:50	0.7388	0.7274	0.7284	0.7390*	0.7240	0.7251	0.7239	0.7379	0.7244	0.7196	0.7267	0.7340
	SMOTE:35	0.7369	0.7249	0.7344	0.7392	0.7307	0.7219	0.7325	0.7394*	0.7322	0.7275	0.7275	0.7350
	SMOTE:50	0.7362	0.7265	0.7282	0.7405	0.7238	0.7213	0.7247	0.7413*	0.7231	0.7179	0.7250	0.7341
5-NN	RUS:35	0.7234	0.7280	0.7333	0.7357	0.7321	0.7364	0.7443*	0.7419	0.7276	0.7367	0.7396	0.7421
	RUS:50	**0.7359**	**0.7433**	**0.7490**	**0.7455**	**0.7519**	**0.7534**	**0.7532**	**0.7575***	**0.7366**	**0.7402**	**0.7467**	**0.7456**
	ROS:35	0.7262	0.7272	0.7309	0.7319*	0.7233	0.7244	0.7255	0.7262	0.7145	0.7201	0.7219	0.7290
	ROS:50	0.7237	0.7283*	0.7274	0.7271	0.7162	0.7242	0.7206	0.7186	0.7116	0.7162	0.7197	0.7267
	SMOTE:35	0.7330	0.7344	0.7364	0.7397	0.7351	0.7386	0.7450*	0.7426	0.7178	0.7270	0.7294	0.7414
	SMOTE:50	0.7349	0.7355	0.7397	0.7364	0.7409*	0.7374	0.7374	0.7399	0.7162	0.7271	0.7355	0.7376
SVM	RUS:35	**0.7469**	0.7283	0.7145	0.7308	**0.7494***	**0.7363**	0.7274	0.7427	0.7443	0.7245	0.7259	0.7352
	RUS:50	0.7417	**0.7378**	**0.7242**	**0.7335**	0.7383	0.7307	**0.7375**	**0.7470***	0.7417	**0.7251**	**0.7277**	**0.7431**
	ROS:35	0.7453	0.7220	0.7166	0.7252	0.7441	0.7202	0.7157	0.7236	**0.7457***	0.7203	0.7126	0.7184
	ROS:50	0.7401	0.7238	0.7139	0.7258	0.7409*	0.7199	0.7106	0.7248	0.7405	0.7183	0.7110	0.7180
	SMOTE:35	0.7431*	0.7184	0.7213	0.7283	0.7408	0.7158	0.7202	0.7267	0.7427	0.7218	0.7126	0.7183
	SMOTE:50	0.7393*	0.7180	0.7134	0.7268	0.7380	0.7135	0.7127	0.7256	0.7391	0.7196	0.7117	0.7180
RF100	RUS:35	0.7497	0.7544	0.7594	0.7579	0.7459	0.7468	0.7565	0.7644*	0.7450	0.7497	0.7534	0.7569
	RUS:50	**0.7562**	**0.7582**	**0.7664**	**0.7687**	**0.7533**	**0.7630**	**0.7692**	**0.7697***	**0.7491**	**0.7587**	**0.7585**	0.7549
	ROS:35	0.7487	0.7490	0.7556	0.7579*	0.7461	0.7486	0.7550	0.7549	0.7455	0.7440	0.7523	0.7534
	ROS:50	0.7463	0.7494	0.7581	0.7620*	0.7445	0.7463	0.7535	0.7605	0.7377	0.7458	0.7545	**0.7576**
	SMOTE:35	0.7426	0.7509	0.7613*	0.7582	0.7466	0.7582	0.7567	0.7579	0.7423	0.7497	0.7550	0.7503
	SMOTE:50	0.7461	0.7534	0.7560	0.7630*	0.7483	0.7503	0.7565	0.7579	0.7435	0.7438	0.7520	0.7543
LR	RUS:35	0.7156*	0.6686	0.6543	0.6557	0.6938	0.6776	0.6827	0.6898	0.6873	**0.6707**	0.6677	0.6832
	RUS:50	**0.7180***	**0.6796**	**0.6738**	0.6551	0.6811	**0.6871**	**0.6969**	**0.6946**	0.6700	0.6670	**0.6834**	**0.7035**
	ROS:35	0.7060	0.6632	0.6600	0.6522	**0.7080***	0.6683	0.6518	0.6386	0.7009	0.6642	0.6522	0.6425
	ROS:50	0.6915	0.6561	0.6506	0.6523	0.7055*	0.6609	0.6433	0.6331	**0.7019**	0.6598	0.6481	0.6390
	SMOTE:35	0.6982	0.6597	0.6561	0.6530	0.7013*	0.6622	0.6560	0.6488	0.6996	0.6618	0.6489	0.6402
	SMOTE:50	0.6952	0.6607	0.6542	**0.6615**	0.7027*	0.6613	0.6424	0.6374	0.6987	0.6596	0.6474	0.6405

sifier. DS-FS:UnSam is followed by DS-FS:Sam and FS-DS with 12 (33.33%) and 10 (27.78%) combinations respectively. Lastly, we see that 35:65 is the most frequent top performing post-sampling class distribution ratio with 129 out of 216 (59.72%) combinations of data sampling technique and classifier.

