# ANALYSIS OF MACHINE LEARNING ALGORITHMS ON BIOINFORMATICS DATA OF VARYING QUALITY

by

Ahmad Abu Shanab

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

**ANALYSIS OF MACHINE LEARNING ALGORITHMS ON BIOINFORMATICS DATA OF VARYING QUALITY**

by

Ahmad Abu Shanab

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.

Ionut Cardei, Ph.D.

Hanqi Zhuang, Ph.D.

Bassem Alhalabi, 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

3/23/15
Date

## ACKNOWLEDGEMENTS

I would like to acknowledge my advisor, Dr. Taghi M. Khoshgoftaar, for his support and invaluable guidance of my research and graduate studies at Florida Atlantic University. I also thank Dr. Bassem Alhalabi, Dr. Martin K. Solomon, Dr. Hanqi Zhuang, and Dr. Ionut Cardei for serving on my dissertation committee.

I would also like to thank the members of the Empirical Software Engineering and Data Mining and Machine Learning Laboratories at Florida Atlantic University. It has been a great pleasure working with such an exceptional team

Finally, I am most grateful to my family especially my parents for their support and encouragement during my graduate studies and all other endeavors.

## ABSTRACT

| | |
|---|---|
| Author: | Ahmad Abu Shanab |
| Title: | Analysis of Machine Learning Algorithms on Bioinformatics Data of Varying Quality |
| Institution: | Florida Atlantic University |
| Dissertation Advisor: | Dr. Taghi M. Khoshgoftaar |
| Degree: | Doctor of Philosophy |
| Year: | 2015 |

One of the main applications of machine learning in bioinformatics is the construction of classification models which can accurately classify new instances using information gained from previous instances. With the help of machine learning algorithms (such as supervised classification and gene selection) new meaningful knowledge can be extracted from bioinformatics datasets that can help in disease diagnosis and prognosis as well as in prescribing the right treatment for a disease. One particular challenge encountered when analyzing bioinformatics datasets is data noise, which refers to incorrect or missing values in datasets. Noise can be introduced as a result of experimental errors (e.g. faulty microarray chips, insufficient resolution, image corruption, and incorrect laboratory procedures), as well as other errors (errors during data processing, transfer, and/or mining). A special type of data noise called class noise, which occurs when an instance/example is mislabeled. Previous research showed that class noise has a detrimental impact on machine learning algorithms (e.g. worsened classification performance and unstable feature selection). In addition to data noise, gene expression datasets can suffer from the problems of high dimensionality (a very large feature space) and class imbalance (unequal distribution of instances between classes). As a result of these inherent problems, constructing accurate classification models becomes more challenging.

To provide guidance to researchers and practitioners in deciding which machine learning algorithms to apply for their analysis, this dissertation performs thorough empirical investigations of machine learning algorithms on bioinformatics data of varying data quality. Comprehensive experiments are performed to assess the robustness of machine learning techniques to class noise. First, we provide a detailed experimental analysis of feature selection techniques as well as classification algorithms in the context of data quality. We then investigate the effectiveness of three forms of ensemble classification techniques when learning from balanced bioinformatics datasets in the context of data quality. We also investigate the importance of alleviating class imbalance for classification problems on bioinformatics datasets. Finally, we address the combined problem of high dimensionality and class imbalance in the context of data quality.

*To my father, Dr. Abdul Latif Abu Shanab, my mother, Mrs. Khitam Yasin, my charming and selfless sister Enas, and my siblings, Rula, Ehab, Saji, and Anas.*

**ANALYSIS OF MACHINE LEARNING ALGORITHMS ON BIOINFORMATICS DATA OF VARYING QUALITY**


**List of Tables** . . . . . . . . . . . . . . . . . . . . . . . . . . . . xii

**List of Figures** . . . . . . . . . . . . . . . . . . . . . . . . . . . . xiv

**1 Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.1.1 Data Noise . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.1.2 High Dimensionality . . . . . . . . . . . . . . . . . . . . . 4

1.1.3 Class Imbalance . . . . . . . . . . . . . . . . . . . . . . . 5

1.1.4 The Combined Problem of High Dimensionality and Class Imbalance . . . . . . . . . . . . . . . . . . . . . . . . . . . 6

1.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . 7

1.3 Dissertation Structure . . . . . . . . . . . . . . . . . . . . . . 8

**2 Methodology** . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.1 Quality of Data . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.2 Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.3 Noise Injection . . . . . . . . . . . . . . . . . . . . . . . . . . 13

2.4 Feature Selection . . . . . . . . . . . . . . . . . . . . . . . . . 13

2.4.1 Filter-Based Feature Ranking . . . . . . . . . . . . . . . . 15

2.4.2 Filter-Based Subset Evaluation . . . . . . . . . . . . . . . 20

2.4.3 Wrapper-Based Subset Selection . . . . . . . . . . . . . . . 21

2.5 Sampling Techniques . . . . . . . . . . . . . . . . . . . . . . . 22

2.6 Approaches for Combining Feature Selection and Data Sampling . . . 23
2.7 Classifiers . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
2.8 Performance Evaluation . . . . . . . . . . . . . . . . . . . . . . . . 26

**3 How Ranker and Learner Choice Affects Classification Performance on Noisy Bioinformatics Data** . . . . . . . . . . . . . 28
3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28
3.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
3.4 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.5 Results and Analysis . . . . . . . . . . . . . . . . . . . . . . . . . 33
3.6 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

**4 Evaluation of Subset-Based Feature Selection Using Biological Data with Varying Levels of Data Quality** . . . . . . . . . . . . . . . . . 37
4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 37
4.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38
4.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39
4.4 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.5 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.6 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

**5 Comparing Feature Ranking, Filter-Based Feature Subset Selection, and Wrapper-Based Feature Subset Selection for Classification of Noisy Bioinformatics Data** . . . . . . . . . . . . . 46
5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
5.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
5.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47
5.4 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
5.5 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . . 52
5.6 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

**6 Ensemble Classification Performance on Balanced Bioinformatics Data with Varying Levels of Data Quality** . . . 64

6.1 Introduction . . . 64

6.2 Contributions . . . 66

6.3 Related Work . . . 66

6.4 Ensemble Classification Approaches . . . 69

6.5 Methodology . . . 72

6.6 Experimental Results . . . 73

6.7 Chapter Summary . . . 78

**7 The Importance of Alleviating Class Imbalance for Classification Problems on Bioinformatics Data** . . . 80

7.1 Introduction . . . 80

7.2 Contributions . . . 81

7.3 Related Work . . . 82

7.4 Methodology . . . 84

7.5 Empirical Results . . . 85

7.6 Chapter Summary . . . 89

**8 Comparison of Three Approaches for Combining Feature Selection and Data Sampling Using Bioinformatics Data Varying Levels of Data Quality** . . . 91

8.1 Introduction . . . 91

8.2 Contributions . . . 92

8.3 Related Work . . . 93

8.4 Methodology . . . 97

8.5 Experimental Results . . . 99

8.5.1 Robustness of Classification Algorithms and Rankers to Class Noise . . . 102

8.6 Chapter Summary . . . 107

**9 How to Optimally Combine Univariate and Multivariate Feature Selection with Data Sampling for Classifying Noisy, High Dimensional, Class Imbalanced DNA Microarray Data** . . . . . . 110

9.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110

9.2 Contributions . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110

9.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111

9.4 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114

9.5 Experimental Results . . . . . . . . . . . . . . . . . . . . . . . . 115

9.5.1 Statistical Tests . . . . . . . . . . . . . . . . . . . . . . . 118

9.6 Chapter Summary . . . . . . . . . . . . . . . . . . . . . . . . . . 120

**10 Conclusion and Future Work** . . . . . . . . . . . . . . . . . . . . . . 123

10.1 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 124

10.2 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129

**Bibliography** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130

## LIST OF TABLES


2.1 Dataset Characteristics . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

3.1 Details of the Datasets . . . . . . . . . . . . . . . . . . . . . . . . . . . 32

3.2 Average AUC for High-Quality datasets . . . . . . . . . . . . . . . . . 33

3.3 Average AUC for Average-Quality datasets . . . . . . . . . . . . . . . 34

3.4 Average AUC for Low-Quality datasets . . . . . . . . . . . . . . . . . 34

4.1 Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality) . . . . . . . . . . . . . . . . . . . . . 44

5.1 Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality) . . . . . . . . . . . . . . . . . . . . . 53

5.2 ANOVA Results: Feature Selection Techniques Across All Learners . 54

5.3 ANOVA Results: Feature Selection Techniques For Each Classifier (High-Quality) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

5.4 ANOVA Results: Feature Selection Techniques For Each Classifier (Average-Quality) . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

5.5 ANOVA Results: Feature Selection Techniques For Each Classifier (Low-Quality) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

6.1 Average AUC Values for High-Quality Datasets . . . . . . . . . . . . 73

6.2 Average AUC Values for Average-Quality Datasets . . . . . . . . . . 73

6.3 Average AUC Values for Low-Quality Datasets . . . . . . . . . . . . . 74

6.4 Average AUC Values for All Datasets . . . . . . . . . . . . . . . . . . 75

6.5 ANOVA Results: Ensemble Approaches . . . . . . . . . . . . . . . . . 77

7.1 Average AUC Values . . . . . . . . . . . . . . . . . . . . . . . . . . . 87

7.2 z-test Results . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89

8.1 Average AUC Values For The Approaches to Combining Feature Selection and Data Sampling . . . . . . . . . . . . . . . . . . . . . . . 101

8.2 Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality): NB, MLP, and 5NN . . . . . . . . . . . 103

8.3 Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality): SVM, RF100, and LR . . . . . . . . . 104

8.4 Average AUC Values: Learner, Ranker, and Subset Size . . . . . . . . 106

9.1 Average AUC Values . . . . . . . . . . . . . . . . . . . . . . . . . . . 116

9.2 ANOVA Results: Feature-Selection/Data-Sampling Strategies Across All Learners . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120

## LIST OF FIGURES


2.1 Feature Selection and Data Sampling Approaches . . . . . . . . . . . 24

5.1 Tukey's HSD Results: Feature Selection Techniques Across All Learners . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58

5.2 Tukey's HSD Results: Feature Selection Techniques For Each Classifier (High-Quality) . . . . . . . . . . . . . . . . . . . . . . . . . . 61

5.3 Tukey's HSD Results: Feature Selection Techniques For Each Classifier (Average-Quality) . . . . . . . . . . . . . . . . . . . . . . . . 62

5.4 Tukey's HSD Results: Feature Selection Techniques For Each Classifier (Low-Quality) . . . . . . . . . . . . . . . . . . . . . . . . . . 63

6.1 Select-Bagging . . . . . . . . . . . . . . . . . . . . . . . . . . . 70

6.2 Select-Boosting . . . . . . . . . . . . . . . . . . . . . . . . . . . 71

6.3 Tukey's HSD Results: Ensemble Approaches . . . . . . . . . . . . . . 79

7.1 Evaluation Approaches . . . . . . . . . . . . . . . . . . . . . . . 85

8.1 Tukey's HSD Results: Low-Quality . . . . . . . . . . . . . . . . . . 101

8.2 Tukey's HSD Results: Average-Quality . . . . . . . . . . . . . . . . 102

8.3 Tukey's HSD Results: High-Quality . . . . . . . . . . . . . . . . . . 102

8.4 Tukey's HSD Results: Learners . . . . . . . . . . . . . . . . . . . 107

8.5 Tukey's HSD Results: Rankers . . . . . . . . . . . . . . . . . . . . 108

8.6 Tukey's HSD Results: Subset Sizes . . . . . . . . . . . . . . . . . . 108

9.1 Tukey's HSD Results: Feature-Selection/Data-Sampling Strategies Across All Learners . . . . . . . . . . . . . . . . . . . . . . . . . . . . 121

# CHAPTER 1

# INTRODUCTION

The emergence of DNA microarray chips is a major advancement in biological research. Scientists use DNA microarrays to measure the expression levels of a large number of genes simultaneously. The data from microarray experiments (i.e. gene expression data) are usually organized in a matrix form of gene expressions (rows) and samples (columns). The large size of gene expression datasets makes manual human analysis infeasible, therefore practitioners apply data mining techniques to analyze the gene expression data. With the help of data mining techniques, they will be able to distinguish between healthy and diseased tissue [17, 47], discover biomarkers [1], distinguish between different types of cancer or subtypes of the same cancer [22, 54], and predict patient response to a drug treatment [43, 83]. Data mining techniques can generally be grouped into two broad categories: unsupervised learning and supervised learning. Unsupervised learning techniques seek to find potentially interesting, important, and hidden relationships between any attributes or sets of attributes using unlabeled data; these can look for any rules that strongly associate different attribute values, or group the data instances into a selected number of groups (i.e. clusters) in such a way that instances in the same cluster are similar to each other based on some measure of similarity; these are called association rules and clustering, respectively. Supervised learning techniques build prediction models using labeled data to predict either a categorical value or a numerical value; these are called classification models and regression models, respectively. In this work, we are mainly interested in the problem of classification in bioinformatics.

Data is the cornerstone of any knowledge discovery endeavor using data mining techniques. The quality of the knowledge/results obtained will rely heavily on the quality of data being analyzed. High quality data is defined as data that satisfy the requirements of the intended use. Data quality consists of many factors, including accuracy, completeness, consistency, timeliness, believability, and interpretability. Accuracy is the degree of closeness of measurements of a quantity to that quantity's actual/true value. Completeness means that all data fields necessary for an observation unit are captured. Consistency is a very general term which demands that the data must meet all validation rules. Timeliness indicates the quality of being available on time and would measure the same time period. Believability represents the data source trustworthiness. Interpretability indicates ease of understanding and concise representation of data.

## 1.1 MOTIVATION

Noise is one of the major data quality challenges when analyzing real-world bioinformatics datasets, which refers to incorrect or missing values in a dataset. This problem illustrates three of the elements defining data quality: accuracy, completeness, and consistency. Several studies have noted the suboptimal performance (e.g., weak performance of classification models, low stability of feature selection techniques, and extended time of analysis) as a result of low quality data. For this reason it is necessary to handle low quality data before further analysis. Two additional challenges exhibited by many real-world datasets are high-dimensionality and class imbalance. The overabundance of attributes is commonly known as high dimensionality, while class imbalance refers to the unequal distribution of instances between classes. High dimensionality adds more challenges to data mining, resulting in suboptimal predictive accuracy of classifiers and large computation time of analysis, because not all features make the same contributions to the class. On the other hand, class imbalance not only can harm the classification performance (a classifier may have a very high rate of false negatives due to a bias towards the majority class) but can also influence the stability performance of some feature selection techniques. Most gene expression datasets are characterized by having the three aforementioned challenges simultaneously. Thus, constructing accurate classification models becomes more challenging.

### 1.1.1 Data Noise

The presence of noise is inevitable in gene expression datasets. Noise is random error in a measured variable, which can be divided into two types: attribute noise and class noise. Attribute noise occurs when values in the independent attributes are incorrect (for example, gene expression levels not recorded correctly), while class noise refers to incorrect values in the dependent attribute (for example, cancerous instances labeled as noncancerous). Zhu and Wu [117] examined these two types of noise and concluded that class noise has a more harmful effect on classification performance than attribute noise. Previous research found that class noise can also cause a feature ranker to produce unstable output [6].

Several studies investigated the impact of noise injection (as well as other data changes such as sampling) on the stability of feature selection techniques [3, 109, 110, 111, 19]. The stability of a feature selection technique is normally defined as the sensitivity of a technique to perturbations (changes) in the input data. Although stability is an important evaluation criterion for feature ranking techniques, a feature ranker's stability is not an indicator of its performance in classification [2].

In classification problems, where the main concern is to have an accurate inductive model, many efforts have been spent on noise tolerant inductive learners Some use tree pruning [91], while others use several boosting methods [66, 84].

While the problem of noise in bioinformatics datasets is prevalent, only a limited number of studies in this domain have attempted to quantify noise and understand how it can impact supervised learning. Pechenizkiy et al. [85] analyzed the impact of class noise on supervised learning using datasets from the medical domain. Dietterich [40] explored the effect of noise on the classification performance of ensemble techniques using datasets from different domains. Jiang showed that Boosting can cause over-fitting in the presence of noise [63].

To quantify class noise and understand how it can impact data mining techniques, all experiments in this work were performed on data which (after having been determined to be relatively free of noise) was modified by injecting artificial class noise in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality).

### 1.1.2 High Dimensionality

In addition to noise, many gene expression datasets are also characterized by high dimensionality, which occurs when the dataset contains a very large number of features. In bioinformatics, the problem of high dimensionality is more challenging because most gene expression datasets have thousands (sometimes tens of thousands) of genes and a much smaller sample size [99]. However, the extremely large number of genes makes traditional data mining techniques inefficient and ineffective. With a large number of genes, these techniques become computationally expensive and time consuming. Additionally, it is expected that many of these genes are irrelevant (having little or no correlation with the class) or redundant (containing information already represented in other genes) in relation to the question at hand, subsequently leading to suboptimal results (reduced performance and interpretability of predictive models). Previous studies showed that feature selection (a commonly-used process to alleviate high-dimensionality) can help achieve better performance (i.e. faster learning process, simplified classifiers, and improved model interpretability) by creating a smaller feature subset including only the most important features.

Feature selection techniques can be divided into three major forms: ranker-based techniques, filter-based subset selection, and wrapper-based feature selection. Ranker-based techniques evaluate each feature individually using different statistical methods. Filter-based subset selection techniques also use statistical methods alone, however they evaluate multiple features (subsets) at a time rather than individual features. Finally, wrapper-based techniques use a classifier to directly find the subset of features which performs best. Due to its importance, feature selection has received a lot of attention in the past few years. A broad survey of feature selection is presented by Guyon and Elisseeff [57], who outlined key approaches used for attribute selection, including feature construction, feature ranking, multivariate feature selection, efficient search methods, and feature validity assessment methods.

Most studies in bioinformatics employ ranker-based techniques due to their increased speed compared to the other techniques. Only few studies considered subset-based techniques [62, 114, 100].

Chapter 5 compares feature selection approaches (rankers as well as subset-selection techniques) when learning from high dimensional datasets with varying levels of data quality due to noise injection, motivated by the lack of research which gives a complete perspective on the effectiveness of different feature selection strategies in the context of data noise.

### 1.1.3 Class Imbalance

Class imbalance occurs when instances which belong to the positive class (which is usually the class of interest) are outnumbered by instances of the other class(es). Many real-world bioinformatics datasets are characterized by class imbalance [92, 97, 61]. In such cases, traditional classifiers often result in suboptimal classification performance [59, 108]. Additionally, data mining activities such as attribute selection can be impacted by imbalanced class distributions [18]. Data sampling is the most popular technique for handling class imbalanced data [73], where the dataset is transformed into a more balanced one by adding or removing instances. A comprehensive study on different sampling techniques was performed by Kotsiantis [73], Guo [56], and Van Hulse [104], including both oversampling and undersampling techniques (which add instances to the minority class and remove instances from the majority class, respectively), and both random and directed forms of sampling.

While class imbalance is a frequent problem within bioinformatics datasets, only a few studies investigated this problem and applied techniques to cope with it. In 2005, Al-Shahib et al. [16] applied undersampling to build classifiers to predict protein function from amino acid sequence features. Blagus and Lusa [26] utilized two data sampling techniques (SMOTE and Random Undersampling) to build classifiers on high-dimensional class-imbalanced gene expression datasets.

Chapter 7 investigates the importance of alleviating class imbalance (by applying data sampling) for classification problems on bioinformatics datasets to show the necessity of applying techniques (e.g. data sampling) to alleviate class imbalance.

### 1.1.4 The Combined Problem of High Dimensionality and Class Imbalance

Although these two problems (high dimensionality and class imbalance) are prevalent in bioinformatics, very few studies have addressed both problems simultaneously, let alone considered the problem of data noise. Blagus and Lusa [25] employed data sampling as well as variable selection on class imbalanced data. In a more recent study, Blagus and Lusa [26] performed a study using data sampling on high-dimensional class-imbalanced breast gene expression datasets. Both studies considered only one possible order of feature selection and data sampling while this research uses three approaches to combining feature selection and data sampling.

Al-Shahib et al. [16] used undersampling as well as a wrapper based-feature selection to build classifiers to predict protein function from amino acid sequence features. This paper only investigates a single data sampling technique (i.e. Random Undersampling) while our research uses three: Random Undersampling, Random Oversampling, and SMOTE.

Others [5] have considered the effect of class noise on the classification performance, however, they have used datasets from different application domains with different characteristics which might cause validity problems.

Chapter 8 compares three different approaches for combining feature selection and data sampling using real-world bioinformatics datasets that exhibit both high dimensionality and class imbalance in the context of data quality to determine best practices.

## 1.2 CONTRIBUTIONS

This dissertation involves investigating the different aspects related to data quality (data noise, high dimensionality, and class imbalance), studying their effects on data mining, and providing guidelines on best practices for improving classification performance for bioinformatics data. Research contributions are listed below.

- We evaluate the robustness of ten filter-based feature selection techniques and six classification algorithms to class noise in Chapter 3. Namely, this chapter examines the impact of data quality on classification performance in an attempt to find the best performing feature selection technique and learner that are less sensitive to class noise.

- We provide the first study to evaluate subset-based feature selection (both filter-based subset selection and wrapper-based subset selection) when learning from high dimensional bioinformatics datasets with varying levels data quality due to noise injection in Chapter 4.

- Chapter 5 provides the first comprehensive examination of the effectiveness of three major forms of feature selection when learning from datasets with varying levels of data quality due to noise injection. This comprehensive study provides guidelines on best practices in dealing with high dimensionality in the presence of noise.

- The effectiveness of three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest) is empirically examined in Chapter 6. To our knowledge, this is the first time ensemble classification techniques have been studied when learning from balanced bioinformatics datasets in the context of data quality.

- The importance of alleviating class imbalance (by applying data sampling) for classification problems on bioinformatics datasets is investigated in Chapter 7.

- In Chapter 8 we address the combined problem of high dimensionality and class imbalance in the context of data quality. We compare three different approaches for combining feature selection and data sampling in the context of data quality to determine best practices.

- Chapter 9 provides a comprehensive empirical analysis to give practitioners guidance on best practices when classifying bioinformatics data that exhibit both high dimensionality and class imbalance in the context of data noise.

## 1.3 DISSERTATION STRUCTURE

This dissertation is organized as follows:

- Chapter 2 provides an overview of the methodology followed in the experiments performed throughout the work. This chapter describes the quality of data and noise injection, the datasets, the feature selection techniques, the data sampling techniques, the approaches for combining feature selection and data sampling, the classifiers, and the performance evaluation.

- Chapter 3 evaluates the robustness of ten filter-based feature selection techniques and six classification algorithms to class noise. Namely, this chapter examines the impact of data quality on classification performance in an attempt to find the best performing feature selection technique and learner that are less sensitive to class noise.