When we look at ROC, we see that RUS, with either 35:65 or 50:50 as the post-sampling class distribution ratio, is the most frequent top performing data sampling technique with 45 of the possible 72 (62.50%) data sampling approach, classifier, and feature subset size combinations, followed by ROS with 17 (23.61%) combinations and SMOTE with 10 (13.89%) combinations. In terms of the most frequent top

performing data sampling approach we see that DS-FS:UnSam wins out with 16 out of 36 (44.44%) possible combinations of data sampling technique, post-sampling class distribution ratio, and classifier. DS-FS:UnSam is followed by FS-DS and DS-FS:Sam with 12 (33.33%) and 8 (22.22%) combinations respectively. Lastly, we see that 35:65 is the most frequent top performing post-sampling class distribution ratio with 119 out of 216 (55.09%) combinations of data sampling technique and classifier.

In terms of S2N, we see that RUS, with either 35:65 or 50:50 as the post-sampling class distribution ratio, is the most frequent top performing data sampling technique with 58 of the possible 72 (80.56%) data sampling approach, classifier, and feature subset size combinations, followed by SMOTE with 8 (11.11%) combinations and ROS with 6 (8.33%) combinations. We see that DS-FS:Sam is the most frequent top performing data sampling approach with 19 out of 36 (44.44%) possible combinations of data sampling technique, post-sampling class distribution ratio, and classifier. DS-FS:Sam is followed by DS-FS:UnSam and FS-DS with 16 (44.44%) and 1 (2.78%) combinations respectively. Lastly, we see that 35:65 is the most frequent top performing post-sampling class distribution ratio with 114 out of 216 (52.78%) combinations of data sampling technique and classifier.

7.5.1 Statistical Analysis

To validate our results and to determine if any of the discovered patterns can be considered statistically significant, we conducted an ANalysis Of VAriance (ANOVA) [14]. Table 7.5 contains the results of three factor ANOVA test with the factors being the choice of data sampling technique (denoted as DS Technique), the choice of data sampling approach (denoted as DS Approach), and the choice of post-sampling class distribution ratio (denoted as Class Ratio). We chose a significance level of 5% for this ANOVA analysis and a "Prob>F" score of less than 0.05 is considered to be statistically significant. The results show that the choice of data sampling technique and

Table 7.5: ANOVA Results

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
DS Approach	1	2	0.51025	10.34	0
DS Technique	4.3	2	2.12724	43.12	0
Class Ratio	0	1	0.04575	0.93	0.3355
Error	19179	388794	0.04933		
Total	19184.4	388799			

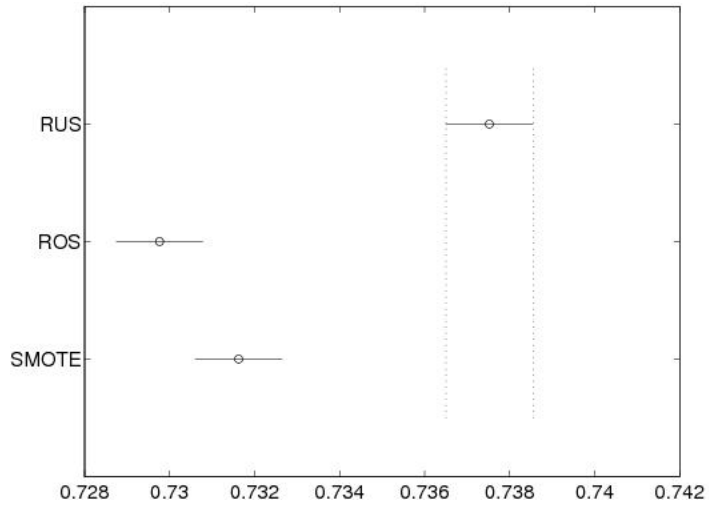
the choice of data sampling approach is significant and the the choice of post-sampling class distribution ratio is not.

In order to determine which of the choices for the data sampling technique and the data sampling approach are significantly better or worse from the others we performed a multiple comparison test with Tukey's Honestly Significant Difference (HSD) [14] test. Figures 7.1 and 7.2 contain the results of the Tukey's HSD tests for the data sampling technique and the data sampling approach respectively. Looking at the data sampling technique, we see that RUS is significantly better than both ROS and SMOTE and that ROS and SMOTE are not significantly different from each other, though SMOTE is better. In terms of the data sampling approach, we see that FS-DS is significantly worse than the other two approaches. DS-FS:UnSam and DS-FS:Sam are not significantly different from each other through, DS-FS:UnSam is slightly better. Thus, it is our recommendation to use RUS using the DS-FS:UnSam approach with either post-sampling class distribution ratio in order to maximize performance when using data sampling to alleviate class imbalance.

7.6 CONCLUSIONS

Class imbalance is a common problem found in bioinformatics datasets. With complications such as bias towards the majority class and potentially reduced performance it is recommended to combat this issue with techniques like data sampling. However, a question that remains is how one should perform the process of data sampling. This

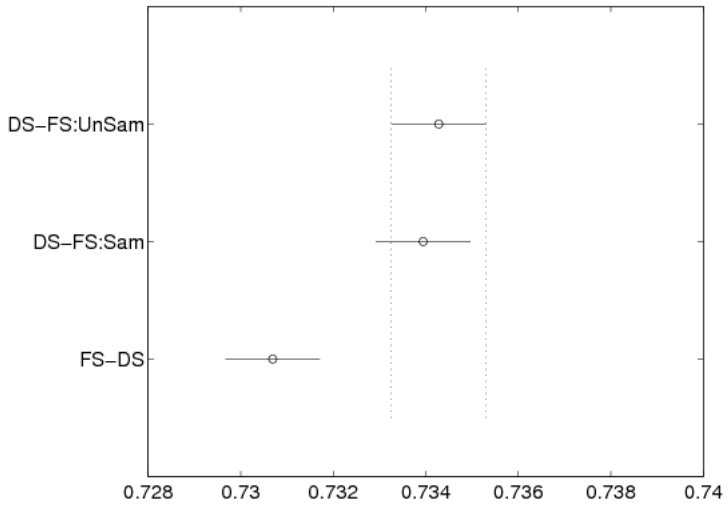
Figure 7.1: Tukey's HSD Results: Data Sampling Process: Technique



work focuses on three aspects of the data sampling process: data sampling technique, data sampling approach, and post-sampling class distribution ratio in terms of classification performance. We use a collection of three data sampling techniques, three data sampling approaches, and two post-sampling class distribution ratios. In order to test these aspects, we use fifteen imbalanced datasets (with the largest minority class percentage being 25.56%), three feature selection techniques, four feature subset sizes, and six classifiers. To our knowledge, this is the most comprehensive study on the process of data sampling within the domain of bioinformatics.

Our results, including statistical analysis with ANOVA and Tukey's HSD tests, show that RUS is clearly the most frequent top performing data sampling technique, followed by SMOTE and ROS, and that RUS is significantly better than the other data sampling techniques. In terms of the data sampling approach, we see that DS-FS:UnSam is the most frequent top performing data sampling approach, followed by DS-FS:Sam and FS-DS and that DS-FS:UnSam and DS-FS:Sam are significantly better than FS-DS, with DS-FS:UnSam being slightly better. Lastly, for post-sampling class distribution ratio, we see that 35:65 is the most frequent top performing post-sampling class distribution ratio, followed by 50:50, but the two post-sampling class

Figure 7.2: Tukey's HSD Results: Data Sampling Process: Approach



distribution ratios are not significantly different. Thus, our recommendation is to use RUS with either the DS-FS:Unsam or DS-FS:Sam approach and a post-sampling class-distribution ratio of 35:65 in order to optimize classification performance when performing data sampling and to minimize data loss by using the less aggressive post-sampling class distribution ratio. Future work in this area will focus on applying these techniques on a set of datasets with a more focused purpose such as only tumor classification datasets or only patient response datasets.

CHAPTER 8
CONCLUSION AND FUTURE WORK

Bioinformatics data is known for being challenging for researchers to work with containing obstacles like high-dimensionality, gene (feature) selection instability, class imbalance, noisy data, and difficult to learn class boundaries, which have to be accounted for. However, the domain of machine learning has a number of techniques designed to tackle these problems and make the analysis process easier for researchers. This research focuses on applying machine learning techniques to alleviate specific problems that are present within bioinformatics datasets. Additionally, this work does not only look at each challenge in isolation but seeks to present approaches towards the appropriate combining of techniques so as to tackle multiple issues at once.

8.1 CONCLUSIONS

Gene selection stability is an important factor to consider when utilizing gene selection techniques and when using predictive models built from existing gene subsets when new data is added or removed. Chapter 3 discusses how various data characteristics can affect gene selection stability. In terms of factors inherent to the data itself, we see that as the difficulty-of-learning increases or the level of balance increases, stability decreases, though this may be due to how the changes are distributed and the distribution of different balance levels across difficulty levels. We also see that as both the number of instances and the number of attributes increases, the stability increases. As for the experimental factors, as the size of the feature subset size

increases, stability increases and as dataset perturbation increases, stability decreases. When we looked closer at datasets difficulty, we see that more unstable ranker will have a more drastic change between the easy and moderate datasets and the more stable feature rankers will have a larger difference between the moderate and hard datasets. Additionally, we found that the larger the feature subset size is the more resilient to difficulty the feature subset is.