- Chapter 4 provides, to our knowledge, the first evaluation of subset-based feature selection techniques when learning from high dimensional bioinformatics datasets with varying levels of data quality.

- Chapter 5 provides the first comprehensive examination of feature selection approaches (rankers as well as subset-selection techniques) when learning from bioinformatics datasets with varying levels of data quality due to noise injection.

- Chapter 6 assesses the effectiveness of three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest) as well as no ensemble classification when learning from balanced bioinformatics datasets with varying levels of data quality.

- Chapter 7 investigates the importance of alleviating class imbalance (by applying data sampling) for classification problems on bioinformatics datasets.

- Chapter 8 compares three different approaches for combining feature selection and data sampling in the context of data quality to determine best practices, and identifies the effectiveness of classification algorithms and feature ranking techniques as well as the choice of feature subset size when learning from high dimensional class imbalanced bioinformatics datasets with varying levels of data quality due to noise injection.

- Chapter 9 provides a comprehensive empirical analysis to give practitioners guidance on best practices when analyzing bioinformatics data that exhibit both high dimensionality and class imbalance in the context of data noise.

- Finally, Chapter 10 presents conclusions and suggestions for future work.

# CHAPTER 2

# METHODOLOGY

This chapter provides an overview of the methodology followed in the experiments. More specifically, Section 2.1 describes our measurement for data quality. Section 2.2 presents the datasets used in the work. Section 2.3 outlines our noise injection process. Section 2.4 represents the feature selection techniques. Section 2.5 discusses the data sampling techniques. Section 2.6 outlines the approaches for combining feature selection and data sampling. Section 2.7 introduces the learners used to create our classification models. Lastly, Section 2.8 presents the cross validation process and discusses the performance metric used in this work.

## 2.1 QUALITY OF DATA

Noise is prevalent among bioinformatics datasets and it can impair the performance of classification models. In this way, data quality can be viewed as the level of noise present in a dataset and how that affects the classification performance. In this work, we obtain the quality of data by first finding the performance of six commonly-used learners: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), and two versions of C4.5 decision trees (C4.5 D and C4.5 N) using the AUC performance metric on the raw dataset with no noise injection or other preprocessing. Then the average AUC across all learners is used to categorize the dataset(s) according to the following ranges: High-Quality ($> 0.8$), Average-Quality ( $\leq 0.8$ and $> 0.7$), and Low-Quality ($\leq 0.7$). All learners and parameters used are explained in Section 2.7 except for the C4.5 D and C4.5 N

| Name | # Minority Instances | Total # of Instances | % Minority Instances | # of Attributes | Average AUC |
|---|---|---|---|---|---|
| Ovarian Cancer [89] | 91 | 253 | 35.97% | 15155 | 0.97388 |
| ALL AML Leukemia [105] | 25 | 72 | 34.72% | 7130 | 0.90908 |
| CNS MAT [33] | 30 | 90 | 33.33% | 7130 | 0.83551 |
| Prostate MAT [33] | 26 | 89 | 29.21% | 6001 | 0.90466 |
| MLL Leukemia [105] | 20 | 72 | 27.78% | 12583 | 0.89615 |
| Lymphoma MAT [45] | 19 | 77 | 24.68% | 7130 | 0.83659 |
| ALL [105] | 79 | 327 | 24.16% | 12559 | 0.84748 |
| Lung Clean [3] | 23 | 132 | 17.42% | 12601 | 0.92351 |
| Lung Cancer [55] | 31 | 181 | 17.13% | 12534 | 0.93885 |
| Lung Michigan [21] | 10 | 96 | 10.42% | 7130 | 0.97384 |

**Table 2.1**: Dataset Characteristics

learners. C4.5 D is the C4.5 decision tree classifier where the default parameters are used and C4.5 N is the C4.5 decision tree classifier with pruning turned off and Laplace smoothing turned on. Note, that this process is only used to determine the quality level of the raw or noise-injected datasets and does not affect the experiment beyond this measurement.

## 2.2 DATASETS

Ten binary (i.e. each instance is assigned one of two class labels) real-world bioinformatics datasets were used in this study. Table 2.1 lists the datasets, including their characteristics in terms of the number of minority instances, total number of instances, percentage of minority instances, and total number of attributes. In addition to the basic properties of each dataset, the table presents the average AUC across the six learners discussed in Section 2.1. All of these datasets have average AUC values greater than 0.8, thus they qualify as High-Quality data according to our measure in Section 2.1.

## 2.3 NOISE INJECTION

We used the same noise injection mechanism proposed by Van Hulse et al. [103] where class noise is injected into the training datasets using two simulation parameters. That is, the levels of class noise are regulated by two noise parameters. The first parameter, denoted $\alpha$ ($\alpha = 10\%, 20\%, 30\%, 40\%, 50\%$), is used to determine the overall class noise level in the data. Precisely, $\alpha$ is the noise level relative to the number of instances belonging to the positive class, i.e., the number of examples to be injected with noise is $2 \times \alpha \times |P|$, where $|P|$ is the number of examples in the smaller class (which is often the positive class). This ensures that the positive class is not drastically impacted by the level of corruption, especially if the data is highly imbalanced. The second parameter, denoted $\beta$ ($\beta = 0\%, 25\%, 50\%, 75\%, 100\%$), represents the percentage of class noise injected in the positive instances and is referred to as noise distribution. Note also that because the number of instances to be corrupted is tied to the number of minority-class instances, the quantity of noise injected into the dataset can be somewhat misleading: more imbalanced datasets will be injected with less noise overall, even at higher noise levels. With five values for $\alpha$ and $\beta$, there are 24 different noise injection patterns (because the case with $\alpha = 50\%$ and $\beta = 100\%$ would convert all positive-class instances into negative-class instances, leaving no counterexamples to learn from).

## 2.4 FEATURE SELECTION

Feature selection techniques can generally be grouped into two broad categories: ranker-based techniques and subset-based techniques. Ranker-based techniques evaluate each feature individually using different statistical measures, while subset-based techniques evaluate whole subsets at a time either using statistical measures (filter-based subset selection) or using a classifier (wrapper-based feature selection). Feature rankers are much less computationally expensive than other feature selection techniques because a ranker need only provide a single score for each feature and then subsets can be built based on ranked feature lists.

On the other hand, subset-based selection techniques evaluate subsets (groups of features) rather than each individual feature. Thus, the number of calculations reaches to $O(2^n)$ if all possible subsets are evaluated. Although more efficient methods exist, subset-based methods will nonetheless take more computational resources than feature rankers. It is of note that subset-based selection techniques have the advantage of being able to detect redundancy (highly correlated features in the selected set) among features. Detecting redundant features is important in reducing the size and enhancing the comprehensibility of the final classification model. Thus, in principle, subset-based selection techniques can give better performance than feature rankers.

Three different approaches to feature selection are considered in this work: filter-based feature ranking, filter-based subset selection, and wrapper-based subset selection. Filter-based feature ranking techniques score each feature individually and the top N features are used. More information on this is found in Section 2.4.1. With the two subset evaluation-based groups, though, a search technique must be used to explore the space of all possible feature subsets, to reduce the problem from being $O(2^n)$. Based on preliminary experimentation, we chose the Greedy Stepwise approach [31] for all subset evaluation experiments, which uses forward selection to build the full feature subset starting from the empty set. At each point in the process, the algorithm creates a new family of potential feature subsets by adding every feature (one at a time) to the current best-known set. The merits of all these sets are evaluated, and whichever performs best is the new best-known set. This algorithm stops when none of the new sets outperforms the previous best-known set, or when a user-defined maximum number of features (in our study, 100) is reached.

### 2.4.1 Filter-Based Feature Ranking

Ten feature ranking techniques were used in this study: Gini Index (GI), Kolmogorov-Smirnov statistic (KS), Mutual Information (MI), Probability Ratio (PR), Area Under the Receiver Operating Characteristic Curve (ROC), Area Under the Precision Recall Curve (PRC), Signal-to-Noise Ratio (S2N), Wilcoxon Rank Sum (WRS), Significance Analysis of Microarrays (SAM), and Chi Squared (CS). These can be divided into three groups: threshold-based feature selection techniques (TBFS) which use the feature values as posterior probabilities to estimate classification errors (this includes GI, KS, MI, PR, ROC, and PRC), first-order-statistics based techniques (FOS) that employ mean and standard deviation values to determine feature relevance (this includes S2N, WRS, and SAM), and techniques commonly used in the literature (this includes CS). In the following sections, we discuss each of these families of techniques in greater detail. All techniques were implemented within the WEKA machine learning framework [115] by our research team.

*Threshold-Based Feature Selection Techniques (TBFS)*

These feature ranking techniques were proposed and implemented recently by our research group [107, 44]. Six of the techniques used in this work (GI, KS, MI, PR, ROC, and PRC) fall under the category of TBFS techniques. In TBFS, each attribute is evaluated against the class, independent of all other attributes 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 (e.g., whether instances with values above the threshold are considered positive or negative class examples). The normalized values are treated as posterior probabilities: however, no real classifiers are being built. Instead, these ersatz posterior probabilities are used to calculate various classifier performance metrics, and the results of these metrics are the quality of the feature being examined.

1. Gini Index (GI) measures the impurity of a dataset [27]. 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:

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

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 [27].

2. The Kolmogorov-Smirnov Statistic (KS) [44] measures a feature's relevance by dividing the data into clusters based on the class and comparing the distribution of that particular attribute among the clusters. It is effectively the maximum difference between the curves generated by the true positive and false positive rates ($TPR(t)$ and $FPR(t)$) of the ersatz "classifier" as the decision threshold changes from 0 to 1, and its formula is given as follows:

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

3. Mutual Information (MI) [87] computes the mutual information criterion with respect to the number of times a feature value and a class co-occur, the feature value occurs without the class, and the class occurs without the feature value. Mutual information is defined as:

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

where $y(x)$ is the actual class of instance $x$, $\hat{y}^t(x)$ is the predicted class based on the value of the attribute $X_j$ at a threshold $t$,

$$p\left(\hat{y}^t = \alpha, y = \beta\right) = \frac{\left|\left\{\left(x \,\middle|\, \hat{X}^j(x) = \alpha\right) \cap (y(x) = \beta)\right\}\right|}{|P| + |N|}$$

$$p\left(\hat{y}^t = \alpha\right) = \frac{|\{(x|y(x) = \alpha)\}|}{|P| + |N|}$$

$$\alpha, \beta \in \{P, N\}$$

Note that the class (actual or predicted) can be either positive ($P$) or negative ($N$).

4. Probability Ratio (PR) [49] is the sample estimate probability of the feature given the positive class divided by the sample estimate probability of the feature given the negative class [49].

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

5. Area Under the ROC Curve (ROC) [90], the area under the receiver operating characteristic (ROC) curve, is a single-value measure based on statistical decision theory and was developed for the analysis of electronic signal detection. It is the result of plotting $FPR(t)$ against $TPR(t)$. In this study, ROC is used to determine each feature's predictive power. ROC curves are generated by varying the decision threshold $t$ used to transform the normalized attribute values into a predicted class. That is, as the threshold for the normalized attribute varies from 0 to 1, the true positive and false positive rates are calculated.

6. Area Under the PRC Curve (PRC) [44], the area under the precision-recall characteristic curve, is a single-value measure depicting the trade-off between precision and recall. It is the result of plotting $TPR(t)$ against precision, $Pre(t)$. Its value ranges from 0 to 1, with 1 denoting a feature with highest predictive power. The PRC curve is generated by varying the decision threshold $t$ from 0 to 1 and plotting the recall (x-axis) and precision (y-axis) at each point in a similar manner to the ROC curve.

*First Order Statistics Feature Selection Techniques (FOS)*

This section presents a set of three 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 [69] combined these and other related techniques into a single family and studied their similarity to each other, and how they perform in classification.

1. Signal-to-Noise ratio (S2N) [74] as it relates to classification or feature selection, represents how well a feature separates two classes. The equation for signal to noise is:

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

   Where $\mu_P$ and $\mu_N$ are the mean values for the feature from the positive class and negative class, respectively, and $\sigma_P$ and $\sigma_N$ are the corresponding standard deviations.

2. Wilcoxon Rank Sum [30] (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 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:

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

3. Significance Analysis of Microarrays [102] (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:

$$SAM = (\mu_P - \mu_N)/(\sigma^* + \sigma_0)$$

where $\sigma_0$ represents the exchangeability constant and $\sigma^*$ represents an overall standard deviation. The role of $\sigma_0$ is to prevent attributes whose standard deviations are small from having large score. $\sigma_0$ is a customizable factor which is generally the top 90-percentile of standard deviations. For this experiment we use this value. $\sigma^*$ which is calculated as:

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

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.

*Commonly Used Feature Selection Techniques*

Finally, one technique commonly used in the machine learning literature (and not fitting into either of the above families) was used:

1. Chi Squared [76]. This method utilizes the $\chi^2$ statistic, evaluating features independently with respect to the class labels. The larger the chi-squared, the more relevant the feature is with respect to the class. The values of the features must first be discretized into a number of intervals using some discretization method [77]. The chi-squared value of each feature is computed as:

$$\chi^2 = \sum_{i=1}^{I} \sum_{j=1}^{B} \frac{\left[A_{ij} - \frac{R_i \times B_j}{N}\right]^2}{\frac{R_i \times B_j}{N}}$$

where $I$ denotes the number of intervals, $B$ the number of classes, $N$ the total number of instances, $R_i$ the number of instances in the $i$th interval, $B_j$ the number of instances in the $j$th class, and $A_{ij}$ the number of instances in the $i$th interval and $j$th class. The larger this chi-squared statistic, the more unlikely it is that the distributions of values and classes are independent; that is, they are related, and the feature in question is relevant to the class. Note that for the Chi Squared approximation to be valid, the test requires a sufficient sample size.

### 2.4.2 Filter-Based Subset Evaluation

Two filter-based feature subset selection techniques were used in this research: Correlation Based Feature Selection and Consistency. These were chosen due to being the most widely-used in the literature, as well as being the only two with implementations in WEKA machine learning software [115]. In the following sections, we discuss each of these techniques in greater detail.

1. Correlation-Based Feature Selection (CFS) [58]. This employs the Pearson correlation coefficient, a correlation metric designed to balance the need to have the features correlate with the class and the need to have the features not correlate with one another. The Pearson correlation coefficient is found with the following formula:

$$M_S = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}}$$

   In this formula, $M_S$ is the merit of the current set of features, $k$ is the number of features, $\overline{r_{cf}}$ is the mean of the correlations between each feature and the class, and $\overline{r_{ff}}$ is the mean of the pairwise correlations between every two features. In both cases, correlations are calculated using symmetric uncertainty, an information-theoretic measure of how changes in one feature affect the uncertainty of the other, and which compensates for inherent entropy in either feature. As desired, the numerator increases when the set of features is particularly good at classifying the data, while the denominator increases when the set has a significant amount of self-correlation, which implies redundancy.

2. Consistency [37]. Seeks to find the largest possible feature subset which is consistent. This is most easily understood in terms of its converse, inconsistency: a feature subset is considered "inconsistent" if two instances share all the same values for the given features but differ in their class variables. Mathematically, the inconsistency of a feature subset is found by first going through the dataset and finding all unique patterns produced by the current feature mask. A pattern is a tuple of feature values, considering only the features in the current feature set and not including the class attribute. The total number of instances which match a given pattern is represented by $n$, while the number of instances in classes $C_1$,$C_2$, $C_3$, etc. are represented by $c_1$, $c_2$, $c_3$, and so on. Without loss of generality, assume that class $C_3$ has the most instances for the chosen pattern, and thus $c_3$ is the largest of the $cs$; in this case, the inconsistency count of the pattern is found by $n - c_3$. To find the inconsistency count of the whole dataset, the individual counts from each pattern are added together, and the final result is divided by the total number of instances in the original dataset. Lower values of inconsistency count are preferred, because these have greater consistency.

### 2.4.3 Wrapper-Based Subset Selection

Wrapper-based subset selection evaluates subsets of features using a classifier. The chosen subset is used to build a classification model, and the performance of this model is then used as the score for that feature subset. In this work, we use the Naïve Bayes (discussed further in Section 2.7) classifier to build our models. We selected this learner because in practice it is a very effective classifier on a wide variety of datasets [46], as well as its relative simplicity compared to other learners (e.g., SVM and MLP). Note that Naïve Bayes was used within the wrapper regardless of the learner which would be eventually used to build the classification model. We use the AUC (Area Under the ROC Curve) metric as the performance metric within the wrapper. The AUC (discussed further in Section 2.8) performance metric has also been proven to be statistically consistent [64].

## 2.5 SAMPLING TECHNIQUES

In this study, we apply three common sampling techniques. These three techniques have been shown to be effective at improving classification performance in previous research [104]:

1. Random undersampling (RUS): Randomly delete instances from the majority class until the target class ratio is reached.

2. Random oversampling (ROS): Randomly duplicate instances in the minority class until the target class ratio is reached.

3. Synthetic minority oversampling technique (SMOTE) [32]: Add artificially-generated minority samples by extrapolating between preexisting minority instances. The technique chooses a random minority-class instance, finds its k nearest neighbors which are also minority class instances, and generates an artificial instance a randomly-chosen distance between the chosen instance and one of these nearest neighbors. Instances continue to be added in this fashion until the desired class ratio is reached.

Additionally, we performed sampling to obtain a balanced class ratio: 50:50 majority:minority. That is to say, sampling was performed until the number of majority-class instances was equal to the minority-class instances.

## 2.6 APPROACHES FOR COMBINING FEATURE SELECTION AND DATA SAMPLING

Feature selection and data sampling have become necessary steps when analyzing high dimensional class imbalanced bioinformatics datasets. Although these two techniques have received tremendous attention, most work has utilized them separately. However, applying them jointly to improve the classification performance has not been thoroughly explored.

We investigated three approaches that are used to deal with both high dimensionality and class imbalance. All approaches combine feature selection and data sampling; the difference between one approach and another is the order in which they are applied (whether sampling takes place before or after feature selection) and the dataset (unsampled or sampled) used for classification. We excluded two other approaches, where only one technique (feature selection or data sampling) is used alone, because all datasets investigated in this research are imbalanced and exhibit high dimensionality. Both feature selection and sampling are necessary to help alleviate class imbalance and cope with high dimensionality.

The three approaches are outlined [15] in Figure 2.1. In the first approach (DS-FS-UnSam), data sampling takes place before feature selection is performed, and then a classifier is built using the selected features and the unsampled data. In the second approach (DS-FS-Sam), data sampling also takes place before feature selection is performed, however, a classifier is built using the selected features and the sampled data. On the other hand, in the third approach (FS-DS), feature selection takes place before data sampling is performed, and then a classifier is built using the selected features and the sampled data.

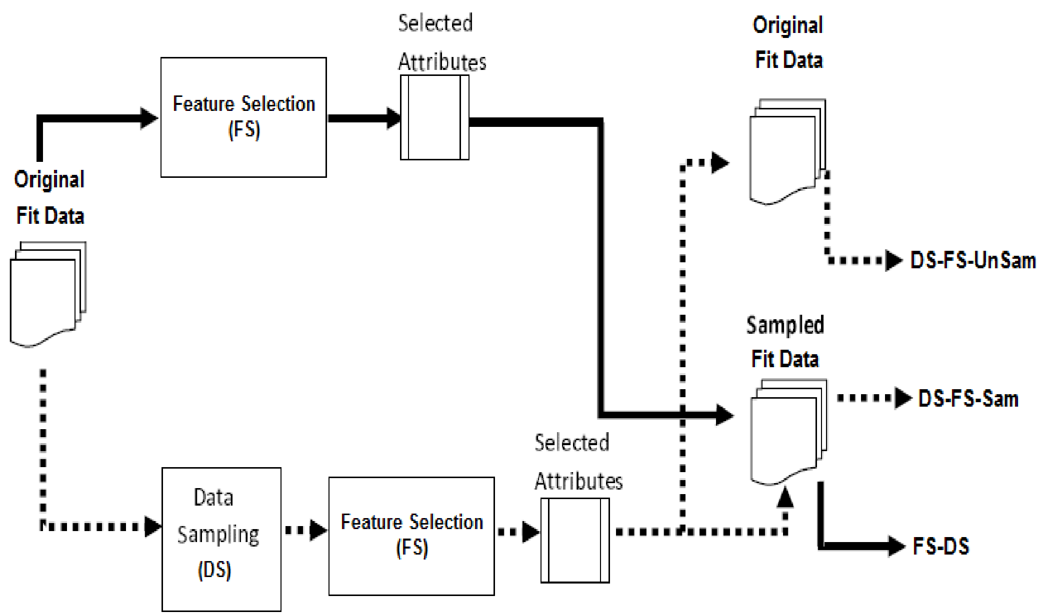



**Figure 2.1**: Feature Selection and Data Sampling Approaches

## 2.7 CLASSIFIERS

Six classifiers were used in this study: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). These were selected due to their diversity and their prevalence in the literature. All classifiers were built using the Weka machine learning software [115], using the default parameters unless noted otherwise. Previous research has shown that the changes described below are appropriate for improving classification models [104].

Naïve Bayes (NB) [81] is a simple Bayesian classifier which uses Bayes's Theorem to find the posterior probability of an instance being in each class based on its feature values. Although mathematically it depends on all features being independent of (unrelated to) one another, in practice it is a very effective classifier on a wide variety of datasets. No changes to the default parameters were made in our experiments.

Multilayer Perceptron (MLP) [24] is a neural-net-based classifier, using a simple feedforward network with distinct layers. Each layer is fully connected to the ones before and after it, and neurons compute their outputs by finding the weighted sum of their inputs and passing this to a sigmoid function. In these experiments, the `hiddenLayers` parameter was set to 3 to build a network with one hidden layer containing three nodes, and the `validationSetSize` parameter was set to 10 so that the classifier would leave 10% of the instances out to determine when to stop training.

$k$-nearest neighbors [50], or $k$-NN, learner is an example of an instance based and lazy learning algorithm. Instance based algorithms use only the training data without creating statistics on which to base their hypotheses. The $k$-NN learner does this by calculating the distance of the test sample from every training instance, and the predicted class is derived from the k nearest neighbors. In the $k$-NN learner, when there is a test sample that needs to be classified, the classes for each of the $k$ closest training samples (a $k$ of five was used in this research, hence the name "5-NN") are tabulated and the weight of each neighbor is determined by taking a measurement of $\frac{1}{distance}$ where distance is the *distance* from the test sample. After the classes and weights are tabulated, all of the weights from the neighbors of the positive class together and all of the weights of the negative class are added together. The prediction will be the class with the largest cumulative weight.

Support Vector Machines (SVM) [35] is a linear discriminant classifier which assumes that the best discriminant maximizes the distance between the two classes. This is measured in the distance from the discriminant to the samples of both classes [79]. In Weka, the complexity parameter "c" was changed from 1.0 to 5.0 and `buildLogisticModels` which allows proper probability estimates to be obtained, was set to true. In particular, the SVM learner used a linear kernel.