Ensembles have been primarily applied toward the learning portion of the machine learning process but ensemble can also be applied toward the feature/gene selection process. In Chapter 4, we presented a thorough study of the ensemble gene selection process, including observations on the ensemble approach, the rank aggregation technique, and the appropriate number of iterations for certain ensemble approaches. Our results show that functional diversity and hybrid diversity are clearly distinct from the commonly used ensemble approach and thus, require further study. We also found that as feature subset size increases similarity between the three approaches increases. In terms of classification, we find that ensembles are best suited for the more difficult datasets and the singular feature selection techniques are comparable on the less difficult-to-learn datasets. We also found that different classifiers are affected by ensemble gene selection differently and to be aware of this factor when designing the experiment. In terms of the rank aggregation technique, we found that most techniques perform similarly and that outside of lowest rank (which was had significantly worst performance than the other techniques) any rank aggregation technique is appropriate but we recommend using a simpler technique, such as mean aggregation, for ease of use. Lastly, in terms of number of iterations, we found that the appropriate number of iteration is 20 and that 10 iterations is insufficient to perform ensemble gene selection.

The process of machine learning can be difficult even for practitioners and researchers in said field. Thus, it is beneficial to make the process easier to implement.

Chapter 5 present a framework that simplifies the process of machine learning by removing certain factor choices by either deciding on an answer ahead of time or making the decision insignificant. Our results including statistical analysis show that Random Forest with 100 trees is the top performer in a majority of scenarios. Additionally, we show that the choice of ranking is of little importance, as long as feature selection occurs to reduce the feature set to a more manageable number.

Ensemble learning combined with feature selection is a powerful tool for analyzing challenging data. Chapter 6 looks at two approaches, Select-Bagging and Select-Boosting, for utilizing both feature selection and ensemble learning. Our results show both ensemble approaches are significantly better than when not using an ensemble. The two ensemble approaches are not significantly different from each other but Select-Bagging is slightly better than Select-Boosting. Thus, we recommend Select-Bagging for ensemble learning with bioinformatics.

Class imbalance is a problem in many bioinformatics datasets. Chapter 7 presents a thorough analysis of the entire data sampling process including, the data sampling technique, the approach of combining data sampling and feature selection, and the post-sampling class distribution ratio. Our results show that random undersampling is the top performing technique, that data sampling should be performed before feature selection, and that the post-sampling class distribution ratio does not matter in terms of classification results but a less aggressive ratio results in less data loss. The combination of these factors will maximize your classification performance and reduce the amount of data loss.

Overall, machine learning is an excellent companion to bioinformatics. With the ability to handle a number of challenges that are commonplace within the data and techniques in place to build inductive models, machine learning allows researchers to efficiently and effectively analyze complex biological datasets. This work gives recommendations on how to apply machine learning techniques both singularly and

in conjunction with other techniques so as to optimize performance.

8.2 FUTURE WORK

Opportunities for future work include:

- All of our datasets in this work only had two classes and multiclass datasets are still available for study.
- Additionally, all of our datasets were specifically focused on gene microarrays and some mass spectrometry. Other datasets types including, patient vitals, hospital admissions, or proteomic data, should be tested to see if the findings presented hold true.
- In terms of feature selection, all of our works utilized univariate feature selection. Thus, the proper application of multivariate feature selection may potentially be of benefit.
- In this work we combined feature selection with data sampling, and feature selection with ensemble learning. The next step is to combine all three so as to cover more challenges within a single framework.

BIBLIOGRAPHY

[1] T. Abeel, T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys. Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics*, 26(3):392–398, February 2010.

[2] G. Abraham, A. Kowalczyk, S. Loi, I. Haviv, and J. Zobel. Prediction of breast cancer prognosis using gene set statistics provides signature stability and biological context. *BMC Bioinformatics*, 11:277–291, 2010.

[3] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Impact of noise and data sampling on stability of feature ranking techniques for biological datasets. In *2012 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 415–422, August 2012.

[4] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse. Evaluation of the importance of data pre-processing order when combining feature selection and data sampling. *IJBIDM*, 7(1/2):116–134, 2012.

[5] A. Al-Shahib, R. Breitling, and D. Gilbert. Feature selection and the class imbalance problem in predicting protein function from sequence. *Applied Bioinformatics*, 4(3):195–203, 2005.

[6] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, 96(12):6745–6750, 1999.

[7] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 276–284, New York, NY, USA, 2001. ACM.

[8] W. Awada, T. M. Khoshgoftaar, D. J. Dittman, and R. Wald. The effect of number of iterations on ensemble gene selection. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 198 –203, dec. 2012.

[9] W. Awada, T. M. Khoshgoftaar, D. J. Dittman, R. Wald, and A. Napolitano. A review of the stability of feature selection techniques for bioinformatics data. In *Information Reuse and Integration (IRI), 2012 IEEE International Conference on*, Aug. 2012.

[10] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions On Neural Networks*, pages 537–550, 1994.

[11] R. Batuwita and V. Palade. A new performance measure for class imbalance learning. application to bioinformatics problems. In *International Conference on Machine Learning and Applications*, pages 545–550, Dec. 2009.

[12] D. G. Beer, S. L. Kardia, C.-C. Huang, T. J. Giordano, A. M. Levin, D. E. Misek, L. Lin, G. Chen, T. G. Gharib, D. G. Thomas, M. L. Lizyness, R. Kuick, S. Hayasaka, J. M. Taylor, M. D. Iannettoni, M. B. Orringer, and S. Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 8:816–824, 2002.

[13] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4):559–583, 2000.

[14] M. L. Berenson, M. Goldstein, and D. Levine. *Intermediate Statistical Methods and Applications: A Computer Package Approach 2nd Edition*. Prentice Hall, 1983.

[15] R. Blagus and L. Lusa. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*, pages 523–539, 2010.

[16] R. Blagus and L. Lusa. Evaluation of smote for high-dimensional class-imbalanced microarray data. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 89–94, 2012.

[17] A.-L. Boulesteix and M. Slawski. Stability and aggregation of ranked gene lists. *Briefings in Bioinformatics*, 10(5):556–568, 2009.

[18] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[19] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[20] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[21] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification And Regression Trees*. Chapman and Hall, 1993.

[22] R. Breitling and P. Herzyk. Rank-based methods as a non-parametric alternative of the t-statistic for the analysis of biological microarray data. *Journal of bioinformatics and computational biology*, 3(5):1171–1189, oct 2005.

[23] T. Chen. A selective ensemble classification method on microarray data. *Journal of Chemical & Pharmaceutical Research*, 6(6):2860–2866, 2014.

[24] W. Chen, H. Lu, M. Wang, and C. Fang. Gene expression data classification using artificial neural network ensembles based on samples filtering. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 1, pages 626–628. IEEE, 2009.

[25] X. Chen and M. Wasikowski. Fast: a roc-based feature selection metric for small samples and imbalanced data classification problems. In *Proceedings of the 14th ACM SIGKDD Int'l Conf. Knowldege Discovery and Data Mining (KDD '08)*, pages 124–132, New York, NY, 2008. ACM.

[26] W. J. Conover. *Practical Nonparametric Studies*. John Wiley and Sons, 2nd edition, 1971.

[27] R. Diaz-Uriarte and S. Alvarez de Andres. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:1–13, 2006.

[28] T. Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Computer Science*, pages 1–15. Springer Berlin Heidelberg, 2000.

[29] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano. Selecting the appropriate data sampling approach for imbalanced and high-dimensional bioinformatics datasets. In *Bioinformatics and Bioengineering (BIBE), 2014 14th IEEE International Conference on*, pages 304–310, 2014.

[30] D. J. Dittman, T. M. Khoshgoftaar, and A. Napolitano. A comprehensive study of the data sampling process on imbalanced bioinformatics data. Technical report, February 2015 In Submission.

[31] D. J. Dittman, T. M. Khoshgoftaar, A. Napolitano, and A. Fazelpour. Selectbagging: Effectively combining gene selection and bagging for balanced bioinformatics data. In *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*, pages 413–419, Nov 2014.

[32] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Hulse. Feature selection algorithms for mining high dimensional dna microarray data. In *Handbook of Data Intensive Computing*, pages 685–710. Springer New York, 2011.

[33] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Random forest: A reliable tool for patient response prediction. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM) Workshops*, pages 289–296. BIBM, 2011.

[34] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Comparing two new gene selection ensemble approaches with the commonly-used approach. In *Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA): Health Informatics Workshop*, pages 184–191. ICMLA, 2012.

[35] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Determining the number of iterations appropriate for ensemble gene selection on microarray data. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, pages 82 –89, dec. 2012.

[36] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Similarity analysis of feature ranking techniques on imbalanced dna microarray datasets. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 398–402. BIBM, 2012.

[37] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Classification performance of rank aggregation techniques for ensemble gene selection. In *Florida Artificial Intelligence Society (FLAIRS), 26th International Conference*, pages 420–425, 2013.

[38] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Comparison of rank-based vs. score-based aggregation for ensemble gene selection. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 225–231, Aug 2013.

[39] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Gene selection stability's dependence on dataset difficulty. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 341–348, Aug 2013.

[40] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Maximizing classification performance for patient response datasets. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 454–462, Nov 2013.

[41] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Simplifying the utilization of machine learning techniques for bioinformatics. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 2, pages 396–403, Dec 2013.

[42] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Comparison of data sampling approaches for imbalanced bioinformatics data. In *Florida Artificial Intelligence Society (FLAIRS), 27th International Conference*, pages 268–271, 2014.

[43] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Selecting the appropriate ensemble learning approach for balanced bioinformatics data. In *Florida Artificial Intelligence Society (FLAIRS), 28th International Conference*, page In Press, 2015.

[44] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. A comprehensive study of the effects of data characteristics on the stability of feature selection in bioinformatics. Technical report, September 2015 In Submission.

[45] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse. Comparative analysis of dna microarray data through the use of feature selection techniques. In *Proceedings of the Ninth IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 147–152. ICMLA, 2010.

[46] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and H. Wang. Stability analysis of feature ranking techniques on biological datasets. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 252–256. BIBM, 2011.