Random Forest (RF100) [29] is an ensemble classifier implemented in the WEKA data mining tool. Specifically, Random Forest builds a set of unpruned decision trees and classifies an instance based on the majority vote of the decision trees. The number of trees used is determined by the `numTrees` attribute. Previous research [71] shows that the optimum number of trees is 100, so that is the number used in this study. After the trees are created the learner begins testing the instances. Each instance is passed through each tree and the predicted class is chosen. The class that is chosen by the most trees becomes the chosen class of the instance.

Logistic Regression (LR) [60] is a very simple classifier; it shares much in common with a linear regression, with the output run through a logistic function. In this study, the Weka default parameter settings were used for this classifier.

## 2.8 PERFORMANCE EVALUATION

We used four runs of five-fold cross-validation [72] to build and test our models. Cross-validation allows for all instances to participate both in training and testing models, without the risk of overfitting which can come from using an instance for both at the same time. The cross-validation process begins by dividing the data into $N$ equal-size subsets (folds), and then one of these was held aside as a test (hold-out) fold. The remaining $N-1$ folds, collectively called the training fold, first had noise injected according to one of the 24 noise patterns, and then models were built on this noisy training fold and evaluated on the clean test fold. This process is repeated $N$ times, so that each fold is used as the hold-out fold exactly once. Once all $N$ folds have been used as the test datasets, the results from all test datasets are integrated into a single performance value for that dataset. The value $N = 5$ was used in this work and we repeat this process 4 times for a total of 20 times for each of the datasets.

The performance of all classification models is evaluated using the area under the receiver operating characteristic curve (AUC) [96]. This performance metric was chosen because it is commonly used in the literature. The ROC curve plots the graph of the true positive rate on the $y$-axis versus the false positive rate on the $x$-axis as the classifier decision threshold is varied from 0 to 1. The area under this curve is used as a single numerical metric to indicate the performance of the classifier. Although we used the area under the receiver operating characteristic curve as both a classifier performance metric (AUC) and as a feature ranker (ROC), these uses are independent of each other.

# CHAPTER 3

# HOW RANKER AND LEARNER CHOICE AFFECTS CLASSIFICATION PERFORMANCE ON NOISY BIOINFORMATICS DATA

## 3.1 INTRODUCTION

Two challenging problems are often encountered in real-world gene expression datasets: (1) noise, and (2) high dimensionality. Noise refers to incorrect or missing values in datasets which can be caused by faulty microarray chips, insufficient resolution, and image corruption. Noise has a detrimental impact on both classification models and feature selection techniques, confusing data mining techniques and subsequently leading to suboptimal results (worsened classification performance, unstable feature selection). For this reason, it is important to study and quantify this noise, and understand how it can impact data mining techniques (feature selection techniques and classification models).

High dimensionality is another challenge encountered in real-world gene expression datasets, which occurs because there are large number of attributes (genes). Most of these attributes are redundant (containing information already represented in other attributes) or useless (not having much correlation with the class) for building a predictive model. The primary technique used to cope with high dimensionality is known as feature selection, which chooses the attributes/features (e.g., gene markers, for microarray data) which are most highly correlated with the class attribute (e.g., whether a sample comes from a cancerous or noncancerous patient). Feature selection can improve classification performance of predictive models, improve model interpretability, and speed up the learning process. Thus, feature selection techniques are not only useful but often necessary.

In this chapter [7], we evaluate the robustness of ten filter-based feature selection techniques and six classification algorithms, examining the impact of data quality level (artificially created using noise injection in a controlled fashion) on classification performance. In this study, we inject 24 different class noise patterns into 12 bioinformatics datasets (all of which started out being noise-free) creating three data quality levels (High-Quality, Average-Quality, and Low-Quality). After this step, we used ten feature rankers (with four feature subset sizes) and six classification techniques to build models from each of the noisy datasets, and examined the results for each group of data quality levels. This is the first study to consider the role of feature selection in the context of bioinformatics datasets which have varying levels of data quality through the injection of artificial class noise.

## 3.2 CONTRIBUTIONS

The primary contribution of this chapter is to provide guidance on best performing learner and best performing filter-based feature selection technique that are less sensitive to class noise and are safe choices for machine learning. In this study we: (1) investigate the robustness of ten feature ranking techniques (with four subsets: 25, 50, 100, and 200) to class noise; (2) simulate real-world scenarios by injecting class noise into twelve real-world gene-expression datasets (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality) and (3) employ six classifiers that are commonly-used in the literature.

The remainder of this chapter will be organized as follows: Section 3.3 presents related works on the topics of high dimensionality and data noise. Section 3.4 outlines the methods used in this chapter. In Section 3.5, we present our results. Finally, Section 3.6 presents concluding remarks for this chapter.

## 3.3 RELATED WORK

High dimensionality is a significant problem in data mining, and occurs when the dataset contains a very large number of features. This can make machine learning computationally expensive and reduce the prediction accuracy of classifiers, because in most cases some of these attributes are redundant (containing information already represented in other attributes) or useless (not having much correlation with the class) for building an inductive model. Feature selection reduces high dimensionality by finding a minimum subset of features that have the highest correlation with the class, removing irrelevant and redundant attributes. Much work has been done on feature selection, comparing different techniques and finding the best number of features to include in datasets. Guyon and Elisseeff [57] gave a general review of attribute selection, including feature construction, feature ranking, multivariate feature selection, efficient search methods, and feature validity assessment methods. Li et al. [75] compared eight feature selection techniques on nine multi-class gene expression datasets. They concluded that there was no clear winner among the feature selection techniques, and the accuracy of the classification was highly dependent on the choice of classifier rather than the choice of feature selection method.

Noise is another challenging problem in data mining. This problem is found when a dataset has errors or missing values. Noise can occur in the dependent feature when an instance is assigned to a wrong class (or if the class assignment is missing) and in the independent features when there are incorrect or missing values in the independent features. The former is called class or label noise while the latter is known as attribute noise. Zhu and Wu [117] showed that class noise has a more detrimental impact on classifier performance than attribute noise. A comprehensive survey on the different types of label noise, their consequences, and the algorithms that consider label noise can be found in the work of Frénay and Verleysen [51]. Many efforts have been spent on noise-tolerant inductive learners [67].

These two important problems (high dimensionality and noise) in the area of data mining and machine learning have been well studied in the past few years. However, most of the work has focused on dealing with each problem separately. Few studies focused on both problems simultaneously. Some [109] focused on the stability of feature ranking techniques (robustness of outputs in the face of perturbation) in the presence of noise, where the impact of class noise (data perturbation due to class noise injection) on the stability of feature selection techniques is investigated. Others [5] consider the effect of class noise on the classification performance, without examining the validity problems due to the use of datasets from different application domains and the different characteristics of the datasets.

Although few works have examined the challenges of noise injection and high dimensionality, relatively little work has considered the two problems simultaneously in the context of varying levels of data quality due to the presence of noise. Data noise can lead to poor results across a wide range of classification algorithms, and thus all experiments in this chapter were performed on data which was modified by injecting artificial class noise in a controlled fashion. To the best of our knowledge this is the first work to study both noise injection and high dimensionality in the context of data quality using gene expression datasets.

## 3.4 METHODOLOGY

In this chapter, ten feature ranking techniques (with four subsets: 25, 50, 100, and 200) were used: Gini Index (GI), Kolmogorov-Smirnov statistic (KS), Mutual Information (MI), Probability Ratio (PR), Area Under the Receiver Operating Characteristic Curve (ROC), Area Under the Precision Recall Curve (PRC), Signal-to-Noise Ratio (S2N), Significance Analysis of Microarrays (SAM), Wilcoxon Rank Sum

| Name | # Minority Instances | Total # of Instances | % Minority Instances | # of Attributes | Average AUC |
|---|---|---|---|---|---|
| BCancer 50k [44] | 200 | 400 | 50.00% | 54614 | 0.85636 |
| Ovarian Cancer [105] | 91 | 253 | 35.97% | 15155 | 0.97388 |
| ALL AML Leukemia [105] | 25 | 72 | 34.72% | 7130 | 0.90908 |
| CNS MAT [33] | 30 | 90 | 33.33% | 7130 | 0.83551 |
| Colon 50k [44] | 130 | 400 | 32.50% | 54614 | 0.85323 |
| Prostate MAT [33] | 26 | 89 | 29.21% | 6001 | 0.90466 |
| MLL Leukemia [105] | 20 | 72 | 27.78% | 12583 | 0.89615 |
| Lymphoma MAT [33] | 17 | 77 | 24.68% | 7130 | 0.83659 |
| ALL [105] | 79 | 327 | 24.16% | 12559 | 0.84748 |
| Lung Clean | 23 | 132 | 17.42% | 12601 | 0.92351 |
| Lung Cancer [106] | 31 | 181 | 17.13% | 12534 | 0.93885 |
| Lung Michigan [21] | 10 | 96 | 10.42% | 7130 | 0.97384 |

**Table 3.1**: Details of the Datasets

(WRS), and Chi Squared (CS). These feature ranking techniques are all discussed in greater detail in Section 2.4.1. Additionally, six classifiers were used (discussed in Section 2.7): Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). Twelve binary (i.e. each instance is assigned one of two class labels) real-world bioinformatics datasets were used in this study. Table 2.1 lists the datasets and their characteristics. We created three levels of data quality (High-Quality, Average-Quality, and Low-Quality) artificially by injecting noise into these datasets which were first determined to be free of noise. Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively. For all experiments in this chapter, we used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8.

| Learner | Ranker | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CS | PR | GI | MI | KS | ROC | PRC | S2N | WRS | SAM |
| NB | 0.958009 | 0.925117 | 0.924747 | 0.960844 | 0.960505 | 0.959871 | 0.957474 | 0.956678 | 0.958624 | 0.954577 |
| MLP | 0.956045 | 0.940836 | 0.941180 | 0.958364 | 0.960348 | 0.961663 | 0.956034 | 0.953952 | 0.961508 | 0.949479 |
| 5NN | 0.964492 | 0.928781 | 0.928161 | 0.964931 | 0.967278 | 0.965749 | 0.959432 | 0.959253 | 0.965613 | 0.959907 |
| SVM | 0.937670 | 0.925342 | 0.925574 | 0.940289 | 0.943247 | 0.944678 | 0.939117 | 0.939836 | 0.944551 | 0.929670 |
| RF100 | **0.976732** | **0.963136** | **0.963097** | **0.977224** | **0.977974** | **0.978680** | **0.976430** | **0.974029** | **0.978661** | **0.972589** |
| LR | *0.827167* | *0.817515* | *0.817715* | *0.833160* | *0.843483* | *0.845948* | *0.832534* | *0.833172* | *0.845476* | *0.823007* |
| Average | 0.936686 | 0.916788 | 0.916746 | 0.939135 | 0.942139 | 0.942765 | 0.936837 | 0.936153 | 0.942405 | 0.931538 |

**Table 3.2**: Average AUC for High-Quality datasets

## 3.5 RESULTS AND ANALYSIS

The experiment was conducted using 12 noise-free bioinformatics datasets. We created 288 noisy datasets by injecting 24 different combinations of noise level and noise distribution into the 12 noise-free datasets. The resulting noisy datasets were then defined as "High-Quality," "Average-Quality," "or Low-Quality," (160, 79, and 49 datasets, respectively) according to our measure in Section 2.1. For all datasets, we applied ten filter-based feature ranking techniques (considering four choices for feature subset size), and we used six learners (NB, MLP, 5-NN, SVM, RF100, and LR) to build predictive models, and performance was measured using AUC. To avoid any validity problems related to overfitting we used four runs of five-fold cross-validation to build and test our models. The results are presented in Tables 3.2 through 3.4. Each table presents the average classification performance (in terms of AUC) for every combination of data quality, learner, and ranker, across all 4 subset sizes and all appropriate datasets (those which reached the relevant data quality level when injected with noise). Within each column, **bold** values represent the best performance for that ranker, and *italics* values represent the worst performance for that ranker. The tables also present the average performance (last row of the tables) of each ranker over the six learners.

The results demonstrate that RF100 outperforms all other learners across all rankers for High-Quality and Average-Quality datasets. When considering Low-

| Learner | Ranker | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CS | PR | GI | MI | KS | ROC | PRC | S2N | WRS | SAM |
| NB | 0.896830 | 0.845754 | 0.841334 | 0.902730 | 0.898680 | 0.899346 | 0.900591 | 0.892111 | 0.896050 | 0.897935 |
| MLP | 0.873175 | 0.847624 | 0.845770 | 0.876051 | 0.878747 | 0.878635 | 0.870443 | 0.869103 | 0.880212 | 0.864398 |
| 5NN | 0.878132 | 0.809302 | 0.798609 | 0.883490 | 0.889789 | 0.882686 | 0.866398 | 0.859325 | 0.882487 | 0.878493 |
| SVM | 0.839915 | 0.821528 | 0.816703 | 0.845312 | 0.846425 | 0.850127 | 0.841264 | 0.845156 | 0.850103 | 0.835358 |
| RF100 | **0.912477** | **0.881345** | **0.872925** | **0.915911** | **0.916611** | **0.917557** | **0.913059** | **0.909449** | **0.917590** | **0.909486** |
| LR | *0.727882* | *0.717533* | *0.719359* | *0.732093* | *0.734584* | *0.739044* | *0.727976* | *0.732277* | *0.738897* | *0.724234* |
| Average | 0.854735 | 0.820514 | 0.815783 | 0.859265 | 0.860806 | 0.861233 | 0.853289 | 0.851237 | 0.860890 | 0.851651 |

**Table 3.3**: Average AUC for Average-Quality datasets

| Learner | Ranker | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | CS | PR | GI | MI | KS | ROC | PRC | S2N | WRS | SAM |
| NB | 0.753682 | **0.732808** | **0.724379** | **0.786103** | 0.765448 | 0.776546 | **0.784980** | 0.756023 | 0.770274 | **0.795850** |
| MLP | 0.743490 | 0.720354 | 0.720797 | 0.750186 | 0.752893 | 0.753741 | 0.747866 | 0.734422 | 0.757395 | 0.743245 |
| 5NN | 0.706376 | 0.642701 | 0.636494 | 0.710639 | 0.735163 | 0.722119 | 0.693448 | 0.698584 | 0.720897 | 0.731511 |
| SVM | 0.714982 | 0.700010 | 0.698185 | 0.724994 | 0.727786 | 0.730560 | 0.720082 | 0.724880 | 0.732023 | 0.717365 |
| RF100 | **0.773850** | 0.728134 | 0.718775 | 0.785362 | **0.795168** | **0.792988** | 0.779497 | **0.786127** | **0.792961** | 0.790445 |
| LR | *0.640624* | *0.635217* | *0.633962* | *0.644617* | *0.644739* | *0.649423* | *0.643023* | *0.638811* | *0.648713* | *0.634812* |
| Average | 0.722167 | 0.693204 | 0.688765 | 0.733650 | 0.736866 | 0.737563 | 0.728149 | 0.723141 | 0.737044 | 0.735538 |

**Table 3.4**: Average AUC for Low-Quality datasets

Quality datasets, RF100 is again among the top learners, either the best or second best (after NB) an equal number of times. This indicates that the RF100 is the least sensitive to class noise and a good candidate for classification across all data quality levels. The results also show that LR shows the worst performance across all rankers and data quality levels.

Furthermore, we see that CS, MI, KS, ROC, WRS, and SAM are particularly robust and are able to improve the classification performance on Low-Quality datasets for all learners except LR. With these rankers, all of the learners (other than LR) had performance values above 0.7, meaning that the rankers turned Low-Quality datasets into Average-Quality datasets. There was no ranker that was able to improve the classification performance on Low-Quality datasets for LR, however, where the average AUC remained below 0.7 (i.e. Low-Quality). By the same token, no ranker was able to improve the classification performance on Average-Quality datasets for LR, where the average AUC remained below 0.8 (i.e. Average-Quality).

Although no ranker significantly outperformed the others, ROC shows the best average performance (across all learners) for all quality levels, while GI shows the worst performance across all data quality levels on average. The best average AUC was obtained when the learner RF100 was used with ROC for High-Quality datasets (0.978680), and for Average-Quality datasets when the learner RF100 was used with WRS (0.917590). When considering Low-Quality datasets, the best average AUC was obtained when the learner NB was used with SAM (0.795850). On the other hand, the worst average AUC was obtained when the learner LR was used with PR for High-Quality and Average-Quality datasets (0.817515, 0.717533) respectively, while the worst performance was obtained when the learner LR was used with GI for Low-Quality datasets (0.633962).

### 3.6 CHAPTER SUMMARY

In this chapter we evaluated the robustness of ten filter-based feature selection techniques and six classification algorithms, examining the impact of data quality level on classification performance. We used 12 noise-free bioinformatics datasets along with 24 noise injection patterns to create three collections of High-Quality, Average-Quality, and Low-Quality data. We then used ten feature rankers and six classification techniques to build models from each of the noisy datasets, and examined the results for each group of data quality levels to determine the best performing techniques that are less sensitive to class noise and are able to help improve the classification performance of learners.

The experimental results suggest that RF100 is the least sensitive learner to class noise and a good candidate for classification across all rankers and data quality levels, while LR shows the worst performance across all rankers and data quality levels. Additionally, we find that the CS, MI, KS, ROC, WRS, and SAM rankers are particularly robust and are able to improve the classification performance on the Low-

Quality datasets for all learners except LR. Finally, there was no ranker that was able to significantly improve the classification performance on Low-Quality datasets and Average-Quality datasets for LR. Overall, using the KS, ROC, or WRS rankers along with the RF100 learner gave the best results for the High-Quality and Average-Quality datasets, while the SAM ranker with the NB learner was best for the Low-Quality datasets (although using KS, ROC, or WRS along with RF100 still performed quite well).

# CHAPTER 4

# EVALUATION OF SUBSET-BASED FEATURE SELECTION USING BIOLOGICAL DATA WITH VARYING LEVELS OF DATA QUALITY

## 4.1 INTRODUCTION

High dimensionality is one of the common characteristics exhibited by many real-world gene expression datasets. High dimensionality refers to datasets where a large number of features describe each instance, with the number of features sometimes exceeding the number of instances. This large number of features makes the analysis of such datasets more challenging, as in general, most of these features will be irrelevant to the problem at hand. The process of eliminating irrelevant features is known as feature selection, which can lead to better performance by reducing computation time, increasing the prediction accuracy of inductive models, and improving model interpretability. There are two broad categories of feature selection: ranker-based techniques and subset-based techniques. Ranker-based techniques evaluate each feature individually using different statistical measures, while subset-based techniques evaluate whole subsets at a time either using statistical measures (filter-based subset selection) or using a classifier (wrapper-based feature selection). For researchers and practitioners in bioinformatics a small set of features or genes is very desirable or required. Subset-based feature selection is particularly useful here because it provides a small set of features which are unique and not highly correlated with other features in the selected set.

Noise is another difficulty encountered in many real-world gene expression datasets. Noise refers to erroneous (incorrect or missing) values in datasets, which can occur in the dependent value (class noise) or the independent values (attribute noise). Noise can hinder data mining techniques and subsequently result in poor classification performance across a wide range of classification algorithms. To the best of our knowledge no previous works have investigated the effectiveness of subset-based feature selection in the context of data quality.

In this chapter [10, 4], we create three tiers of data quality (High-Quality, Average-Quality, and Low-Quality) by injecting class noise in a controlled fashion into ten gene expression datasets which were first determined to be noise-free. The more noisy the dataset, the lower the data quality, and vice versa. By injecting class noise into a noise-free dataset we avoid any validity problems related to injecting class noise into already noisy datasets. To counter high dimensionality we employ three subset-based feature selection techniques: Correlation-based Feature Selection (CFS), Consistency, and wrapper-based feature selection using the Naïve Bayes learner (WrapNB). Classification models were then built using either all features (no feature selection is performed "No FS") or the selected features, using four commonly used classifiers (Naïve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, and Support Vector Machines). The evaluation was carried out using the area under the ROC curve (AUC) classifier performance metric. This allowed us to discover the effectiveness of subset-based feature selection under different data quality levels.

## 4.2 CONTRIBUTIONS

Compared to rankers, subset-based feature selection techniques have the advantage of selecting a small set of features which are unique and not highly correlated with other features in the selected set. Due to the computational complexity required by subset-based feature selection techniques limited number of studies focus on subset-based feature selection. The primary contribution of this chapter is to provide the first study to evaluate subset-based feature selection (both filter-based subset selection and wrapper-based subset selection) in the context of data quality in bioinformatics. This chapter: (1) investigates the effectiveness of subset-based feature selection (both filter-based subset selection and wrapper-based subset selection) when learning from high dimensional datasets with varying data quality levels; (2) employs four classifiers that are commonly-used in the literature and (3) injects class noise into the data (after having been determined to be relatively free of noise) creating three learning data quality tiers (High-Quality, Average-Quality, and Low-Quality).

The remainder of this chapter will be organized as follows: Section 4.3 presents related works on the topics of dimensionality and data noise. Section 4.4 outlines the methods used in this work. In Section 4.5, we present our results. Finally, Section 4.6 concludes this chapter.

## 4.3 RELATED WORK

Datasets characterized by high dimensionality have a large number of features describing each instance, or sample (in most gene expression datasets the number of features exceeds the number of instances). This overabundance of features can worsen the performance of classification models and increase computation time, because usually, most of these features are useless for building a classification model. Feature selection is the most popular technique used to counter high dimensionality, which selects the most important features and removes irrelevant and redundant features. Reducing the number of features in a dataset can improve the classification performance of classifiers, reduce the complexity of classification models, and speed up the learning process.

Much research has been done on feature selection. A good survey on various aspects of the attribute selection problem was done by Guyon and Elisseeff [57]. The authors divide feature selection techniques into two broad categories: wrapper-based techniques and filter-based techniques. Wrapper-based techniques evaluate subsets of features using a classifier. This classifier is usually the same one which will be used for building the final model. On the other hand, filter-based techniques use different statistical measures to determine which features have the highest correlation with the class rather than using a classifier. Another comprehensive survey of feature selection techniques in bioinformatics can be found in the work of Saeys et al. [94]. Due to the computational complexity required by subset-based techniques, most research work focuses on feature ranking techniques, especially in bioinformatics. We provide, to our knowledge, the first assessment of the effectiveness of subset-based feature selection in the context of data quality.

Noise is another prevalent challenge exhibited by gene expression datasets. Noise refers to incorrect values in the data, which can be divided into two types: attribute noise and class noise. Attribute noise occurs when values in the independent attributes are incorrect (for example, gene expression levels not recorded correctly), while class noise refers to incorrect values in the dependent attribute (for example, cancerous instances incorrectly classified as noncancerous). Zhu and Wu [117] examined these two types of noise and concluded that class noise has a more harmful effect on classification performance than attribute noise. Frénay and Verleysen [51] performed a comprehensive survey on class noise. They concluded that many open research questions related to class noise remain to be explored.