[47] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[48] X. Fan, L. Shi, H. Fang, Y. Cheng, R. Perkins, and W. Tong. Dna microarrays are predictive of cancer prognosis: A re-evaluation. *Clinical Cancer Research*, 16:629–636, 2010.

[49] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861 – 874, 2006.

[50] U. M. Fayyad and K. B. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.

[51] G. Forman. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, pages 3:1289–1305, 2003.

[52] Y. Freund, R. Schapire, et al. Experiments with a new boosting algorithm. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 148–156. MORGAN KAUFMANN PUBLISHERS, INC., 1996.

[53] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 148–156, 1996.

[54] K. Gao, T. M. Khoshgoftaar, and A. Napolitano. Impact of data sampling on stability of feature selection for software measurement data. In *Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on*, pages 1004–1011, Nov 2011.

[55] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[56] Q. Gu, Z. Li, and J. Han. Generalized fisher score for feature selection. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 266–273. AUAI Press, July 2011.

[57] G. Guile and W. Wang. Factors affecting boosting ensemble performance on dna microarray data. In *Neural Networks (IJCNN), The 2010 International Joint Conference on*, pages 1 –7, july 2010.

[58] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, Nov. 2009.

[59] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(6):392–398, November/December 2003.

[60] M. A. Hall and L. A. Smith. Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. In *Proceedings of the Twelfth International Florida Artificial Intelligence Research Society Conference*, pages 235–239, May 1999.

[61] C. Hatzis, L. Pusztai, V. Valero, and et al. A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer. *JAMA*, 305(18):1873–1881, 2011.

[62] A.-C. Haury, P. Gestraud, and J.-P. Vert. The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PLoS ONE*, 6(12):e28210, 12 2011.

[63] S. Haykin. *Neural Networks: A Comprehensive Foundation 2nd edition*. Prentice Hall, 1998.

[64] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, Sept. 2009.

[65] I. Inza, P. Larraaga, R. Blanco, and A. J. Cerrolaza. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 31(2):91–103, 2004.

[66] I. Jeffery, D. Higgins, and A. Culhane. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics*, 7(1), 2006.

[67] A. Kalousis, J. Prados, and M. Hilario. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems*, 12(1):95–116, Dec. 2006.

[68] A. Kamal, X. Zhu, A. Pandya, S. Hsu, and M. Shoaib. The impact of gene selection on imbalanced microarray expression data. In S. Rajasekaran, editor, *Bioinformatics and Computational Biology*, volume 5462 of *Lecture Notes in Computer Science*, pages 259–269. 2009.

[69] T. M. Khoshgoftaar, D. J. Dittman, R. Wald, and W. Awada. A review of ensemble classification for dna microarrays data. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 381–389, Nov 2013.

[70] T. M. Khoshgoftaar, D. J. Dittman, R. Wald, and A. Fazelpour. First order statistics based feature selection: A diverse and powerful family of feature selection techniques. In *Proceedings of the Eleventh International Conference on Machine Learning and Applications (ICMLA): Health Informatics Workshop*, pages 151–157. ICMLA, 2012.

[71] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse. An empirical study of learning from imbalanced data using random forest. In *19th IEEE International Conference on Tools with Artificial Intelligence, 2007. ICTAI 2007.*, volume 2, pages 310–317, October 2007.

[72] T. M. Khoshgoftaar, C. Seiffert, J. Van Hulse, A. Napolitano, and A. Folleco. Learning with limited minority class data. In *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pages 348–353, Dec 2007.

[73] T. M. Khoshgoftaar, R. Wald, D. J. Dittman, and A. Napolitano. Feature list aggregation approaches for ensemble gene selection on patient response datasets. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 317–324, Aug 2013.

[74] T. M. Khoshgoftaar, R. Wald, D. J. Dittman, and A. Napolitano. Classification performance of three approaches for combining data sampling and gene selection on bioinformatics data. In *Information Reuse and Integration (IRI), 2014 14th IEEE International Conference on*, pages 315–321, 2014.

[75] K. Kira and L. A. Rendell. The feature selection problem: Traditional methods and a new solution. In *AAAI '92: Proc. 10th Nat'l Conf. on Artificial Intelligence*, number 10, pages 129–134. John Wiley & Sons, Ltd., July 1992.

[76] R. Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *IJCAI*, volume 14, pages 1137–1145, 1995.

[77] R. Kolde, S. Laur, P. Adler, and J. Vilo. Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580, 2012.

[78] I. Kononenko. Estimating attributes: Analysis and extensions of relief. *Lecture Notes in Computer Science*, pages 171–182, 1994.

[79] J. A. Koziol, A. C. Feng, Z. Jia, Y. Wang, S. Goodison, M. McClelland, and D. Mercola. The wisdom of the commons: ensemble tree classifiers for prostate cancer prognosis. *Bioinformatics*, 25(1):54–60, 2009.

[80] L. I. Kuncheva. A stability index for feature selection. In *Proceedings of the 25th IASTED International Multi-Conference: Artificial Intelligence and Applications*, pages 390–395, Anaheim, CA, USA, 2007. ACTA Press.