Others have examined the impact of noise on the stability/robustness of feature selection through the direct injection of artificial noise. Wald et al. [109] examined six feature rankers, and their chosen feature lists were compared both between clean and noise-injected data and among the multiple runs of noise injection. They showed that ReliefF was a particularly stable ranker, and that comparing either noisy vs. clean or noisy vs. noisy gave similar results in terms of which rankers performed best. Abu Shanab et al. [2] evaluated six commonly used feature rankers, and their chosen feature lists were compared under different data perturbation (noise injection, sampling, and noise injection followed by sampling) and for different feature subset sizes. They showed that although stability is an important evaluation criterion for feature ranking techniques, a feature ranker's stability is not an indicator of its performance in classification.

Although previous work has evaluated the challenges of high dimensionality and noise injection, no previous work evaluated subset-based feature selection in the context of data quality using high dimensional gene expression datasets. Wald et al. [112] explores the degree to which class imbalance (unequal distribution of instances between classes) and difficulty of learning (due to class noise) affect one another and the best choices of learner and feature selection. Another study, Dittman et al. [42] investigated the effects of dataset difficulty due to noise injection on the stability of feature selection. The authors showed that in general, as the dataset difficulty increases, the stability of the generated feature subsets decreases.

## 4.4 METHODOLOGY

In this chapter, two forms of subset-based feature selection were used: filter-based subset techniques (CFS and Consistency) and wrapper-based subset selection using the Naïve Bayes learner ("WrapNB"). These techniques are discussed in greater detail in Section 2.4.2 and 2.4.3, respectively. Additionally, we used four classifiers (discussed in Section 2.7): Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), and Support Vector Machines (SVM). All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively. For all experiments in this chapter, we used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8.

## 4.5 EXPERIMENTAL RESULTS

In this work, we assess the effectiveness of subset-based feature selection in the context of data quality. We compare no feature selection with the use of CFS, Consistency, and wrapper-based feature selection using the Naïve Bayes learner and the AUC performance metric inside the wrapper process. Then, final (external) classification models are built with the selected features using four different learners (NB, MLP, 5NN, and SVM) which were then evaluated using the same metric (i.e. AUC). All experiments were performed on bioinformatics data which was first determined to be noise-free, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). We used the Greedy Stepwise approach as the search algorithm, and to avoid any validity problems related to overfitting we used four runs of five-fold cross-validation to build and test our models, presenting the average values across all folds and runs. The results are presented in Table 4.1. Within each column, **bold** values represent the best AUC value, and *italics* values represent the worst value.

Looking at the results, we see that CFS consistently outperforms the other subset-based feature selection across all data quality levels and learners. CFS was as much as 14% better than Consistency feature selection, and as much as 18% better than wrapper feature selection. Additionally, we found that CFS technique is able to improve the classification performance for all learners. Improving the classification performance on Average-Quality datasets to the level of High-Quality datasets. By the same token, CFS was able to improve the classification performance on Low-Quality datasets for all learners except the 5-NN learner, in the case of this exception the average AUC was 0.690596 (i.e. close to being Average-Quality). On the other hand, Wrapper subset selection and Consistency failed to significantly improve the classification performance on both Average-Quality and Low-Quality datasets for all learners, where the average AUC remained below 0.7 (i.e. Low-Quality) for Low-Quality datasets, and below 0.8 (i.e. Average-Quality) for Average-Quality datasets.

Additionally, we find that whether or not subset-based feature selection will improve performance compared to the no-selection (i.e. No FS) case will depend on the choice of learner: Using wrapper-based subset selection, for the MLP learner, the performance with the selected features was much better than the performance using all features, while the NB learner showed only slightly better performance with all features than with the wrapper-based features. However, for the remaining learners, wrapper-based feature selection led to a significant decrease in classification performance, and for these we would not recommend this form of feature selection. When using CFS or Consistency, both the NB and MLP learners showed better performance compared to the no-selection regardless of the data quality level, while the 5-NN learner showed only slightly better performance with all features than with the CFS features (except for High-Quality data with CFS). However, for the SVM learner, both CFS and Consistency led to a significant decrease in classification performance, and for this we would not recommend this form of feature selection.

Nonetheless, subset-based feature selection is useful beyond improving classification performance: it can help reveal which features are most important, giving a much smaller list compared to the full feature list and eliminating redundant features which might clutter up the list. For our datasets, subset-based feature selection chose fewer than 100 features, compared to the full feature sets which contained between 6,001 and 15,155 features. Even though subset-based feature selection was not always useful for improving classification performance, we still recommend that it be considered, especially CFS, because it is not obvious in advance whether a given choice of learner will benefit or suffer from the use of subset-based feature selection.

| Learner | Data Qyality<br>High-Quality | | | | Average-Quality | | | | Low-Quality | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | No FS | WrapNB | CFS | Consistency | No FS | WrapNB | CFS | Consistency | No FS | WrapNB | CFS | Consistency |
| NB | 0.876450 | *0.867986* | 0.949556 | 0.895332 | 0.755179 | *0.721312* | 0.851431 | 0.781433 | 0.630153 | 0.621130 | 0.725770 | 0.659367 |
| MLP | *0.784365* | **0.894154** | 0.960303 | **0.898343** | *0.667342* | **0.778169** | **0.862326** | **0.787789** | *0.599262* | **0.655581** | **0.745163** | **0.670370** |
| 5-NN | 0.947170 | 0.873441 | **0.966507** | *0.879339* | 0.868226 | 0.746842 | 0.856435 | *0.748018* | 0.728286 | *0.613167* | *0.690596* | *0.610049* |
| SVM | **0.962189** | 0.892210 | *0.940327* | 0.892767 | **0.901720** | 0.770426 | *0.829493* | 0.776205 | **0.763046** | 0.633657 | 0.727546 | 0.655261 |

**Table 4.1**: Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality)

## 4.6 CHAPTER SUMMARY

To the best of our knowledge this is the first study to evaluate subset-based feature selection when learning from high dimensional bioinformatics datasets in the context of data quality. We injected 24 different combinations of noise level and noise distribution into ten gene expression datasets which were first determined to be noise-free creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). We then performed feature selection using either wrapper feature selection (using the Naïve Bayes learner internally), CFS, or Consistency. This step was followed by external classification using one of four learners (Na¨ve Bayes, Multilayer Perceptron, 5-Nearest Neighbor, Support Vector Machines, and Logistic Regression). We used the area under the ROC curve (AUC) performance metric within the wrapper and to evaluate the final classification models due to the presence of imbalanced data.

Our experimental results demonstrate that the effectiveness of subset-based feature selection when learning from noisy high dimensional datasets depends on the choice of learner: with Multilayer Perceptron, subset-based feature selection is better than no feature selection, with Naïve Bayes learner we find that CFS and Consistency are better than no feature selection, while wrapper-based feature selection is slightly worse than no feature selection, and with the remaining two learners subset-based feature selection is worse than no feature selection (with one exception). However, because it may not be clear in advance whether subset-based feature selection will improve or worsen a given choice of learner, and due to the other benefits gained when employing subset-based feature selection (such as finding the most important features which are unique and not highly correlated with other features), we recommend exploring the use of subset-based feature selection (especially CFS) for bioinformatics, specifically if feature reduction (and elimination of redundant features) is more important than raw classification performance.

# CHAPTER 5

# COMPARING FEATURE RANKING, FILTER-BASED FEATURE SUBSET SELECTION, AND WRAPPER-BASED FEATURE SUBSET SELECTION FOR CLASSIFICATION OF NOISY BIOINFORMATICS DATA

## 5.1 INTRODUCTION

In Chapters 3 and 4 we utilized the three major types of feature selection (i.e. ranker-based techniques, filter-based subset selection, and wrapper-based subset selection) separately. We investigated their effectiveness in the context of learning from high-dimensional bioinformatics datasets with varying level of data quality due to noise injection. In this chapter [8], we investigate their effectiveness side-by-side. We perform an extensive study on the three types of feature selection in the context of learning from bioinformatics (gene microarray) datasets with varying levels of data quality (High-Quality, Average-Quality, and Low-Quality) due to noise injection. We evaluate 11 different feature selection strategies: three feature rankers each coupled with three feature subset sizes, a filter-based subset evaluator, and wrapper-based feature selection using Naïve Bayes inside the wrapper. We also consider the performance of all feature subsets using six different choices of classification learner. Statistical analysis, including the ANalysis Of VAriance (ANOVA) and Tukey's Honestly Significant Difference (HSD) criterion, is used within each grouping of data quality level to validate our results. We also performed a set of two-factor ANOVA tests to investigate the statistical significance of the choice of feature selection approach for each combination of learner and data quality.

## 5.2 CONTRIBUTIONS

The primary contribution of this chapter is to provide guidelines on best practices in dealing with high dimensionality in the presence of noise. We provide a thorough analysis of three major forms of feature selection when learning from datasets with varying levels of data quality due to noise injection. This chapter: (1) investigates the effectiveness of the three major types of feature selection: ranker-based techniques, filter-based subset selection, and wrapper-based subset selection; (2) simulates real-world datasets by injecting class noise into ten real-world gene-expression datasets (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality) and (3) employs six classification algorithms that are commonly-used in the literature.

The remainder of this chapter will be organized as follows: Section 5.3 presents related works on the topics of feature selection, and data noise. Section 5.4 outlines the methods used in this work. In Section 5.5, we present our results. Finally, Section 5.6 concludes this chapter.

## 5.3 RELATED WORK

High dimensionality refers to the large abundance of features, most often exceeding the number of instances in a dataset [99]. In bioinformatics datasets, this is especially unfortunate because most of these features provide little or no information for building a classification model. Feature selection is a common technique used to alleviate high dimensionality, by selecting a subset of the original set of features to be used in the learning process. Feature selection techniques can be divided into three major forms: ranker-based techniques, filter-based subset selection, and wrapper-based feature selection. Ranker-based techniques evaluate each feature individually using different statistical methods. Filter-based subset selection techniques also use statistical methods alone, however they evaluate multiple features (subsets) at a time rather than individual features. Finally, wrapper-based techniques use a classifier to directly find the subset of features which performs best. Due to its importance, feature selection has received a lot attention in the past few years. A comprehensive study on the concepts and algorithms of feature selection can be found in the work of Liu and Yu [78]. Sayes et al. [93] studied the use of ensemble feature selection methods and showed that the ensemble approach provides more robust feature subsets than a single feature selection method.

In bioinformatics, the problem of high dimensionality is more challenging because most gene expression datasets have thousands (sometimes tens of thousands) of genes and a much smaller sample size. A comprehensive survey on the concepts and algorithms of cancer classification using gene expression data can be found in the work of Lu and Han [80]. This paper compares cancer classification techniques in addition to various proposed gene selection techniques. The authors compared individual gene ranking and gene subset ranking and argued that, although gene subset ranking techniques are computationally more expensive, they are favored when analyzing gene expression datasets due to their ability to discover important information through the interactions among genes, and reveal complementary roles among genes helping in class distinction. Dittman et al. [43] investigated the best classifier for patient response prediction to a drug treatment and the degree to which feature selection can help increase the accuracy of learners. The authors showed that if gene selection occurs to a specific degree (selecting around 200 to 1000 genes) then the classifier will achieve excellent classification results when compared to previous studies using the same data. Wang and Gotoh [113] applied two gene selection techniques to reduce the number of genes used in microarray-based classifiers to a small number (single and double genes). The result showed that classifiers that include few genes can perform well in practice, and are advantageous in terms of practical implementation and interpretation. Abeel et al. [1] studied the process for selecting biomarkers from microarray data and presented a general framework for stability analysis of such feature selection techniques.

Due to the computational complexity required by wrapper-based techniques and the chance of building an overfitted inductive model, only few studies focus on wrapper-based techniques, especially in bioinformatics. Xiong et al. [116] consider wrapper feature selection, using three learners and three datasets. They found that selecting more than one top feature was able to improve performance over using just one feature, and that a more advanced search technique capable of backtracking showed greater performance than simple forward selection. A study performed by Inza et al. [62] compared filter-based feature ranking with wrapper-based subset selection, using two bioinformatics datasets and six feature ranking techniques along with four choices of learner. The results show that wrapper feature selection outperforms filter-based ranking, but at a high computational cost. Wang et al. [114] compare all three forms of feature selection, using four filter-based rankers, one filter-based subset evaluator, and three classifiers for both wrapper selection and final classification. Results are evaluated using two gene microarray datasets, and the authors find that on the first dataset, all three techniques are very consistent in terms of one gene found to have extremely high connection to the class in question; more varied results are found on the second dataset. Nonetheless, they find that the filter- and wrapper-based subset selection approaches can give good performance while selecting a smaller subset of features.

Another major challenge in bioinformatics is noise, which occurs as a result of faulty microarray chips, insufficient resolution, image corruption, and other reasons. There are two types of noise: attribute noise and class noise. Attribute noise occurs when values in the independent attributes are incorrect (for example, gene expression levels not recorded correctly), while class noise refers to incorrect values in the dependent attribute (for example, cancerous instances labeled as noncancerous). Zhu and Wu [117] examined these two types of noise and concluded that class noise has a more harmful effect on classification performance than attribute noise. Wald et al. [109] examined the stability of six feature rankers by comparing their chosen feature lists between clean and noise-injected data and among the multiple runs of noise injection. They showed that ReliefF was a particularly stable ranker, and that comparing either noisy vs. clean or noisy vs. noisy gave similar results in terms of which rankers performed best. Unfortunately, many data mining techniques are sensitive to data noise, thus, low quality data can result in suboptimal predictive classification performance and can also impact the effectiveness of feature selection. Therefore, it is important to understand how low quality data can impact data mining techniques (feature selection techniques and classification models).

Since gene expression data exhibit both problems (high dimensionality and noise) simultaneously, it is important to understand how different feature selection types perform under different data quality levels (i.e. different levels and distribution of noise). This study is the first to compare three major forms of feature selection when learning from bioinformatics datasets with varying levels of data quality due to noise injection. To assess this effectiveness, we compare three forms of feature ranking (with three choices of feature subset size for each), one form of filter-based subset evaluation, and wrapper subset selection. We perform experiments using ten gene expression datasets which were first determined to be free of noise, and then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality), and we build our final models using six different classification algorithms.

## 5.4 METHODOLOGY

In this chapter, we utilized three feature rankers each coupled with three feature subset sizes (i.e. 25, 50, and 100), a filter-based subset evaluator, and wrapper-based feature selection using Naïve Bayes inside the wrapper. For feature ranking, we choose three representative techniques: Chi Squared (CS), Area Under the Receiver Operating Characteristic (ROC) Curve, and Wilcoxon Rank Sum (WRS). For filter-based subset feature selection, we choose Correlation-Based Feature Selection (CFS) [58]. This technique is the most commonly-used form of filter-based subset evaluation in the literature, and the study in Chapter 4 suggested that it performed much better than alternatives such as Consistency [36]. On the other hand, for wrapper-based subset selection we use the Naïve Bayes (discussed further in Section 2.7) classifier to build our models and the AUC (discussed further in Section 2.8) performance metric as the performance metric within the wrapper. All feature selection techniques are discussed in more detail in Section 2.4.

Additionally, six classifiers were used (discussed in Section 2.7): Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively. For all experiments in this chapter, we used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8.

## 5.5 EXPERIMENTAL RESULTS

The primary objective of this work is to investigate the effectiveness of three major types of feature selection (ranker-based techniques, filter-based subset selection, and wrapper-based subset selection) when learning from high dimensional bioinformatics datasets with varying levels of data quality due to noise injection. The results for all combinations of feature selection approach, learner, and data quality level are presented in Table 5.1. Each cell contains the average AUC performance across four runs of five-fold cross-validation when applying the given choice of feature selection and learner to the dataset or datasets which match that data quality level. In the "Feature Selection Approach" column, the rankers (CS, ROC, and WRS) are followed by a number, which represents the number of features chosen from that ranked list, and the wrapper-based selection approach which uses the NB learner inside the wrapper is abbreviated as "WrapNB" for space considerations. Within each block of data quality level, the best and worst choices of feature selection approach for each learner are printed in **bold** and *italics*, respectively.

The results demonstrate that feature rankers outperform the two subset-based approaches for all combinations of learner and data quality level (except High-Quality data with the LR learner). Feature rankers were also robust in not being the worst choice: for only 3 of the 15 combinations (LR learner combined with three data quality levels) was a ranker at the bottom of the pack. In the case of these exceptions the findings are not significant as LR is the worst performing learner across all data quality levels and feature selection techniques and we recommend against the use of LR for classification. Thus, based on these results and the fact that feature rankers are computationally much less expensive than the two subset-based approaches we recommend using feature rankers to reduce high dimensionality when analyzing bioinformatics datasets regardless of the data quality level (i.e. noise level).

Furthermore, wrapper-based subset selection was the worst performing for 12 of

| Data Quality Level | Feature Selection Approach | NB | MLP | 5-NN | SVM | RF100 | LR |
|---|---|---|---|---|---|---|---|
| High Quality | CS25 | 0.961932 | 0.952085 | 0.959592 | 0.945954 | 0.971598 | 0.856493 |
| | CS50 | 0.961620 | 0.955400 | 0.965779 | 0.937508 | 0.977615 | 0.822673 |
| | CS100 | 0.958673 | 0.960675 | 0.970540 | 0.936207 | 0.981070 | *0.814190* |
| | ROC25 | **0.968102** | 0.962383 | 0.961666 | **0.959196** | 0.975240 | 0.877569 |
| | ROC50 | 0.965055 | 0.963395 | 0.967567 | 0.948559 | 0.979554 | 0.850008 |
| | ROC100 | 0.960835 | **0.965029** | **0.972347** | 0.942010 | **0.982486** | 0.835051 |
| | WRS25 | 0.967041 | 0.962003 | 0.961731 | 0.958641 | 0.975268 | 0.876748 |
| | WRS50 | 0.963858 | 0.963150 | 0.967223 | 0.948243 | 0.979669 | 0.849693 |
| | WRS100 | 0.959325 | 0.964956 | 0.972167 | 0.942467 | 0.982147 | 0.834355 |
| | CFS | 0.949556 | 0.960303 | 0.966507 | 0.940327 | 0.982384 | 0.864865 |
| | WrapNB | *0.867986* | *0.894154* | *0.873441* | *0.892210* | *0.897436* | **0.880030** |
| Average Quality | CS25 | 0.892423 | 0.867767 | 0.859956 | 0.854283 | 0.892008 | 0.758766 |
| | CS50 | 0.896603 | 0.866512 | 0.869389 | 0.835792 | 0.905384 | 0.724907 |
| | CS100 | 0.892595 | 0.870165 | 0.876304 | 0.827590 | 0.916087 | *0.706785* |
| | ROC25 | **0.901329** | 0.874765 | 0.868771 | **0.866969** | 0.900651 | **0.767354** |
| | ROC50 | 0.900525 | 0.874682 | 0.880330 | 0.847837 | 0.913703 | 0.736948 |
| | ROC100 | 0.893658 | 0.876511 | 0.887368 | 0.842273 | 0.919216 | 0.721042 |
| | WRS25 | 0.898082 | 0.875872 | 0.868032 | 0.865305 | 0.901025 | 0.766554 |
| | WRS50 | 0.896780 | 0.877145 | 0.880282 | 0.849376 | 0.912744 | 0.739557 |
| | WRS100 | 0.887536 | **0.878591** | **0.888060** | 0.843384 | **0.919812** | 0.722153 |
| | CFS | 0.851431 | 0.862326 | 0.856435 | 0.829493 | 0.907082 | 0.734070 |
| | WrapNB | *0.721312* | *0.778169* | *0.746842* | *0.770426* | *0.782741* | 0.753664 |
| Low Quality | CS25 | 0.739903 | 0.742270 | 0.699248 | 0.728574 | 0.748440 | 0.658942 |
| | CS50 | 0.739079 | 0.756490 | 0.707660 | 0.728406 | 0.770173 | 0.643609 |
| | CS100 | 0.729980 | 0.758988 | 0.713110 | 0.722922 | 0.781706 | *0.627673* |
| | ROC25 | 0.767019 | 0.766209 | 0.717181 | 0.753189 | 0.773851 | 0.679400 |
| | ROC50 | **0.768437** | 0.765599 | 0.733898 | 0.745796 | 0.787752 | 0.655154 |
| | ROC100 | 0.760696 | 0.767606 | **0.741238** | 0.738545 | 0.798992 | 0.635573 |
| | WRS25 | 0.759576 | 0.769038 | 0.717921 | **0.754169** | 0.769752 | **0.680397** |
| | WRS50 | 0.759713 | **0.771375** | 0.728124 | 0.748108 | 0.787479 | 0.654214 |
| | WRS100 | 0.748870 | 0.771142 | 0.739807 | 0.743059 | **0.800665** | 0.632017 |
| | CFS | 0.725770 | 0.745163 | 0.690596 | 0.727546 | 0.759900 | 0.649234 |
| | WrapNB | *0.621130* | *0.655581* | *0.613167* | *0.633657* | *0.630219* | 0.637428 |

**Table 5.1**: Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality)

| Data Quality Level | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| High-Quality | FS Technique | 50.76 | 10 | 5.07566 | 517.29 | 0 |
| | Error | 1605.92 | 163669 | 0.00981 | | |
| | Total | 1656.68 | 163679 | | | |
| Average-Quality | FS Technique | 54.21 | 10 | 5.42071 | 262.23 | 0 |
| | Error | 1391.37 | 67309 | 0.02067 | | |
| | Total | 1445.58 | 67319 | | | |
| Low-Quality | FS Technique | 38.7 | 10 | 3.8697 | 116.41 | 0 |
| | Error | 1359.86 | 40909 | 0.03324 | | |
| | Total | 1398.56 | 40919 | | | |

**Table 5.2**: ANOVA Results: Feature Selection Techniques Across All Learners

the 15 combinations and was only the top performing for one combination (High-Quality data with the LR learner), while CFS was neither the best choice nor the worst choice for any combination and it was always outperformed by a ranker. For these reasons and the fact that subset-based selection is computationally more expensive compared to rankers, we recommend against using subset-based selection (mainly wrapper-based subset selection), especially if eliminating redundancy among the selected features is not a priority.