[81] S. Le Cessie and J. C. V. Houwelingen. Ridge estimators in logistic regression. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pages 191–201, 1992.

[82] H. Liu, J. Li, and L. Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13:51–60, 2002.

[83] S. Loscalzo, L. Yu, and C. Ding. Consensus group stable feature selection. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 567–576, New York, NY, USA, 2009.

[84] J. L. Lustgarten, V. Gopalakrishnan, and S. Visweswaran. Measuring stability of feature selection in biomedical datasets. In *AMIA 2009 Symposium Proceedings*, pages 406–410, 2009.

[85] L. D. Miller, J. Smeds, J. George, V. B. Vega, L. Vergara, A. Ploner, Y. Pawitan, P. Hall, S. Klaar, E. T. Liu, and J. Bergh. An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proceedings of the National Academy of Sciences of the United States of America*, 102(38):13550–13555, 2005.

[86] G. Mulligan, C. Mitsiades, B. Bryant, F. Zhan, W. J. Chng, S. Roels, E. Koenig, A. Fergus, Y. Huang, P. Richardson, W. L. Trepicchio, A. Broyl, P. Sonneveld, J. Shaughnessy, John D., P. Leif Bergsagel, D. Schenkein, D.-L. Esseltine, A. Boral, and K. C. Anderson. Gene expression profiling and correlation with outcome in clinical trials of the proteasome inhibitor bortezomib. *Blood*, pages 3177–3188, 2007.

[87] S. Nagi and D. K. Bhattacharyya. Classification of microarray cancer data using ensemble approach. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 2(3):159–173, 2013.

[88] National Center for Biotechnology Information. Microarrays factsheet, 2007. `http://www.ncbi.nlm.nih.gov/About/primer/microarrays.htm`.

[89] R. Neumayer, R. Mayer, and K. Nrvg. Combination of feature selection methods for text categorisation. In *Advances in Information Retrieval*, volume 6611 of *Lecture Notes in Computer Science*, pages 763–766. 2011.

[90] Y. Pawitan, J. Bjohle, L. Amler, A.-L. Borg, S. Egyhazi, P. Hall, X. Han, L. Holmberg, F. Huang, S. Klaar, E. Liu, L. Miller, H. Nordgren, A. Ploner, K. Sandelin, P. Shaw, J. Smeds, L. Skoog, S. Wedren, and J. Bergh. Gene

expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Research*, 7(6):R953–R964, 2005.

[91] Y. Peng. A novel ensemble machine learning for robust microarray data classification. *Computers in Biology and Medicine*, 36(6):553–573, 2006.

[92] J. R. Quinlan. *C4.5: Programs For Machine Learning*. Morgan Kaufmann, San Mateo, California, 1993.

[93] S. Ramaswamy, K. N. Ross, E. S. Lander, and T. R. Golub. A molecular signature of metastasis in primary solid tumors. *Nature Genetics*, 33:49–54, 2003.

[94] M. Raponi, J.-L. Harousseau, J. E. Lancet, B. Lwenberg, R. Stone, Y. Zhang, W. Rackoff, Y. Wang, and D. Atkins. Identification of molecular predictors of response in a study of tipifarnib treatment in relapsed and refractory acute myelogenous leukemia. *Clinical Cancer Research*, 13(7):2254–2260, 2007.

[95] Y. Saeys, T. Abeel, and Y. Peer. Robust feature selection using ensemble feature selection techniques. In *ECML PKDD '08: Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, pages 313–325, Berlin, Heidelberg, 2008. Springer-Verlag.

[96] Y. Saeys, I. Inza, and P. Larraaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 2007.

[97] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse. A study on the relationships of classifier performance metrics. In *Proceedings of the 21st IEEE International Conference on Tools with Artifical Intelligence (ICTAI'09)*, pages 59–66, Newark, NJ, November 2009. IEEE Computer Society.

[98] C. Sotiriou, S.-Y. Neo, L. M. McShane, E. L. Korn, P. M. Long, A. Jazaeri, P. Martiat, S. B. Fox, A. L. Harris, and E. T. Liu. Breast cancer classification and prognosis based on gene expression profiles from a population-based study. *Proceedings of the National Academy of Sciences*, 100(18):10393–10398, 2003.

[99] A. Spira, J. E. Beane, V. Shah, K. Steiling, G. Liu, F. S. Schembri, Gilman, Y.-M. Dumas, P. Calner, P. Sebastiani, S. Sridhar, J. Beamis, C. Lamb, T. Anderson, N. Gerry, J. Keane, M. E. Lenburg, and J. S. Brody. Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nature Medicine*, 13:361–366, 2007.

[100] G. Stiglic and P. Kokol. Stability of ranked gene lists in large microarray analysis studies. *Journal of biomedicine biotechnology*, 2010:616358.