Looking at these results on a per-data-quality-level basis, it can be seen that best choice of feature selection can vary depending on the data quality level, nevertheless the choice is always a feature ranker. For High-Quality datasets, ROC was consistently the best performing feature selection technique (except for the LR learner), and the best performance across all learners was obtained when ROC was utilized with the top 100 features using the RF100 learner (0.982486). For Average-Quality datasets, ROC and WRS were the top performing rankers (alternating between first and second place equal number of times), and the best performance across all learners was obtained when WRS was utilized with the top 100 features using the RF100 learner (0.919812). On the other hand, for Low-Quality datasets, although ROC and WRS were consistently at the top of the pack across all learners, WRS was more

| Classifier | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| NB | FS Technique | 20.374 | 10 | 2.0374 | 364.34 | 0 |
| | Error | 152.488 | 27269 | 0.00559 | | |
| | Total | 172.862 | 27279 | | | |
| MLP | FS Technique | 10.444 | 10 | 1.04437 | 183.21 | 0 |
| | Error | 155.447 | 27269 | 0.0057 | | |
| | Total | 165.891 | 27279 | | | |
| 5-NN | FS Technique | 19.972 | 10 | 1.99722 | 374.81 | 0 |
| | Error | 145.305 | 27269 | 0.00533 | | |
| | Total | 165.278 | 27279 | | | |
| SVM | FS Technique | 19.972 | 10 | 1.99722 | 374.81 | 0 |
| | Error | 145.305 | 27269 | 0.00533 | | |
| | Total | 165.278 | 27279 | | | |
| RF100 | FS Technique | 15.194 | 10 | 1.51936 | 436.51 | 0 |
| | Error | 94.914 | 27269 | 0.00348 | | |
| | Total | 110.108 | 27279 | | | |
| LR | FS Technique | 12.711 | 10 | 1.27112 | 61.9 | 0 |
| | Error | 560.003 | 27269 | 0.02054 | | |
| | Total | 572.714 | 27279 | | | |

**Table 5.3**: ANOVA Results: Feature Selection Techniques For Each Classifier (High-Quality)

frequently the top performing approach (for 4 of the 6 combinations), and the best performance across all learners was obtained when WRS was utilized with the top 100 features using the RF100 learner (0.800665). This gives us confidence that ROC and WRS are good choices for feature selection when learning from bioinformatics datasets regardless of the data quality level (i.e. noise level).

When we look at the results in terms of the subset size, we see some interesting trends. The first (applies to 5-NN and RF100) is that, as the subset size increases, the performance increases, regardless of the data quality level and feature ranker. The second trend (applies to SVM and LR) is that, as the subset size increases, the performance decreases, regardless of the data quality level and feature ranker. On the other hand, there was no clear pattern among the other combinations for how the subset size and data quality level affect the AUC value. Both MLP and NB had some internal optimum, a value of subset size which maximizes the performance (Average-Quality data with the CS ranker and NB learner; Low-Quality data with the ROC ranker and NB learner; and Low-Quality data with the WRS ranker and MLP or NB learners) or a dip in the middle (Average-Quality data with the MLP learner and ROC or CS rankers). Overall, the results for (5-NN, RF100, SVM, and LR) suggest that the optimal feature subset size for these learners is related to the learner rather than the choice of ranker or data quality level, while for both NB and MLP the results demonstrate that it is difficult to predict the optimal feature subset size based on the feature ranker and/or the data quality level, and in practice this must be evaluated on a case-by-case basis.

Furthermore, we see that CS, ROC, WRS, and CFS are particularly robust and are able to improve the classification performance for all learners (except LR) when Average-Quality datasets are used. This is shown by the improved performance of classifiers (NB, MLP, 5-NN, SVM, and RF100) when learning from Average-Quality datasets resulting in AUC values greater than 0.8 (i.e. High-Quality datasets). There was no ranker that was able to improve the classification performance of Average-Quality datasets for LR, where the average AUC remained below 0.8 (i.e. Average-Quality). By the same token, no ranker was able to improve the classification performance of Low-Quality datasets for LR, where the average AUC remained below 0.7 (i.e. Low-Quality). It is of note that WRS (when utilized with the top 100 features) improved the classification performance of the Random Forest 100 learner resulting in an AUC greater than 0.8 (i.e. High-Quality), this is a significant improvement based on the fact that initially AUC was less than 0.7 (i.e. Low-Quality)

To find statistically significant patterns in the results we performed a set of one-factor ANalysis Of VAriance (ANOVA) tests [23]. We used Matlab to perform the ANOVA and subsequent statistical tests. The results are presented in Table 5.2. Since a significance factor of 5% was chosen, the $Prob > F$ value must be less than

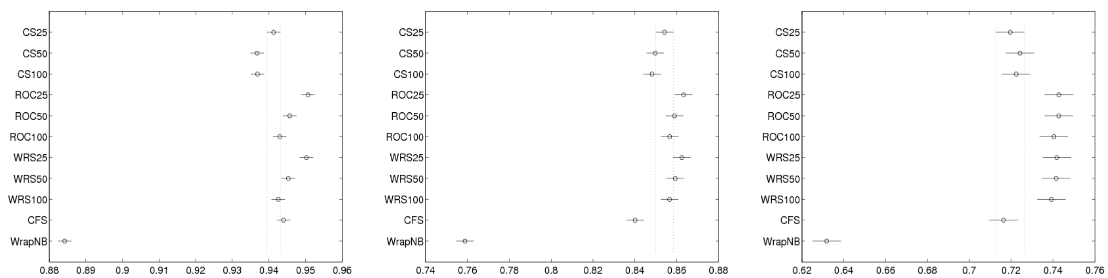
| Classifier | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| NB | FS Technique | 28.67 | 10 | 2.86698 | 191.73 | 0 |
| | Error | 167.611 | 11209 | 0.01495 | | |
| | Total | 196.281 | 11219 | | | |
| MLP | FS Technique | 8.508 | 10 | 0.8508 | 51.7 | 0 |
| | Error | 184.478 | 11209 | 0.01646 | | |
| | Total | 192.986 | 11219 | | | |
| 5-NN | FS Technique | 15.944 | 10 | 1.59435 | 96.03 | 0 |
| | Error | 186.1 | 11209 | 0.0166 | | |
| | Total | 202.043 | 11219 | | | |
| SVM | FS Technique | 6.992 | 10 | 0.69925 | 37.9 | 0 |
| | Error | 206.8 | 11209 | 0.01845 | | |
| | Total | 213.792 | 11219 | | | |
| RF100 | FS Technique | 15.489 | 10 | 1.54892 | 121.83 | 0 |
| | Error | 142.514 | 11209 | 0.01271 | | |
| | Total | 158.003 | 11219 | | | |
| LR | FS Technique | 4.121 | 10 | 0.41205 | 15.19 | 0 |
| | Error | 304.147 | 11209 | 0.02713 | | |
| | Total | 308.267 | 11219 | | | |

**Table 5.4**: ANOVA Results: Feature Selection Techniques For Each Classifier (Average-Quality)

this value (i.e. 0.05) for the result to be significant.

In this analysis, we considered only one factor: the choice of feature selection approach, with 11 different levels of this factor (three feature rankers each coupled with three feature subset sizes, CFS, and wrapper using NB inside the wrapper). Each test was performed on one collection of datasets (that is, one level of data quality). For the ANOVA tests, the AUC results across all six learners were used as the response variable. In addition, the separate results from all four runs of five-fold cross-validation were used separately. Based on these results, we can conclude that the one factor (i.e. choice of feature selection approach) investigated is significant; that is to say, when the data are grouped by the choice of feature selection approach, at least two of those groups will have significantly different means.

We wanted to find out which pairs of means are significantly different, and which are not. We conducted a multiple pairwise comparison by using Tukey's Honestly Significant Difference (HSD) criterion [23]. The significance level for Tukey's HSD test is $\alpha = 0.05$. Figure 5.5 shows the comparison results of the 11 feature selection techniques for the different levels of data quality; the one on the left shows the results for High-Quality datasets, while the one in the middle, those for Average-Quality datasets, and the one on the right shows the results for Low-Quality datasets. The figures display graphs within each group mean represented by a symbol ($\circ$) and the 95% confidence interval as a line around the symbol. Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap.




(a) High-Quality (b) Average-Quality (c) Low-Quality

**Figure 5.1**: Tukey's HSD Results: Feature Selection Techniques Across All Learners

Figure 5.5 supports our conclusion that the top performing choice of feature selection strategy is always a ranker-based approach. The difference between the top performing ranker-based approach and the other feature selection types (i.e. CFS and Wrapper) is statistically significant across all data quality levels. Furthermore, wrapper-based subset selection was significantly the worst performing approach across all data quality levels. On the other hand, CFS results are less clear. For High-Quality datasets, CFS was competitive with the top feature ranking techniques. When considering Average-Quality and Low-Quality datasets, CFS was significantly outperformed by ROC and WRS regardless of the subset size.

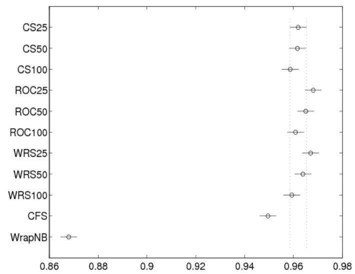
| Classifier | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| NB | FS Technique | 10.674 | 10 | 1.06736 | 32.98 | 0 |
| | Error | 220.336 | 6809 | 0.03236 | | |
| | Total | 231.009 | 6819 | | | |
| MLP | FS Technique | 6.925 | 10 | 0.69247 | 22.68 | 0 |
| | Error | 207.9 | 6809 | 0.03053 | | |
| | Total | 214.825 | 6819 | | | |
| 5-NN | FS Technique | 7.909 | 10 | 0.79088 | 22.25 | 0 |
| | Error | 241.975 | 6809 | 0.03554 | | |
| | Total | 249.884 | 6819 | | | |
| SVM | FS Technique | 6.995 | 10 | 0.6995 | 22.38 | 0 |
| | Error | 212.787 | 6809 | 0.03125 | | |
| | Total | 219.782 | 6819 | | | |
| RF100 | FS Technique | 13.838 | 10 | 1.38384 | 48.76 | 0 |
| | Error | 193.261 | 6809 | 0.02838 | | |
| | Total | 207.1 | 6819 | | | |
| LR | FS Technique | 1.947 | 10 | 0.19474 | 6.1 | 2.64E-09 |
| | Error | 217.523 | 6809 | 0.03195 | | |
| | Total | 219.47 | 6819 | | | |

**Table 5.5**: ANOVA Results: Feature Selection Techniques For Each Classifier (Low-Quality)

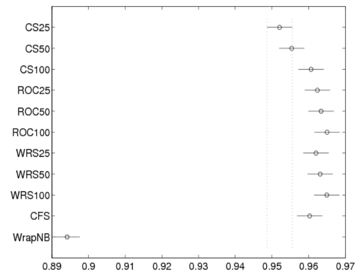
Among the rankers, the results confirm that ROC and WRS are robust and less sensitive to class noise. They are both very good, and the best performing technique among them depends on the level of data quality, but is not statistically significant. CS, on the other hand, is more sensitive to class noise and was always the worst performing feature ranker regardless of the data quality level. The results in terms of subset size show that both ROC and WRS share the decreasing pattern, where the best results are found with size 25 and performance decreases as the number of features grows regardless of the data quality level. However, the difference between all subset sizes is statistically insignificant when using ROC and WRS with Average-Quality and Low-Quality datasets. As for CS, the following three patterns can be observed (decreasing, internal maximum, and internal minimum), although this finding is not significant due to the fact that CS is the worst performing feature ranker and is more sensitive to class noise.

We also performed another series of one-factor ANOVA tests. The results are presented in Tables 5.3, 5.4, and 5.5. In this analysis, we considered the factor of the choice of feature selection approach for each combination of data quality and learner. Each table shows the results for a single data quality level. For the ANOVA tests, the AUC results across all datasets which match that data quality level were used as the response variable. In addition, the separate results from all four runs of five-fold cross-validation were used individually. Based on these results, we can conclude that for each combination of data quality level and learner, at least two different feature selection approaches will have significantly different means.

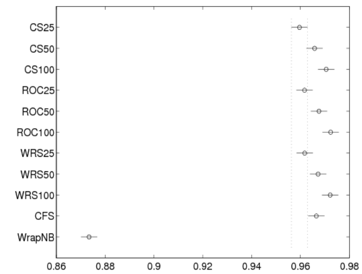
We performed a test of Tukey's HSD criterion to find out which pairs of means are significantly different. Figures 5.5, 5.6, and 5.6 show the comparison results of the 11 feature selection techniques for the different combinations of learner and data quality level. Each figure contains six subfigures: one for each classifier (NB, MLP, 5-NN, SVM, RF100, and LR, respectively). The results, generally speaking, support the results we saw with Table 5.1: as mentioned earlier, wrapper-based subset selection was statistically the worst performing approach for all combinations of learner and data quality level, except for the LR learner. However, it should be noted that LR is the worst performing learner for all scenarios. The results also show that although CFS was consistently outperformed by a feature ranker, the statistical significance in performance (between the top performing ranker and CFS) depends on the learner and data quality level. Thus, based on these results and the fact that CFS is computationally more expensive than rankers, we still recommend using feature rankers as the feature selection strategy when analyzing real-world bioinformatics datasets, regardless of the level of data quality (i.e. level of noise) and the learner being used



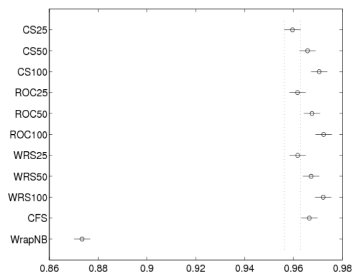

(a) NB



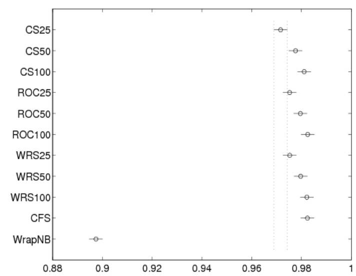

(b) MLP



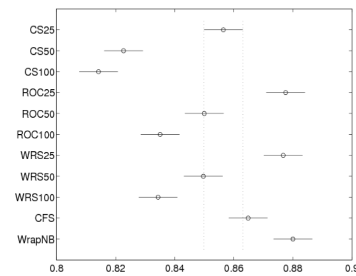

(c) 5-NN




(d) SVM




(e) RF100



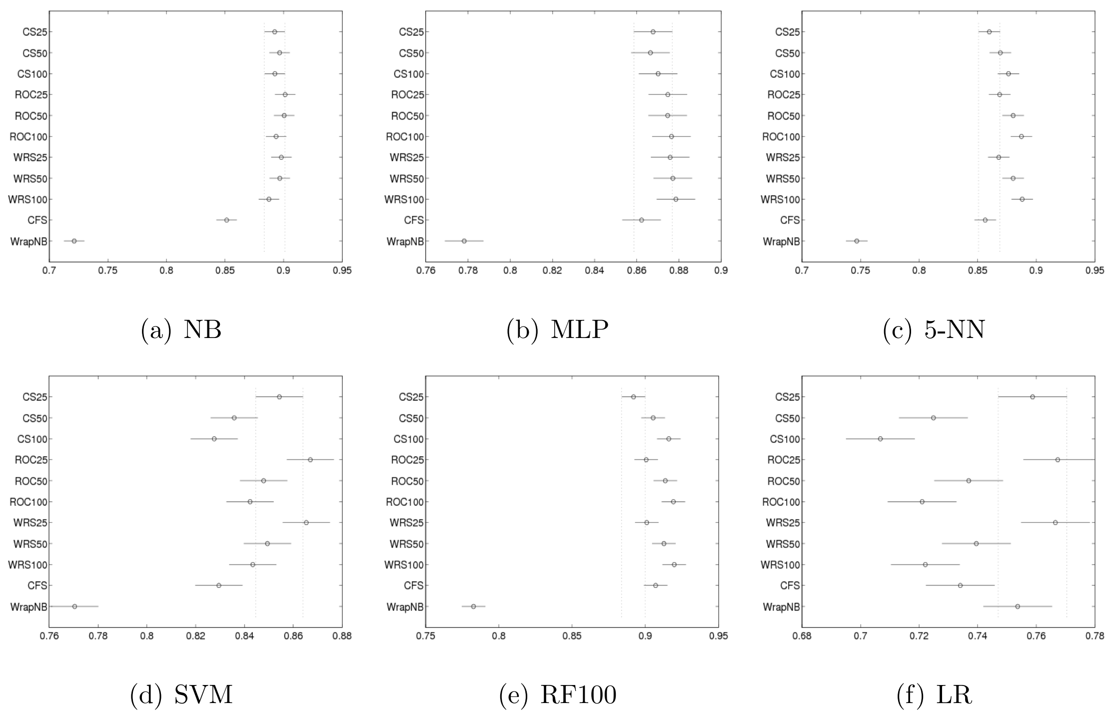

(f) LR

**Figure 5.2**: Tukey's HSD Results: Feature Selection Techniques For Each Classifier (High-Quality)

## 5.6 CHAPTER SUMMARY

To the best of our knowledge, this is the first study to compare three major categories of feature selection when learning from bioinformatics datasets with varying levels of data quality due to noise injection. To investigate their effectiveness, we compare three forms of feature ranking (with three choices of feature subset size for each), one form of filter-based subset evaluation, and wrapper subset selection. We create three levels of data quality (High-Quality, Average-Quality, and Low-Quality) by injecting artificial class noise in a controlled fashion into ten gene expression datasets which were first determined to be High-Quality datasets (i.e. these datasets are relatively clean and noise-free). We build our final models using six different classification algorithms.

The results demonstrate that feature rankers outperform the two subset-based approaches for all combinations of learner and data quality level (except High-Quality

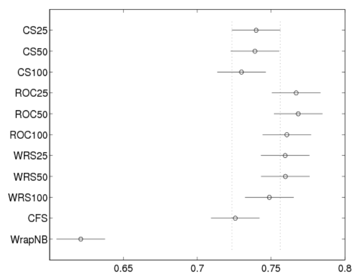



(a) NB (b) MLP (c) 5-NN

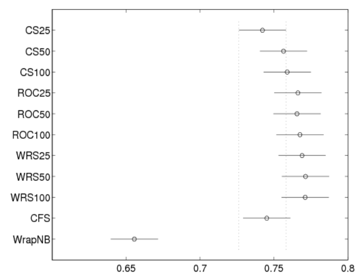
(d) SVM (e) RF100 (f) LR

**Figure 5.3**: Tukey's HSD Results: Feature Selection Techniques For Each Classifier (Average-Quality)

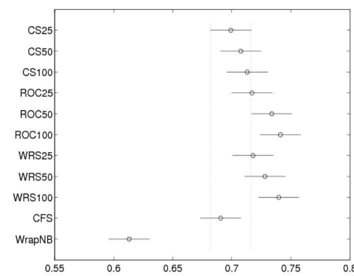
data with the Logistic Regression learner). Feature rankers were also robust in not being the worst choice: for only 3 of the 15 combinations (Logistic Regression learner combined with three data quality levels) was a ranker at the bottom of the pack. In the case of these exceptions the findings are not significant as Logistic Regression is the worst performing learner across all data quality levels and feature selection techniques and we recommend against the use of Logistic Regression for classification. Although CFS performance was competitive, it was always outperformed by a feature ranker. When considering Average-Quality and Low-Quality datasets, CFS was significantly outperformed by Area Under the ROC Curve and Wilcoxon Rank Sum. Wrapper-based subset selection, on the other hand, was significantly the worst performing approach across all data quality levels. All of this analysis was confirmed through ANOVA and Tukey's HSD tests. Thus, based on these results and the fact that feature rankers are computationally much less expensive than the two subset-based approaches, we recommend using feature rankers to reduce high dimensionality when analyzing bioinformatics datasets, regardless of the data quality level (i.e. noise level). In particular, the ROC and WRS rankers are less sensitive to class noise and are good choices for feature selection.



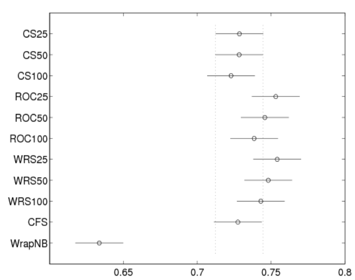

(a) NB



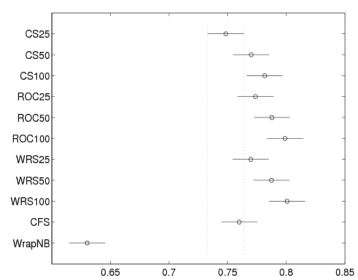

(b) MLP



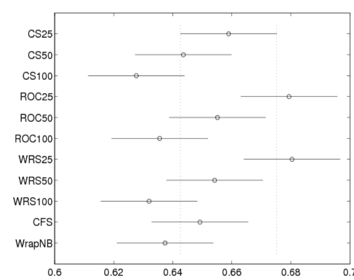

(c) 5-NN




(d) SVM




(e) RF100




(f) LR

**Figure 5.4**: Tukey's HSD Results: Feature Selection Techniques For Each Classifier (Low-Quality)

# CHAPTER 6

# ENSEMBLE CLASSIFICATION PERFORMANCE ON BALANCED BIOINFORMATICS DATA WITH VARYING LEVELS OF DATA QUALITY

## 6.1 INTRODUCTION

Much research in recent years has examined gene expression datasets, using data mining and machine learning tools to build classification models to accurately classify new instances using information gained from previous instances. Unfortunately, gene expression datasets are commonly characterized by two challenging properties: (1) high-dimensionality and (2) small sample size. Many real-world gene expression datasets exhibit high-dimensionality, which refers to the extremely large number of genes (sometimes tens of thousands) in a dataset. Only a select number of the genes are important for the underlying classification problem. The majority of those genes are redundant (containing information already represented in other genes) or irrelevant (having little or no correlation with the class) to the question at hand. Previous studies showed that feature selection (a commonly-used process to alleviate high-dimensionality) can improve the classification performance of inductive models, improve model interpretability, and speed up the learning process. On the other hand, the small sample of gene expression datasets (usually less than one hundred instances) increases the chance of building an over-fitted classification model which has poor generalization capability and fails drastically when classifying new/unseen instances. These challenging properties necessitate the use of advanced data mining techniques.

Ensemble classification is one of the promising methods for resolving the aforementioned problems, which combines multiple classification algorithms and makes the final classification using an aggregation technique. A number of studies have proposed and evaluated the performance of different ensemble techniques. Ensemble classification can help improve the prediction accuracy of the base classifier as well as reduce bias and over-fitting. Three of the most commonly-used ensemble algorithms are Bagging, Random Forest, and Boosting. To our knowledge, no previous work systematically investigated the effectiveness of ensemble techniques in the context of data quality.

Noise is a major data quality problem encountered when analyzing gene expression datasets, which occurs when the data contain erroneous values. Noise can be introduced as a result of experimental errors (e.g. faulty microarray chips, insufficient resolution, image corruption, and incorrect laboratory procedure), as well as other errors (errors during data processing, transfer, and/or mining). Regardless of its source, noise has a detrimental impact on data mining techniques (e.g. suboptimal classification performance and unstable feature selection). A special type of noise called class noise occurs when an instance/example is mislabeled. Previous research showed that class noise has a more harmful effect on classification problems than the other type of noise [117]. Thus, all experiments in this chapter were performed on data which was modified by injecting artificial class noise in a controlled fashion.