[101] A. Tabchy, V. Valero, T. Vidaurre, A. Lluch, H. Gomez, M. Martin, Y. Qi, L. J. Barajas-Figueroa, E. Souchon, C. Coutant, F. D. Doimi, N. K. Ibrahim,

Y. Gong, G. N. Hortobagyi, K. R. Hess, W. F. Symmans, and L. Pusztai. Evaluation of a 30-gene paclitaxel, fluorouracil, doxorubicin, and cyclophosphamide chemotherapy response predictor in a multicenter randomized trial in breast cancer. *Clinical Cancer Research*, 16(21):5351–5361, 2010.

[102] A. C. Tan and D. Gilbert. Ensemble machine learning on gene expression data for cancer classification. 2003.

[103] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences*, 98(9):5116–5121, 2001.

[104] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, ICML '07, pages 935–942, New York, NY, USA, 2007. ACM.

[105] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald. A comparative evaluation of feature ranking methods for high dimensional bioinformatics data. In *Proceedings of the IEEE International Conference on Information Reuse and Integration - IRI'11*, pages 315–320, 2011.

[106] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald. Feature selection with high dimensional imbalanced data. In *Proceedings of the 9th IEEE International Conference on Data Mining - Workshops (ICDM'09)*, pages 507–514, Miami, FL, December 2009. IEEE Computer Society.

[107] R. Wald, T. M. Khoshgoftaar, and D. J. Dittman. Mean aggregation versus robust rank aggregation for ensemble gene selection. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, pages 63–69, Dec 2012.

[108] R. Wald, T. M. Khoshgoftaar, and D. J. Dittman. A new fixed-overlap partitioning algorithm for determining stability of bioinformatics gene rankers. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 2, pages 170–177, 2012.

[109] R. Wald, T. M. Khoshgoftaar, and D. J. Dittman. Ensemble gene selection versus single gene selection: Which is better? In *Florida Artificial Intelligence Society (FLAIRS), 26th International Conference*, pages 350–355, 2013.

[110] R. Wald, T. M. Khoshgoftaar, D. J. Dittman, W. Awada, and A. Napolitano. An extensive comparison of feature ranking aggregation techniques in bioinformatics. In *2012 IEEE 13th International Conference on Information Reuse and Integration*, pages 377–384, Aug. 2012.

[111] R. Wald, T. M. Khoshgoftaar, D. J. Dittman, and A. Napolitano. Random forest with 200 selected features: An optimal model for bioinformatics research.

In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, pages 154–160, Dec 2013.

[112] R. Wald, T. M. Khoshgoftaar, A. Fazelpour, and D. J. Dittman. Hidden dependencies between class imbalance and difficulty of learning for bioinformatics datasets. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 232–238. IEEE, 2013.

[113] H. Wang, T. M. Khoshgoftaar, and A. Napolitano. An empirical study of software metrics selection using support vector machine. In *Proceedings of International Conference on Software Engineering and Knowledge Engineering SEKE'11*, pages 83–88, July 7-9, 2011.

[114] M. Wasikowski and X. wen Chen. Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering*, 22:1388–1400, 2010.

[115] T. Watanabe, Y. Komuro, T. Kiyomatsu, T. Kanazawa, Y. Kazama, J. Tanaka, T. Tanaka, Y. Yamamoto, M. Shirane, T. Muto, and H. Nagawa. Prediction of sensitivity of rectal cancer cells in response to preoperative radiotherapy by dna microarray analysis of gene expression profiles. *Cancer Research*, 66(7):3370–3374, 2006.

[116] G. M. Weiss and F. J. Provost. Learning when training data are costly: the effect of class distribution on tree induction. *J. Artif. Intell. Res.(JAIR)*, 19:315–354, 2003.

[117] D. A. Wigle, I. Jurisica, N. Radulovich, M. Pintilie, J. Rossant, N. Liu, C. Lu, J. Woodgett, I. Seiden, M. Johnston, S. Keshavjee, G. Darling, T. Winton, and B.-J. Breitkreutz. Molecular profiling of non-small cell lung cancer and correlation with disease-free survival1. *Cancer Research*, pages 3005–3008, 2002.

[118] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[119] E. P. Xing, M. I. Jordan, and R. M. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608. Morgan Kaufmann Publishers Inc., 2001.

[120] J. Yang and S. Olafsson. Optimization-based feature selection with adaptive instance sampling. *Computers & Operations Research*, 33(11):3088–3106, 2006.

[121] P. Yang, Y. Hwa Yang, B. B Zhou, and A. Y Zomaya. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5(4):296–308, 2010.

[122] T. yu Liu. Easyensemble and feature selection for imbalance data sets. In *Bioinformatics, Systems Biology and Intelligent Computing, 2009. IJCBS '09. International Joint Conference on*, pages 517 –520, aug. 2009.

[123] Z. Zhang, J. Li, H. Hu, and H. Zhou. A robust ensemble classification method analysis. In H. R. Arabnia, editor, *Advances in Computational Biology*, volume 680 of *Advances in Experimental Medicine and Biology*, pages 149–155. Springer New York, 2010.