In this chapter [14], we investigate the effectiveness of ensemble classification techniques when learning from balanced bioinformatics datasets with varying levels of data quality. We compare three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest) as well as feature selection followed by classification using ten high-dimensional and balanced gene expression datasets which were first determined to be free of noise, and then had artificial class noise injected in a controlled fashion creating three levels of data quality. Select-Bagging and

Select-Boosting (proposed and developed by our data mining research team) implement feature selection within each iteration of their respective ensemble classification frameworks.

## 6.2 CONTRIBUTIONS

Ensemble classification can help improve the prediction accuracy of the base classifier as well as reduce bias and over-fitting, however, no previous work systematically investigated the effectiveness of ensemble techniques in the context of data quality. The primary contribution of this chapter is to provide the first comprehensive examination of ensemble classification techniques when learning from balanced bioinformatics datasets in the context of data quality. This chapter: (1) investigates the robustness of three forms of ensemble classification to class noise; (2) simulates real-world datasets by injecting class noise into ten real-world gene-expression datasets (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality) and (3) employs five classification algorithms that are commonly-used in the literature.

The remainder of this chapter is organized as follows: Section 6.3 presents related works to our topic. Section 6.4 introduces the three ensemble classification methods. Section 6.5 outlines the methodology for our experiments. In Section 6.6, we present our results. Finally, conclusions are presented in Section 6.7.

## 6.3 RELATED WORK

Gene expression datasets are commonly characterized by a number of challenging properties which make it necessary to utilize advanced data mining techniques. One of the prevalent problems exhibited by gene expression datasets is high-dimensionality, which refers to the large number of features (i.e. genes) in datasets. In most cases many of the genes are irrelevant or redundant to the problem being researched. Better classification performance and improved model interpretability can be achieved by selecting the most important genes. Additionally, the presence of noise in gene expression datasets is inevitable, which can adversely impact data mining techniques and subsequently lead to inaccurate classification. Furthermore, the small sample size of gene expression datasets can increase the risk of over-fitting.

The study of ensemble classifiers is currently one of the most active fields of research in bioinformatics. Ensemble classification is the process of generating multiple classification models and then aggregating their decisions into a single final decision. Ensemble classification confers a number of advantages, including: improved classification performance [39], reduced over-fitting [40], and a reduction of bias [95]. Bagging, Random Forest, and Boosting are the three most commonly used ensemble classification techniques. Bagging [28] is an ensemble technique that builds multiple classifiers using perturbed subsets of the original training dataset (using random sampling with replacement from the original dataset) and aggregates the results from these learners using majority voting. Random Forest [29] is another ensemble approach that combines the bagging approach and the random selection of genes to build a set of unpruned decision trees, then uses majority voting to perform prediction. Boosting [52] is a method to improve the performance of weak classifiers which operates by iteratively training weak classifiers while modifying the weight of each instance at each iteration (the weights of misclassified instances will increase while the weights of correctly classified instances will decrease).

Much research has been done on ensemble classification. A survey of ensemble classification for DNA microarray data is presented by Khoshgoftaar et al. [70], detailing the need for ensemble classification and discussing the widely-used ensemble techniques as well as novel approaches to ensemble classification. They concluded that many open research questions related to ensemble classification for DNA microarray datasets remain to be explored (e.g. the effect of class imbalance).

Peng [88] presented a new ensemble technique which focuses on building sets of classifiers based on different gene subsets. This method utilizes SVM as the base classifier and employs the k-means algorithm to distribute the classifiers into different clusters and then selects classifiers with the smallest misclassification rate from the clusters to construct the classification committee. The author compared the proposed SVM ensemble to two ensemble learning methods (i.e. bagging and boosting) as well as a single SVM classifier on five cancer datasets. The results showed that the proposed ensemble achieved the best results. One downside of this study, however, was in performing feature selection outside the ensemble techniques, and as a result the chosen features may not be as valid due to the dataset changes that occur during ensemble learning.

Tan and Gilbert [101] compared ensemble techniques to a single tree classifier on a series of cancer classification gene microarray datasets and showed that ensemble techniques are more robust and accurate compared to the single tree classifier. However, the authors failed to cross-validate all steps (e.g. dimensionality reduction technique) which may lead to substantial bias in the estimated error rate [98]. Additionally, the authors only used a single run of cross-validation which increases the potential bias due to a chance split. Lastly, the authors included a few datasets which are imbalanced with a minority class being as low as 10.07% instead of using only balanced datasets. By only using balanced datasets, a possible source of bias (the level of class imbalance of the data set) is removed from the experiment.

In 2008, Chen and Zhao [34] proposed a new ensemble technique which uses a set of artificial neural networks which are trained from different sampled instances from the original dataset. The authors showed that the method introduced in the paper had superior classification performance using four cancer datasets when compared to a number of other methods. One downside of this work however, was in the ad-hoc comparison being conducted. The authors compared the result of the new ensemble technique to the results from the same datasets by different researchers, and the actual methods used by those other researchers are not discussed whatsoever.

Noise, which refers to erroneous values in the data, is another major challenge in bioinformatics. There are generally two types of noise: attribute noise and class noise. Attribute noise occurs when the independent attributes/features of an instance contain either incorrect or missing values, while class noise occurs when the dependent variable of an instance is erroneous. Previous research found that class noise has a more detrimental effect on classification performance than attribute noise [117] and can also cause a feature ranker to produce unstable output [109]. Dietterich [40] explored the effect of noise on the classification performance of ensemble techniques using 33 datasets from different domains. The author concluded that Bagging is much better than Boosting in the presence of substantial noise. In another study, Jiang showed that Boosting can cause over-fitting in the presence of noise [63]. Although gene expression datasets are inherently noisy, no previous study, that we are aware of, tried to quantify this noise and understand how it can impact ensemble techniques.

## 6.4 ENSEMBLE CLASSIFICATION APPROACHES

Ensemble classification is an important topic in the bioinformatics domain, which can help improve classification performance as well as reduce over-fitting and bias. Ensemble classification combines multiple models and aggregates their results into a single decision to obtain better predictive performance. Bagging, Random Forest, and Boosting are most commonly used ensemble classification techniques.

Bagging developed by Breiman [28] builds multiple classifiers using bootstrapped subsets of the original training dataset (using random sampling with replacement from the original dataset). The final decision is made 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. Recently

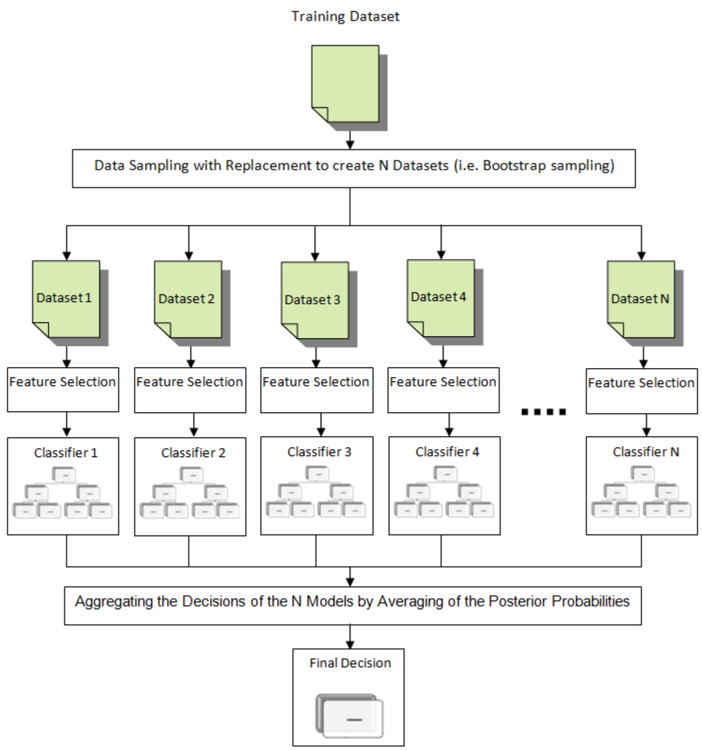



**Figure 6.1**: Select-Bagging

our research group developed an innovative bagging method called Select-Bagging (see Figure 6.4), which combines feature selection and bagging by implementing feature selection within each iteration of the bagging process. Note that feature selection is performed after the sampling with replacement for every iteration.

Boosting was developed by Freund et al. [52] to improve the performance of weak classifiers, which consists of iteratively training weak classifiers while modifying the weight of each instance at each iteration. The final decision is made by aggregating the decisions of all models as the weighted average of the posterior probabilities. Our research group recently proposed a new hybrid boosting algorithm called Select-Boosting(see Figure 6.4), which implements feature selection within each iteration of the boosting algorithm prior to building a classification model.

Both Select-Bagging and Select-Boosting improve upon the corresponding base ensemble process due to the fact that they incorporate a necessary process (i.e. fea-

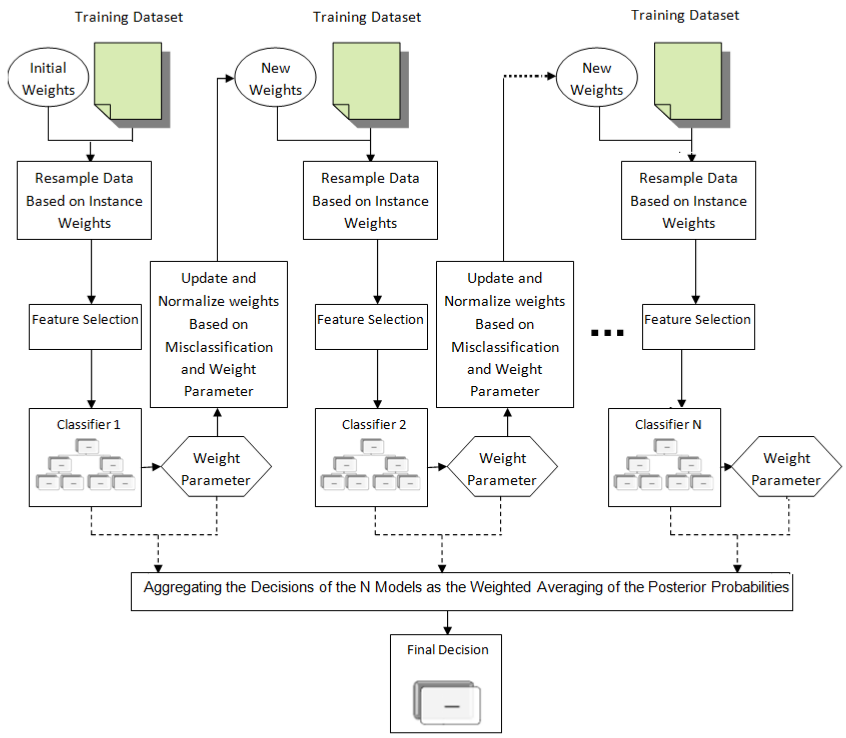



**Figure 6.2**: Select-Boosting

ture selection) to address the challenge of high-dimensionality as well as trying to improve upon the performance of the base classifiers used in this work (discussed in Section 6.5). Additionally, both Select-Bagging and Select-Boosting are implemented in the WEKA machine learning toolkit [115] using the default number of 10 iterations.

Random Forest [29] is another ensemble classifier developed by Breiman, which builds a set of unpruned decision trees on randomly bootstrapped data using a randomly-selected subset of features. A new instance is classified by all decision trees and the final classification is determined based on majority voting. Two of the advantages of Random Forest when compared with bagging and boosting is the simplicity of implementation and the ability to provide internal information including error, strength, and correlation. Random Forest is implemented by WEKA using an algorithm similar to C4.5. In this study, the `numTrees` parameter was set to 100 to build 100 unpruned trees. The classification is then performed by classifying a new instance using all trees (i.e. 100 trees) and choosing the class that is chosen by the most trees. As a result, we refer to Random Forest with 100 trees as RF100.

## 6.5 METHODOLOGY

In this work, we assess the effectiveness of three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest) as well as no ensemble classification when learning from balanced bioinformatics datasets with varying levels of data quality. We create three tiers of data quality by injecting 24 different patterns of noise. Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively. To ensure the validity of our study, we have only used the derived datasets where the final class ratio is not less than minority:majority of 35%:65% (we also take steps to preserve the class distribution of the derived training dataset). This way the results will be based on data that is relatively balanced or balanced rather than being imbalanced. We employ three feature ranking techniques (with three choices of feature subset size: 25, 50, and 100): Chi Squared (CS), Area Under the Receiver Operating Characteristic (ROC) Curve, and Wilcoxon Rank Sum (WRS) and one form of filter-based subset evaluation (i.e. Correlation-Based Feature Selection (CFS)). All feature selection techniques are discussed in more detail in Section 2.4. We utilize five different classifiers (discussed in Section 2.7): Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), and Logistic Regression (LR). We used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8. All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in

| Classification Approach | Learner | Feature Selection Technique | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CFS | CS | | | ROC | | | WRS | | |
| | | | 25 | 50 | 100 | 25 | 50 | 100 | 25 | 50 | 100 |
| FS | NB | 0.938714 | 0.956513 | 0.953103 | 0.943179 | 0.956028 | 0.948666 | 0.939478 | 0.954377 | 0.947126 | 0.937680 |
| | MLP | 0.935672 | 0.943843 | 0.944675 | 0.945738 | 0.947260 | 0.948661 | 0.947074 | 0.946588 | 0.948878 | 0.947759 |
| | 5-NN | 0.948073 | 0.949023 | 0.952733 | 0.954476 | 0.949853 | 0.954684 | 0.958132 | 0.949547 | 0.954197 | 0.958119 |
| | SVM | 0.919025 | 0.941822 | 0.926621 | 0.922729 | 0.946079 | 0.926757 | 0.925516 | 0.945114 | 0.926396 | 0.925454 |
| | LR | 0.835793 | 0.840424 | 0.820336 | 0.783013 | 0.849648 | 0.827367 | 0.796553 | 0.849502 | 0.827048 | 0.797516 |
| S-Bagging | NB | **0.969987** | 0.966241 | **0.967986** | 0.967798 | 0.968410 | 0.968258 | 0.966770 | 0.926520 | 0.953820 | 0.949216 |
| | MLP | 0.965501 | **0.966370** | 0.967258 | 0.967581 | 0.967020 | 0.967054 | 0.966641 | 0.928768 | 0.956948 | 0.948092 |
| | 5-NN | 0.967907 | 0.965915 | 0.966833 | 0.965726 | **0.969032** | 0.969405 | 0.967741 | 0.930339 | 0.960686 | 0.960024 |
| | SVM | 0.958868 | 0.961989 | 0.956894 | 0.955650 | 0.962313 | 0.959493 | 0.956783 | 0.919744 | 0.947789 | 0.943651 |
| | LR | 0.935108 | 0.936042 | 0.923325 | 0.905824 | 0.935723 | 0.924574 | 0.907587 | 0.894113 | 0.911332 | 0.891216 |
| S-Boosting | NB | 0.946000 | 0.944623 | 0.945783 | 0.945374 | 0.946270 | 0.945539 | 0.943422 | 0.876201 | 0.928124 | 0.927052 |
| | MLP | 0.952830 | 0.944086 | 0.948085 | 0.950054 | 0.951770 | 0.950644 | 0.953832 | 0.891908 | 0.938567 | 0.930944 |
| | 5-NN | 0.952348 | 0.951024 | 0.949947 | 0.954385 | 0.962530 | 0.963022 | 0.961809 | 0.899994 | 0.947746 | 0.948891 |
| | SVM | 0.938367 | 0.942414 | 0.941175 | 0.941517 | 0.946538 | 0.944783 | 0.947291 | 0.878087 | 0.921198 | 0.927357 |
| | LR | 0.925248 | 0.916980 | 0.909718 | 0.905929 | 0.931552 | 0.914415 | 0.915470 | 0.849923 | 0.876006 | 0.889813 |
| RF100 | | 0.969810 | 0.961039 | 0.967483 | **0.972250** | 0.964441 | **0.970975** | **0.973510** | **0.965038** | **0.970581** | **0.973878** |

**Table 6.1**: Average AUC Values for High-Quality Datasets

| Classification Approach | Learner | Feature Selection Technique | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CFS | CS | | | ROC | | | WRS | | |
| | | | 25 | 50 | 100 | 25 | 50 | 100 | 25 | 50 | 100 |
| FS | NB | 0.846259 | 0.882115 | 0.887053 | 0.876897 | 0.885221 | 0.883083 | 0.877170 | 0.881925 | 0.880477 | 0.873840 |
| | MLP | 0.841127 | 0.852434 | 0.856643 | 0.860319 | 0.852465 | 0.856454 | 0.858512 | 0.853081 | 0.856271 | 0.860329 |
| | 5-NN | 0.855814 | 0.854757 | 0.868199 | 0.876660 | 0.862673 | 0.869386 | 0.875825 | 0.860668 | 0.870053 | 0.875732 |
| | SVM | 0.818908 | 0.849873 | 0.827917 | 0.819558 | 0.855301 | 0.834271 | 0.823527 | 0.853203 | 0.832122 | 0.826300 |
| | LR | 0.746426 | 0.720921 | 0.707959 | 0.684786 | 0.731980 | 0.706748 | 0.689263 | 0.730871 | 0.708432 | 0.694580 |
| S-Bagging | NB | 0.899554 | 0.893249 | 0.901607 | **0.907346** | **0.907821** | **0.906201** | 0.902498 | 0.862662 | 0.882897 | 0.879014 |
| | MLP | 0.891365 | 0.892398 | 0.898503 | 0.899684 | 0.887261 | 0.890085 | 0.893225 | 0.849952 | 0.882252 | 0.886994 |
| | 5-NN | **0.900312** | **0.897701** | **0.904363** | 0.904781 | 0.901067 | 0.900539 | 0.899024 | 0.864168 | 0.889938 | 0.888273 |
| | SVM | 0.877157 | 0.889609 | 0.887034 | 0.881899 | 0.886238 | 0.884730 | 0.879396 | 0.845708 | 0.871102 | 0.868997 |
| | LR | 0.851757 | 0.840752 | 0.826754 | 0.822460 | 0.841709 | 0.835811 | 0.819069 | 0.799584 | 0.821847 | 0.797783 |
| S-Boosting | NB | 0.852513 | 0.843418 | 0.849068 | 0.852779 | 0.857887 | 0.857541 | 0.862265 | 0.812668 | 0.834319 | 0.844875 |
| | MLP | 0.872789 | 0.848026 | 0.860595 | 0.867191 | 0.859755 | 0.868815 | 0.872070 | 0.812087 | 0.855798 | 0.858059 |
| | 5-NN | 0.875185 | 0.863757 | 0.872753 | 0.875749 | 0.881923 | 0.884691 | 0.887977 | 0.841898 | 0.865152 | 0.866798 |
| | SVM | 0.855512 | 0.847319 | 0.845164 | 0.855011 | 0.853941 | 0.850899 | 0.854290 | 0.791376 | 0.829936 | 0.835787 |
| | LR | 0.841193 | 0.802472 | 0.809123 | 0.813668 | 0.825890 | 0.825272 | 0.821384 | 0.774258 | 0.795880 | 0.803735 |
| RF100 | | 0.887837 | 0.883836 | 0.897846 | 0.906954 | 0.887947 | 0.899205 | **0.905695** | **0.888119** | **0.897195** | **0.908605** |

**Table 6.2**: Average AUC Values for Average-Quality Datasets

Sections 2.1 and 2.3, respectively.

## 6.6 EXPERIMENTAL RESULTS

Tables 6.1 through 6.4 summarize the classification performances in terms of AUC. Each value represents the average AUC over the four runs of five-fold cross-validation when applying the given combination of classification approach, classifier, and feature selection technique to the datasets which match that data quality level. The numbers under the rankers (CS, ROC, and WRS) represent the number of features chosen from that ranked list. Table 6.1 shows the results for High-Quality datasets, Table 6.2, those for Average-Quality datasets, Table 6.3, those for Low-Quality datasets, and

| Classification Approach | Learner | Feature Selection Technique | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CFS | CS | | | ROC | | | WRS | | |
| | | | 25 | 50 | 100 | 25 | 50 | 100 | 25 | 50 | 100 |
| FS | NB | 0.793530 | 0.808023 | 0.809843 | 0.799754 | 0.828779 | 0.828006 | 0.820686 | **0.831360** | **0.826952** | 0.819913 |
| | MLP | 0.764141 | 0.764601 | 0.768631 | 0.773423 | 0.767957 | 0.773082 | 0.781071 | 0.771117 | 0.771100 | 0.780056 |
| | 5-NN | 0.725912 | 0.756649 | 0.762817 | 0.756570 | 0.760597 | 0.772782 | 0.777207 | 0.761619 | 0.769460 | 0.775421 |
| | SVM | 0.744010 | 0.774696 | 0.748190 | 0.730236 | 0.767734 | 0.757484 | 0.747911 | 0.770041 | 0.751905 | 0.745274 |
| | LR | 0.657126 | 0.664013 | 0.635127 | 0.598637 | 0.663187 | 0.626466 | 0.607848 | 0.669345 | 0.623410 | 0.602152 |
| S-Bagging | NB | **0.833435** | 0.811703 | **0.829611** | **0.837530** | **0.838645** | **0.848886** | **0.842858** | 0.750121 | 0.819610 | **0.833371** |
| | MLP | 0.818395 | 0.809911 | 0.814098 | 0.815811 | 0.805779 | 0.810515 | 0.817084 | 0.716904 | 0.805895 | 0.821475 |
| | 5-NN | 0.814445 | 0.798281 | 0.817522 | 0.817999 | 0.818500 | 0.817202 | 0.820258 | 0.738518 | 0.794354 | 0.819458 |
| | SVM | 0.806184 | **0.825355** | 0.808839 | 0.801230 | 0.816545 | 0.808836 | 0.804072 | 0.720228 | 0.795267 | 0.806561 |
| | LR | 0.778043 | 0.764267 | 0.729763 | 0.706923 | 0.759466 | 0.734655 | 0.714104 | 0.678838 | 0.726839 | 0.706406 |
| S-Boosting | NB | 0.761081 | 0.764747 | 0.763389 | 0.764849 | 0.769217 | 0.784227 | 0.772514 | 0.699732 | 0.744739 | 0.778695 |
| | MLP | 0.762938 | 0.754206 | 0.751496 | 0.773166 | 0.772750 | 0.769566 | 0.782609 | 0.702055 | 0.779893 | 0.788252 |
| | 5-NN | 0.774530 | 0.752579 | 0.762052 | 0.755876 | 0.792148 | 0.798079 | 0.792000 | 0.718983 | 0.760961 | 0.790087 |
| | SVM | 0.758864 | 0.766056 | 0.758948 | 0.745565 | 0.779630 | 0.762811 | 0.769304 | 0.701030 | 0.766284 | 0.769015 |
| | LR | 0.725338 | 0.698732 | 0.709069 | 0.711748 | 0.714943 | 0.722122 | 0.713037 | 0.652838 | 0.706740 | 0.691644 |
| RF100 | | 0.797145 | 0.788811 | 0.805099 | 0.814055 | 0.795963 | 0.810046 | 0.824465 | 0.792849 | 0.812036 | 0.824429 |

**Table 6.3**: Average AUC Values for Low-Quality Datasets

Table 6.4 shows the results for All Data Quality Levels (i.e. all datasets). The best choice of classification approach for each combination of feature selection technique and feature subset size are printed in **bold**. Due to space considerations, Select-Bagging and Select-Boosting are abbreviated in all tables as “S-Bagging” and “S-Boosting”.

The results for High-quality data (Table 6.1) demonstrate that two ensemble classification approaches, Select-Bagging and RF100, are the top performing approaches, achieving the highest performance four and six times, respectively. Select-Bagging improved the classification performance for all base learners regardless of the feature selection technique (except WRS with the feature subset size 25). In this single exceptional case, Select-Bagging only improved the classification performance for LR, while only slightly reducing the classification performance for the other learners. The best average AUC was obtained when RF100 was used with the WRS ranker with 100 features, followed by ROC with 100 features and CS with 100 features (0.973878, 0.973510, and 0.972250, respectively).

In Table 6.2, we see the results for Average-quality data. As can be seen, the top performing approaches are still Select-Bagging and RF100. Select-bagging was the top performing approach in 6 out of 10 scenarios, followed by RF100 in the other 4 scenarios. Select-Bagging improved the classification performance for all base learners

| Classification Approach | Learner | Feature Selection Technique | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | CFS | CS | | | ROC | | | WRS | | |
| | | | 25 | 50 | 100 | 25 | 50 | 100 | 25 | 50 | 100 |
| FS | NB | 0.888008 | 0.911911 | 0.912134 | 0.902107 | 0.915267 | 0.910577 | 0.902792 | 0.913532 | 0.908711 | 0.900565 |
| | MLP | 0.881052 | 0.889427 | 0.891858 | 0.894310 | 0.891617 | 0.894397 | 0.895271 | 0.891869 | 0.894206 | 0.896160 |
| | 5-NN | 0.888200 | 0.892003 | 0.899511 | 0.902713 | 0.895757 | 0.902148 | 0.906790 | 0.894999 | 0.901737 | 0.906536 |
| | SVM | 0.861981 | 0.888665 | 0.869675 | 0.862488 | 0.891998 | 0.873149 | 0.867488 | 0.891017 | 0.871518 | 0.868137 |
| | LR | 0.782181 | 0.776234 | 0.757655 | 0.725526 | 0.784912 | 0.759836 | 0.735284 | 0.785176 | 0.759910 | 0.737015 |
| S-Bagging | NB | **0.928245** | **0.921420** | **0.927485** | **0.930403** | **0.931026** | **0.931593** | 0.928763 | 0.882364 | 0.912183 | 0.910042 |
| | MLP | 0.921160 | 0.920965 | 0.924127 | 0.924926 | 0.918958 | 0.920560 | 0.922265 | 0.874970 | 0.911931 | 0.910903 |
| | 5-NN | 0.925157 | 0.921242 | 0.926426 | 0.926058 | 0.926501 | 0.926349 | 0.925305 | 0.883499 | 0.915257 | 0.917326 |
| | SVM | 0.911130 | 0.919536 | 0.913978 | 0.910569 | 0.917434 | 0.914500 | 0.910599 | 0.869149 | 0.901879 | 0.900325 |
| | LR | 0.886254 | 0.881125 | 0.865332 | 0.851945 | 0.880727 | 0.869829 | 0.852503 | 0.834250 | 0.856978 | 0.835403 |
| S-Boosting | NB | 0.890154 | 0.886604 | 0.889078 | 0.890377 | 0.893206 | 0.894502 | 0.893696 | 0.832153 | 0.872348 | 0.879665 |
| | MLP | 0.901228 | 0.886719 | 0.892998 | 0.898997 | 0.897162 | 0.899456 | 0.903851 | 0.840390 | 0.889729 | 0.887582 |
| | 5-NN | 0.903231 | 0.895795 | 0.899610 | 0.902255 | 0.913066 | 0.915030 | 0.914853 | 0.857358 | 0.895598 | 0.900281 |
| | SVM | 0.886999 | 0.887017 | 0.884744 | 0.886861 | 0.893174 | 0.889148 | 0.892452 | 0.825624 | 0.869754 | 0.875391 |
| | LR | 0.870999 | 0.849567 | 0.849426 | 0.849413 | 0.867521 | 0.859248 | 0.857307 | 0.799033 | 0.826849 | 0.835045 |
| RF100 | | 0.919580 | 0.912578 | 0.922928 | 0.929760 | 0.916686 | 0.925827 | **0.931211** | **0.916684** | **0.925137** | **0.932446** |

**Table 6.4**: Average AUC Values for All Datasets

regardless of the feature selection technique, except for WRS with feature subset size 25 with the NB, MLP, and SVM learners. The best average AUC was obtained when RF100 was used with WRS with 100 features, followed by Select-Bagging using ROC with 25 features and CS with 100 features (0.908605, 0.907821, and 0.907346, respectively).

Looking at the results for Low-quality data in Table 6.3, it can be seen that Select-Bagging is most frequently the top performing approach (8 out of 10 scenarios), followed by FS (2 out of 10 scenarios). Select-Bagging improved the classification performance for all base learners regardless of the feature selection technique, with exceptions only when considering WRS with feature subset size 25 and feature subset size 50. The best average AUC was obtained when Select-Bagging was used with ROC with 50 features, 100 features, and 25 features (0.848886, 0.842858, and 0.838645, respectively).

Looking at the results for all datasets in Table 6.4, we find that Select-bagging was the top performing approach in 6 out of 10 scenarios, followed by RF100 in the other 4 scenarios. The best average AUC was obtained when RF100 was used with WRS with 100 features, followed by Select-Bagging with ROC with 50 features and RF100 with ROC with 100 features (0.932446, 0.931593, and 0.931211, respectively).

Furthermore, we see that both ensemble classification approaches Select-Bagging and Select-Boosting consistently improved the classification performance for LR regardless of the data quality level and feature selection technique (except for low-quality data when using Select-Boosting with the ranker WRS with 25 features). This gives us confidence to recommend these ensemble classification approaches (especially Select-Bagging) to improve the classification performance for the LR learner regardless of the data quality level and feature selection technique. Similarly, Select-Bagging and Select-Boosting consistently improved the classification for all base learners when the CFS feature selection technique is used regardless of the data quality level (except for low-quality data when using the NB and MLP learners).

We also performed a set of ANalysis Of VAriance (ANOVA) tests [23] to find statistically significant patterns. We used MATLAB to perform the ANOVA analysis and subsequent statistical tests. In this analysis, we considered only one factor: the choice of classification approach, with four different levels of this factor (FS, Select-Bagging, Select-Boosting, and RF100). The tests performed were across all datasets and factors, and for each level of data quality. For the ANOVA tests, the AUC results across all learners were used as the response variable. This was done to get a general trend in terms of the FS, Select-Bagging, and Select-Boosting approaches. In addition, the separate results from all four runs of five-fold cross-validation were used as individual values in the analysis. The results are presented in Table 6.5. Note that a significance factor of 5% was chosen, and thus the $Prob > F$ value must be less than this value (e.g. 0.05) for the result to be significant. These results show that the choice of classification approach is significant across all data quality levels (i.e. all datasets), as well as each level of data quality; that is to say, when the data are grouped by the classification approach, at least two of those groups will have significantly different means.

To discover which pairs of means are significantly different, we conducted multiple pairwise comparison by using Tukey's Honestly Significant Difference (HSD)

| Datasets | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| All | Approach | 78.52 | 3 | 26.1739 | 1500.5 | 0 |
| | Error | 4186.35 | 239996 | 0.0174 | | |
| | Total | 4264.88 | 239999 | | | |
| High | Approach | 27.16 | 3 | 9.0541 | 1028.08 | 0 |
| | Error | 1099.06 | 124796 | 0.00881 | | |
| | Total | 1126.22 | 124799 | | | |
| Average | Approach | 41.11 | 3 | 13.7019 | 751.21 | 0 |
| | Error | 1575.85 | 86396 | 0.0182 | | |
| | Total | 1616.95 | 86399 | | | |
| Low | Approach | 14.291 | 3 | 4.76358 | 199.79 | 2.94E-128 |
| | Error | 686.566 | 28796 | 0.02384 | | |
| | Total | 700.857 | 28799 | | | |

**Table 6.5**: ANOVA Results: Ensemble Approaches

criterion [23]. The significance level for Tukey's HSD test is $\alpha = 0.05$. Figure 6.3 shows the comparison results of the four classification approaches for all data quality levels, and the different levels of data quality; Figure 6.4(a) shows the results for All Data Quality Levels (i.e. all datasets), Figure 6.4(b), those for High-Quality datasets, Figure 6.4(c), those for Average-Quality datasets, and Figure 6.4(d) shows the results for Low-Quality datasets. The figures display graphs with each group mean represented by a symbol ($\circ$) and the 95% confidence interval as a line around the symbol. Two means are significantly different if their intervals are disjoint, otherwise they are not significantly different.

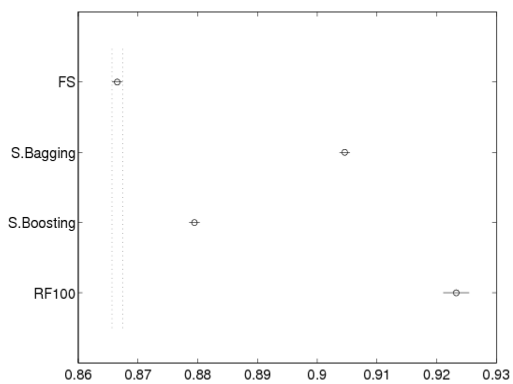
The results show that the top performing classification approach is consistently RF100, followed by Select-Bagging and Select-Boosting, regardless of the data quality level. It also shows that all classification approaches are statistically different from each other for all the datasets together, as well as for each different level of data quality (except for Low-quality data when considering Select-Boosting and FS). In the case of this exception, although there is no statistical difference between Select-Boosting and FS (i.e. No ensemble), Select-Boosting does have a slightly higher average AUC compared to FS. Thus, based on the general performance and the statistical analysis we recommend using ensemble classification techniques to improve the classification performance when learning from balanced bioinformatics datasets regardless of the data quality level, especially RF100 as it is the top performing classification approach and it does not rely on the choice of the base classifier (unlike the other ensemble approaches).

## 6.7 CHAPTER SUMMARY

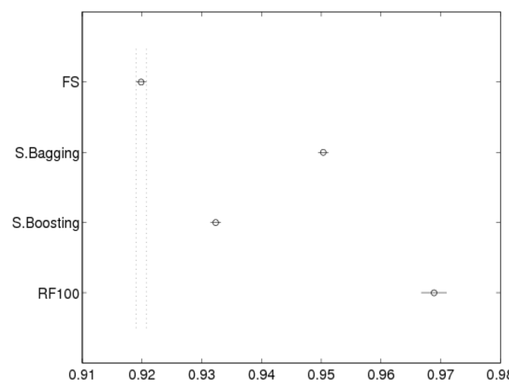
Building classification models using bioinformatics datasets that can accurately classify new instances is a challenging endeavor due to the small sample size and the large number of features. While techniques such as ensemble classification and feature selection help alleviate these problems, no previous work has considered the effectiveness of ensemble classification along with feature selection when learning from balanced bioinformatics datasets in the context of data quality (i.e. varying levels of data quality due to the presence of noise). In this work, we compare three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest), as well as feature selection followed by single classifiers, using ten high-dimensional and balanced gene expression datasets which were first determined to be free of noise, and then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-quality, Average-quality, and Low-Quality). We employ three feature rankers (with three feature subset sizes) along with Correlation Based Feature selection to alleviate the high-dimensionality, and we utilize five classification algorithms to build the classification models.

The experimental results demonstrate that in order to improve the classification performance for models built with balanced bioinformatics datasets, utilizing an ensemble classification technique is the best strategy. We found that the best performing approaches are consistently RF100 and Select-Bagging regardless of the data quality level. These two approaches significantly improved the classification performance compared to no ensemble classification regardless of the data quality level. All of these results are supported by ANOVA and Tukey's Honestly Significant Difference statistical tests. Thus, we recommend using ensemble classification techniques when learning from balanced high-dimensional bioinformatics datasets regardless of any implication of noise. In particular, we recommend using RF100 as it is significantly the top performing classification approach, on average, regardless of the data quality level. Additionally, RF100 does not rely on the choice of the base classifier, unlike the other ensemble approaches. It is of note that Select-Bagging and Select-Boosting consistently improved the classification performance for LR for all data quality levels and feature selection techniques, with one insignificant exception.

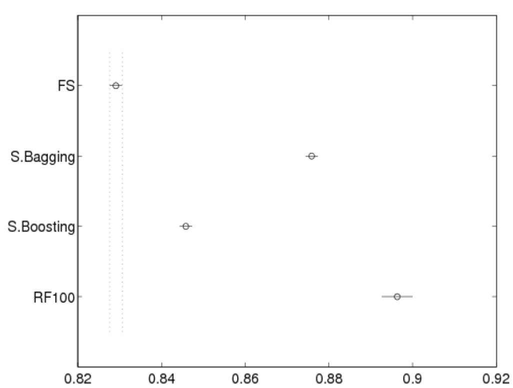
**Figure 6.3**: Tukey's HSD Results: Ensemble Approaches



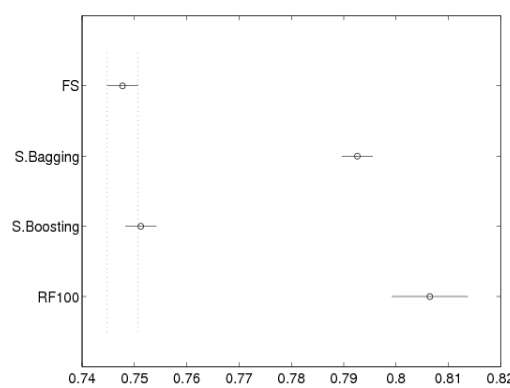

(a) All Datasets




(b) High Quality




(c) Average Quality




(d) Low Quality

# CHAPTER 7

# THE IMPORTANCE OF ALLEVIATING CLASS IMBALANCE FOR CLASSIFICATION PROBLEMS ON BIOINFORMATICS DATA

## 7.1 INTRODUCTION

One of the most important tools for identifying cancerous samples is the use of classification models built using gene expression datasets, which categorizes unknown samples as "healthy" or "cancerous" based on the prior knowledge learned from these datasets. The process of building cancer classification models involves analyzing data with unequal distribution of instances between classes (i.e. class imbalance). Traditional classification algorithms fail here, primarily because they were designed with the goal of maximizing overall classification accuracy without regard to the significance of different classes. Thus, traditional classifier will be biased towards the majority class and will perform poorly on the minority class (having a very high rate of false negatives), which is the class of interest. In the medical domain, the cost of false negatives is overwhelmingly greater than the cost of false positives. Therefore, there is clearly a need to handle class imbalance in this domain.

Data sampling is a popular technique used to alleviate the class imbalance, where the dataset is transformed into a more balanced one by adding or removing instances. There are a number of advantages when class imbalance is alleviated, including: improved classification performance, decreased bias towards the majority class, and decreased rate of false negatives. Despite the prevalence of the class imbalance among gene expression datasets, most previous studies have ignored the subject entirely or provided shallow treatments of the area. This study shows the importance of taking into account class imbalance when analyzing bioinformatics datasets.

In addition to class imbalance, many gene expression datasets are also characterized by the problems of high dimensionality (i.e. large number of attributes) and data noise (erroneous values in the dataset). An extremely large number of studies in bioinformatics have focused on the challenge of high dimensionality and investigated solutions to cope with it (e.g. feature selection) because traditional classifiers become computationally unfeasible with an extremely large number of genes. On the other hand, the presence of data noise is arguably inevitable. Noise can hinder machine learning and make accurate classification more difficult. Therefore, it is important to study the impact of data noise on data mining techniques and discover techniques which are robust and less sensitive to noise.

In this chapter [12], the primary goal is to investigate the importance of taking into account the problem of class imbalance on bioinformatics datasets. To investigate this importance, we compare the classification performance of two approaches. In the first approach feature selection (FS) is performed alone (i.e. no data sampling), and then a classifier is built using the selected features. Alternatively, in the second approach (FS-DS) we apply data sampling after performing feature selection, and then a classifier is built using the selected features and the sampled data. To test these approaches under different circumstances we create three levels of data quality by injecting artificial class noise into ten noise-free gene expression datasets. Furthermore, we employ a diverse range of data mining techniques. We utilize three forms of feature selection techniques as well as six varied classification learners.

## 7.2 CONTRIBUTIONS

The primary contribution of this chapter is to highlight the importance of alleviating class imbalance for classification problems on bioinformatics datasets. This chapter: (1) simulates real-world scenarios by injecting class noise into ten real-world gene-expression datasets (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality); (2) examines three major forms of feature selection techniques (rankers, filter-based subset selection, and wrapper subset selection) and (4) employs six classifiers that are commonly-used in the literature.

The remainder of this chapter will be organized as follows: Section 7.3 presents related works on the topics of high dimensionality, class imbalance, and data noise. Section 7.4 outlines the methods used in this work. In Section 7.5, we present our results. Finally, Section 7.6 concludes our work.

## 7.3 RELATED WORK

A dataset is imbalanced when the positive class instances (those which represent the instances with the most important condition, such as patients with cancer) are significantly outnumbered by the instances of the negative class (those which are less interesting, such as cancer-free patients). Class imbalance can harm classification performance because many classification algorithms were designed with the goal of maximizing overall classification accuracy without regard to the significance of different classes, possibly resulting in an increased bias towards the majority class as well as an increased number of false negatives (misclassification of the class of interest) [59, 108]. Additionally, data mining activities such as attribute selection can be impacted by imbalanced class distributions [18].

The primary method used to deal with this problem is known as data sampling, where the dataset is modified to balance it [73]. Sampling techniques are divided into undersampling and oversampling. In the former, instances of the majority class are deleted until the target class ratio is reached. Oversampling takes the opposite approach, duplicating instances in the minority-class until the target class ratio is reached. In our work, we used Random Undersampling (RUS) to reduce the dataset size and make subsequent analysis computationally more efficient compared to oversampling. Additionally prior research showed its effectiveness [104].

While class imbalance is a frequent problem within bioinformatics datasets, only a few studies investigated this problem and applied techniques to cope with it. In 2005, Al-Shahib et al. [16] used undersampling as well as a wrapper based-feature selection to build classifiers to predict protein function from amino acid sequence features. Classifiers were built on the "one vs. all" model, with each classifier deciding if instances are in a given class or not. The results showed that the top results combined undersampling and the wrapper feature selection along with the SVM classifier. Blagus and Lusa [26] performed a study using data sampling on high-dimensional data. They used two data sampling techniques, Random Undersampling and SMOTE, on high-dimensional class-imbalanced breast cancer gene expression datasets and a series of classifiers. They showed that only the k-NN classifiers seem to benefit substantially from SMOTE and a number of the other classifiers seem to prefer Random Undersampling.

The computational infeasibility of traditional classifiers when dealing with gene expression datasets mandates the use of dimensionality reduction techniques. Therefore, an extremely large number of studies have focused on developing and evaluating gene selection strategies [53, 94, 62, 65, 82, 86]. Unfortunately, the vast majority of these studies have completely ignored the fact that most gene expression datasets are class imbalanced.

Data noise is another frequent problem in the bioinformatics domain. Zhu and Wu [117] showed that data noise can lead to suboptimal classification performance and class noise is more harmful than the other type of noise. Therefore, all experiments presented in this study were performed on data which was first determined to be free of noise and then had artificial class noise added in a controlled fashion. Thus, the results can be used to simulate how they would be used in the field.

## 7.4 METHODOLOGY

Bioinformatics datasets are commonly characterized by their high dimensionality and the skewed class distribution (instances of the class of interest are outnumbered by instances of the other class(es)). Feature selection and data sampling are the two most common processes to counter high dimensionality and class imbalance, respectively. Although bioinformatics datasets exhibit both problems simultaneously, most studies have focused on high dimensionality but have completely ignored class imbalance.

In this work we investigate the importance of alleviating the class imbalance when analyzing bioinformatics datasets. To study this importance we compare two approaches. Figure 7.1 depict graphically the two approaches applied in this study. The first approach (FS) consists solely of feature selection, and then a classifier is built using the selected features. In the second approach (FS-DS) [15], feature selection takes place before data sampling is performed, and then a classifier is built using the selected features and the sampled data. We selected the FS-DS because preliminary experimentation showed that it is the best approach for utilizing feature selection and data sampling. In summary, the difference between the two approaches is based on whether we employ data sampling or not. This way, the results can be used to determine if data sampling is beneficial in improving the performance for classification models built with bioinformatics datasets.

In this work we utilize 11 different feature selection strategies: three feature rankers (Chi Squared (CS), Area Under the Receiver Operating Characteristic (ROC) Curve, and Wilcoxon Rank Sum (WRS)) each coupled with three feature subset sizes (25, 50, and 100), a filter-based subset evaluator (Correlation-Based Feature Selection (CFS)), and wrapper-based feature selection using Naïve Bayes inside the wrapper. All feature selection techniques are discussed in more detail in Section 2.4. Six learners (discussed in Section 2.7) were chosen for our analysis: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines

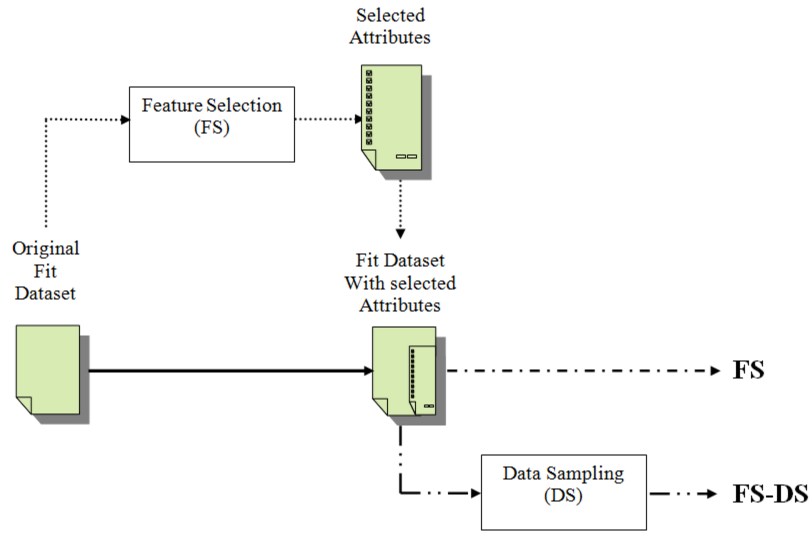



**Figure 7.1**: Evaluation Approaches

(SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively. To ensure the validity of our study, we have only used the derived datasets where the final class ratio is 20%:80%. This way the results will be based on data that exhibit class imbalance rather than being balanced or fairly imbalanced. For all experiments in this chapter, we used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8.

## 7.5 EMPIRICAL RESULTS

The primary goal of our experiments is to investigate the importance of handling class imbalance for classification problems on bioinformatics datasets by applying techniques such as data sampling. To investigate this importance we compare the classification performance of two approaches. The first approach (FS) does not employ any technique to handle class imbalance and only employs feature selection. On the other hand, the second approach (FS-DS) employs data sampling after performing feature selection. We utilized 11 feature selection techniques representing the three major types of feature selection (ranker-based techniques, filter-based subset selection, and wrapper-based feature selection). For the FS-DS we employed random undersampling because it is commonly used in the literature and prior research showed its effectiveness. We created three levels of data quality (High-Quality, Average-Quality, and Low-Quality) by injecting class noise into ten gene expression datasets (relatively free of noise) in a controlled fashion. We build our final models using six commonly-used classification algorithms.

Table 7.1 summarizes the classification performances in terms of AUC. Each value represents the average AUC over the four runs of five-fold cross-validation when applying the given combination of feature selection technique, evaluation approach, and classifier to the datasets which match that data quality level. In the "Feature Selection Technique" column, the rankers (CS, ROC, and WRS) are followed by a number, which represents the number of features chosen from that ranked list, and the wrapper-based selection approach which uses the NB learner inside the wrapper is abbreviated as "WrapNB" for space considerations. The table includes six sub-tables: one for each classifier (NB, MLP, 5-NN, SVM, RF100, and LR, respectively). The sub-tables also present the average performance (last row of the sub-tables) of each of the approaches over the 11 feature selection techniques and datasets which match that data quality level for that specific learner. The last row of the table represents the overall average performance of each of the approaches for that specific data quality level. The best choice of approach for each combination of learner and data quality are printed in **bold**.

Overall, we can make the general statement that in order to improve the perfor-

| Learner | Feature Selection Technique | High Quality | | Average Quality | | Low Quality | |
|---|---|---|---|---|---|---|---|
| | | FS | FS-DS | FS | FS-DS | FS | FS-DS |
| NB | CS25 | 0.969516 | 0.958764 | 0.895590 | 0.897738 | 0.719840 | 0.730657 |
| | CS50 | 0.970508 | 0.956419 | 0.902930 | 0.901652 | 0.719909 | 0.717874 |
| | CS100 | 0.970697 | 0.955836 | 0.901985 | 0.900166 | 0.712186 | 0.716005 |
| | ROC25 | 0.979886 | 0.970850 | 0.912904 | 0.917641 | 0.751071 | 0.772535 |
| | ROC50 | **0.980254** | 0.970873 | 0.911979 | **0.917992** | 0.754139 | 0.786815 |
| | ROC100 | 0.977968 | 0.971127 | 0.902382 | 0.915377 | 0.742264 | **0.788152** |
| | WRS25 | 0.979393 | 0.970763 | 0.910355 | 0.915736 | 0.743097 | 0.766034 |
| | WRS50 | 0.979589 | 0.970059 | 0.908369 | 0.914749 | 0.744373 | 0.777153 |
| | WRS100 | 0.977409 | 0.970455 | 0.895318 | 0.909554 | 0.731238 | 0.779433 |
| | CFS | 0.964765 | 0.968279 | 0.856189 | 0.875105 | 0.712287 | 0.727862 |
| | WrapNB | 0.873768 | 0.875268 | 0.723632 | 0.740975 | 0.615231 | 0.614892 |
| | Average | **0.965796** | 0.958063 | 0.883785 | **0.891517** | 0.722331 | **0.743401** |
| MLP | CS25 | 0.957909 | 0.955020 | 0.874440 | 0.872896 | 0.738998 | 0.744182 |
| | CS50 | 0.960888 | 0.958370 | 0.870329 | 0.882694 | 0.762635 | 0.766761 |
| | CS100 | 0.965276 | 0.970009 | 0.876488 | 0.891741 | 0.766032 | 0.767741 |
| | ROC25 | 0.972561 | 0.972603 | 0.889551 | 0.896138 | 0.772974 | 0.782343 |
| | ROC50 | 0.971411 | 0.974620 | 0.889696 | 0.904404 | 0.773919 | 0.791021 |
| | ROC100 | 0.973277 | **0.979377** | 0.891228 | **0.908642** | 0.775144 | **0.796274** |
| | WRS25 | 0.972087 | 0.972572 | 0.890120 | 0.896583 | 0.776447 | 0.782432 |
| | WRS50 | 0.971478 | 0.974499 | 0.892507 | 0.904135 | 0.779396 | 0.790007 |
| | WRS100 | 0.972624 | 0.978787 | 0.894527 | 0.906057 | 0.777489 | 0.792431 |
| | CFS | 0.972128 | 0.977827 | 0.881703 | 0.895072 | 0.749556 | 0.760544 |
| | WrapNB | 0.897627 | 0.881394 | 0.795654 | 0.772147 | 0.652678 | 0.622512 |
| | Average | 0.962479 | **0.963189** | 0.876931 | **0.884592** | 0.756842 | **0.763295** |
| 5-NN | CS25 | 0.970207 | 0.976777 | 0.856415 | 0.890983 | 0.681958 | 0.744939 |
| | CS50 | 0.976821 | 0.981959 | 0.867209 | 0.901058 | 0.691052 | 0.768582 |
| | CS100 | 0.983206 | **0.986398** | 0.881554 | 0.904065 | 0.695454 | 0.771755 |
| | ROC25 | 0.973206 | 0.981193 | 0.871361 | 0.911186 | 0.701723 | 0.786333 |
| | ROC50 | 0.978648 | 0.983867 | 0.885872 | 0.919467 | 0.718192 | 0.794923 |
| | ROC100 | 0.983968 | 0.986343 | 0.896450 | 0.922232 | 0.724674 | 0.802621 |
| | WRS25 | 0.973461 | 0.980829 | 0.871335 | 0.912627 | 0.703728 | 0.788087 |
| | WRS50 | 0.978743 | 0.983764 | 0.886252 | 0.920598 | 0.710994 | 0.793919 |
| | WRS100 | 0.984148 | 0.986241 | 0.897483 | **0.922761** | 0.724432 | **0.805558** |
| | CFS | 0.974902 | 0.984968 | 0.856348 | 0.906968 | 0.674918 | 0.762576 |
| | WrapNB | 0.871213 | 0.887178 | 0.757480 | 0.779478 | 0.601233 | 0.634457 |
| | Average | 0.968048 | **0.974502** | 0.866160 | **0.899220** | 0.693487 | **0.768523** |
| SVM | CS25 | 0.949075 | 0.942009 | 0.857103 | 0.859561 | 0.720630 | 0.730785 |
| | CS50 | 0.941267 | 0.941478 | 0.838170 | 0.856167 | 0.726486 | 0.742296 |
| | CS100 | 0.945576 | 0.953316 | 0.836320 | 0.859383 | 0.731150 | 0.748844 |
| | ROC25 | 0.965738 | 0.963434 | 0.876210 | 0.886538 | 0.752620 | 0.773483 |
| | ROC50 | 0.957807 | 0.959919 | 0.860315 | 0.881459 | 0.751254 | 0.771793 |
| | ROC100 | 0.953708 | 0.965543 | 0.858416 | 0.883180 | 0.749145 | **0.778840** |
| | WRS25 | 0.965331 | 0.963588 | 0.873680 | **0.886563** | 0.754862 | 0.771534 |
| | WRS50 | 0.957439 | 0.960354 | 0.860538 | 0.883548 | 0.754474 | 0.773359 |
| | WRS100 | 0.953602 | 0.965468 | 0.858431 | 0.883186 | 0.753271 | 0.778099 |
| | CFS | 0.958215 | **0.967371** | 0.848835 | 0.868891 | 0.732260 | 0.750160 |
| | WrapNB | 0.887550 | 0.886718 | 0.772574 | 0.774072 | 0.621365 | 0.627824 |
| | Average | 0.948664 | **0.951745** | 0.849145 | **0.865686** | 0.731593 | **0.749729** |
| RF100 | CS25 | 0.979556 | 0.976871 | 0.893113 | 0.896443 | 0.739130 | 0.759041 |
| | CS50 | 0.984723 | 0.982120 | 0.909327 | 0.914123 | 0.759805 | 0.778019 |
| | CS100 | 0.989225 | 0.986335 | 0.922937 | 0.922435 | 0.773646 | 0.783330 |
| | ROC25 | 0.983257 | 0.978855 | 0.908410 | 0.915990 | 0.767508 | 0.785381 |
| | ROC50 | 0.986803 | 0.985208 | 0.920943 | 0.927072 | 0.782087 | 0.794035 |
| | ROC100 | **0.989852** | 0.987553 | 0.926962 | 0.931585 | 0.793339 | 0.803700 |
| | WRS25 | 0.982729 | 0.977889 | 0.908789 | 0.916362 | 0.764786 | 0.781745 |
| | WRS50 | 0.987269 | 0.984265 | 0.922320 | 0.926414 | 0.783083 | 0.798303 |
| | WRS100 | 0.989569 | 0.987996 | 0.927036 | **0.931910** | 0.795667 | **0.804954** |
| | CFS | 0.989203 | 0.985364 | 0.920940 | 0.905618 | 0.750484 | 0.747825 |
| | WrapNB | 0.880631 | 0.888229 | 0.781657 | 0.779135 | 0.611871 | 0.609267 |
| | Average | **0.976620** | 0.974608 | 0.903858 | **0.906099** | 0.756491 | **0.767782** |
| LR | CS25 | 0.853032 | 0.865661 | 0.774650 | 0.795265 | 0.657146 | 0.715298 |
| | CS50 | 0.829788 | 0.876675 | 0.725889 | 0.789230 | 0.645076 | 0.724821 |
| | CS100 | 0.834142 | 0.887806 | 0.717485 | 0.802440 | 0.635654 | 0.729219 |
| | ROC25 | 0.885725 | 0.921165 | 0.788996 | 0.826819 | 0.683913 | 0.754559 |
| | ROC50 | 0.864656 | 0.926523 | 0.749217 | 0.817484 | 0.659120 | 0.755278 |
| | ROC100 | 0.850500 | 0.923279 | 0.740576 | 0.812882 | 0.643436 | 0.752656 |
| | WRS25 | 0.885544 | 0.923709 | 0.787911 | 0.827983 | 0.683255 | **0.757058** |
| | WRS50 | 0.863800 | 0.925990 | 0.752706 | 0.819583 | 0.658126 | 0.756025 |
| | WRS100 | 0.849328 | 0.923462 | 0.737418 | 0.814444 | 0.638876 | 0.753414 |
| | CFS | 0.867437 | **0.952627** | 0.730162 | **0.842181** | 0.654826 | 0.747270 |
| | WrapNB | 0.887879 | 0.872141 | 0.760602 | 0.743522 | 0.638327 | 0.621363 |
| | Average | 0.861075 | **0.909003** | 0.751419 | **0.808348** | 0.654341 | **0.733360** |
| Overall | Average | 0.947114 | **0.955185** | 0.855216 | **0.875910** | 0.719181 | **0.754348** |

**Table 7.1**: Average AUC Values

mance for classification models built with bioinformatics datasets that exhibit both high dimensionality and class imbalance simultaneously, alleviating class imbalance in conjunction with reducing high dimensionality is the best strategy. The overall average performance shows that FS-DS outperforms FS across the board (regardless of the data quality level). When we look at the "Average" row in each sub-table showing the performance across all feature selection strategies, we find that FS-DS is the best performing approach for all combinations of data quality tiers and learners (except High-Quality with RF100 and NB). For all but 2 of 18 combinations of learner and data quality level (High Quality with the NB and RF100 learners) the best approach was FS-DS. It is of note that FS-DS consistently outperformed FS (regardless of the data quality level and feature selection technique) when the 5-NN learner is used.

When we look at these results in terms of the different feature selection techniques, it can be seen that for the subset selection technique CFS, the best performing approach was consistently FS-DS regardless of the learner and data quality level. The only exception to this is when considering the RF100 learner. When considering rankers, we can see that for all but 6 out of 108 combinations of learner and ranker with both Average-Quality and Low-Quality data (Average-Quality with NB learner and CS ranker with 100 features or RF100 learner; Average-Quality with MLP learner and CS ranker with 25 features or RF100 learner; and Average-Quality and Low-Quality with NB learner and CS ranker with 50 features), the best approach was FS-DS. This is especially important as these two tiers of data quality represent higher levels of noise. When considering High-Quality data, FS-DS was at the top performing approach for 39 out of 66 combinations.

To discover which (if any) of the above patterns were statistically significant, we used a set of two tailed z-test for each paired comparison. The tests performed were across all datasets and factors, and for each level of data quality. The z-test method tests the null hypothesis that the population means related to two independent group

| Datasets | z-value | p-value |
|---|---|---|
| All Data Quality Levels | -26.2156616 | <0.0001 |
| High Quality | -15.29787687 | <0.0001 |
| Average Quality | -16.95388351 | <0.0001 |
| Low Quality | -21.45216765 | <0.0001 |

**Table 7.2**: z-test Results

samples are equal against the alternative hypothesis that the population means are different. p-values are provided for each pair of comparisons in the table. The significance level is set to 0.05; when the p-value is less than 0.05, the two group means are significantly different from one another.

The results are presented in Table 7.2. These results supports our conclusion that the top performing choice of approach is always FS-DS and the difference between the top performing approach and the other approach (i.e. FS) is statistically significant across all data quality levels and for each level of data quality.

## 7.6 CHAPTER SUMMARY

Class imbalance is a prevalent problem in the bioinformatics domain, which occurs when there is unequal distribution of instances between classes. This challenge can adversely affect the classification performance and effectiveness of feature selection. Unfortunately, this problem has almost been completely neglected. In this work we show the importance of alleviating class imbalance for classification problems on bioinformatics datasets. To investigate this importance we compare two approaches. The difference between the two approaches is whether we apply data sampling (commonly used technique to alleviate class imbalance) or not. In the first approach (FS), we employ feature selection alone to reduce high dimensionality. In the second approach (FS-DS), we perform feature selection followed by data sampling. We created three categories of datasets by injecting artificial class noise in a controlled fashion into ten gene expression datasets which were first determined to be relatively free of noise.

We built our final models using six different classification algorithms.

Our results show that feature selection alone does not perform as well as when we incorporate data sampling as well. FS-DS was most frequently the top performing approach and its overall average performance showed superior classification performance compared to FS (regardless of the data quality level). We also performed a series of z-tests and found that the difference between the two approaches is statistically significant across all factors and for each level of data quality. For these reasons we recommend alleviating the class imbalance (e.g. by applying data sampling) to achieve improved classification performance for bioinformatics classification problems.

# CHAPTER 8

# COMPARISON OF THREE APPROACHES FOR COMBINING FEATURE SELECTION AND DATA SAMPLING USING BIOINFORMATICS DATA VARYING LEVELS OF DATA QUALITY

## 8.1 INTRODUCTION

The previous chapter demonstrated that alleviating the class imbalance (e.g. by applying data sampling) can help achieve improved classification performance for bioinformatics classification problems. Additionally, feature selection in bioinformatics has become not only useful but necessary because of the high levels of high-dimensionality. Because of this, in the chapter [9, 13] we chose to utilize both feature selection and data sampling jointly. We compare three different approaches for combining feature selection and data sampling using real-world bioinformatics datasets that exhibit both high dimensionality and class imbalance in the context of data quality. The difference between one approach and another is based on two main questions, whether feature selection or data sampling should come first and whether to use original or sampled data to build the training dataset. With the first two approaches, data sampling is followed by feature selection. In the first approach (DS-FS-UnSam), the features are selected based on the sampled data, and then the unsampled data is used with just the selected features. The second approach (DS-FS-Sam) is similar, but the sampled data is used. Finally, with the third approach (FS-DS), feature selection is performed prior to sampling. To the best of our knowledge this is the first study to compare three approaches for utilizing feature selection and data sampling using real-world bioinformatics datasets that exhibit both high dimensionality and class imbalance in the context of data quality.

In order to compare the three different approaches in the context of data quality, all experiments in this chapter were performed on data which was first determined to be free of noise then had artificial class noise injected in a controlled fashion creating three data quality levels (High-Quality, Average-Quality, and Low-Quality). This way, the results can be used to determine the effectiveness of the aforementioned approaches and points out the importance of considering the order of feature selection and data sampling when working with bioinformatics datasets that exhibit both problems (high dimensionality and class imbalance) simultaneously for different data quality levels. Furthermore, we wanted to discover the robustness of the feature ranking techniques and the classification algorithms to class noise when combined with the best approach for combining feature selection and data sampling.

## 8.2 CONTRIBUTIONS

The primary contribution of this chapter is to address the combined problem of high dimensionality and class imbalance in the context of data quality in order to determine best practices. To determine best practices: (1) this study compares three approaches to combining feature selection and data sampling and determine which one performs best across many real-world bioinformatics datasets, all of which exhibit high dimensionality and class imbalance; (2) this study injects class noise into the data (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality); (3) this study examines ten different feature rankers from three different families with four subset sizes, making the results more generalizable; (4) this study applies three different sampling techniques; (5) this study employs six classifiers that are commonly-used in the literature and (6) this study investigates the robustness of classification algorithms and feature selection techniques to varying data quality levels.

The remainder of this chapter will be organized as follows: Section 8.3 presents related works on the topics of high dimensionality, class imbalance, and data noise. Section 8.4 outlines the methods used in this work. In Section 8.5, we present our results. Finally, Section 8.6 concludes our chapter.

## 8.3 RELATED WORK

Many bioinformatics datasets are characterized by high dimensionality [94], which refers to the high abundance of features, in most cases exceeding the number of instances/cases [99, 80]. This overabundance of features can hinder the data mining process, requiring extensive computation time and reducing the predictive performance of inductive models, because usually, most of these attributes will be irrelevant to the problem at hand. Feature selection is an important preprocessing technique used to counter high dimensionality, which consists of finding a minimum subset of features that have the highest correlation with the class by selecting the most relevant attributes and removing irrelevant and redundant attributes [44]. Reducing the number of features in a dataset can improve classification performance of inductive models, improve model interpretability, and speed up the learning process [94].

Reducing the number of features in bioinformatics data has become not only useful but necessary because of the high levels of high-dimensionality. For this reason, feature selection has received a lot of attention in the past few years. A review of feature selection techniques in bioinformatics can be found in the work of Saeys et al. [94], including both wrapper-based techniques and filter-based techniques. Mishra and Sahu [38] proposed two approaches using signal-to-noise for gene selection. The authors evaluated the effectiveness of each technique using the accuracy of two learners (k-nearest neighbor and support vector machine) and showed that the classification accuracy can be improved after applying signal-to-noise ranking and selecting top scored features from each cluster. Abeel et al. [1] studied the process for selecting biomarkers from microarray data and presented a general framework for stability analysis of such feature selection techniques. The results showed that stability could be improved through ensemble feature selection, where the training data is bootstrapped, recursive feature elimination (RFE) is applied to each subset, and either a complete linear or complete weighted linear aggregation method is used to get a consensus output. A study performed by Inza et al. [62] 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 feature selection drastically reduced computation time.

Class imbalance is another major challenge to machine learning, where there are few cases of the positive class (the class of interest) and many more cases of the negative class. Class-imbalanced data are common in the bioinformatics domain [20]. Ramaswamy et al. [92] performed feature selection on a dataset where only 16% of the instances are in the class of interest. Shipp et al. [97] classified diffuse large B-cell lymphoma from follicular lymphoma using a dataset with a 25% class imbalance. Iizuka et al. [61] constructed a predictive system using a training dataset of 33 patients, 36% of them belong to the positive class.

Traditional classifiers applied to class-imbalanced datasets often result in suboptimal classification performance [108], because many classifiers assume that the classes are equal in size and some performance metrics reach their maximum value without properly balancing the weight of each class, resulting in a large number of false negatives (misclassifications from the positive class) that are considered more expensive than false positives (misclassifications from the negative class). For this reason, certain measures should be taken to help resolve the problem of class imbalance. Data sampling is the most popular technique for handling class imbalanced data [73], where the dataset is transformed into a more balanced one by adding or removing instances. Another technique used in dealing with class imbalance is the use of cost-sensitive learners [48]. In these learners, costs are assigned to misclassification types (false positive and false negative), and a higher cost is assigned to misclassification of the class of interest (false negative). However, it is difficult to determine in advance the appropriate costs of misclassification.

In the past few years, only few studies investigated feature selection and data sampling together, particularly in the bioinformatics domain. Blagus and Lusa [25] employed three sampling techniques (oversampling, downsizing, and multiple downsizing) as well as variable selection on class imbalanced data. Experiments were conducted on both simulated data and a single DNA microarray dataset using a series of k-NN classifiers along with two linear discriminant classifiers, Random Forest, SVM, CART, a logistic regression based classifier, and prediction analysis of microarrays. The results show that only the k-NN classifiers benefitted from oversampling. One downside of this study, however, was in considering only one possible order of feature selection and sampling. Another downside of this work was in the selection of the datasets. Most of the datasets chosen were simulated datasets that are not the best match for real world datasets.

In 2012, Blagus and Lusa [26] used two data sampling techniques, Random Undersampling and SMOTE, on high-dimensional class-imbalanced breast cancer gene expression datasets and a series of classifiers. The results showed that only the k-NN classifiers seem to benefit substantially from SMOTE and a number of the other classifiers seem to prefer Random Undersampling. One downside of this work however, was in the selection of the datasets. Some of the datasets chosen were not particularly imbalanced, with the minority class being as high as 45% of the instances. In these cases, data sampling will have little effect as the classes are fairly balanced to begin with.

Al-Shahib et al. [16] performed a study using undersampling as well as a wrapper based-feature selection to build classifiers on a dataset containing thirteen protein functional groups. Classifiers were built on the "one vs. all" model, with each classifier deciding if instances are in a given class or not. The study found that the classification performance can be improved by combining data sampling and feature selection along with the SVM classifier and that applying the data sampling to improve the class ratio to 50:50 (with or without feature selection) to that same classifier was significantly better than any of the other combinations with few exceptions. This study considers only one possible order of feature selection and sampling, without examining the importance of this order. Moreover, this paper only investigates a single data sampling technique (i.e. Random Undersampling) while our study uses three: Random Undersampling, Random Oversampling, and SMOTE.

Another factor that can characterize real-world bioinformatics datasets is data noise. There are two types of data noise: attribute noise and class noise. Attribute noise occurs when there are incorrect or missing values in the independent feature, while class noise occurs when an instance is assigned to a wrong class. Several studies have noted the suboptimal performance (e.g., weak performance of classification models, low stability of feature rankers, and extended time of analysis) as a result of low quality data. Therefore, it is necessary to handle low quality data before further analysis.

In classification problems, where the main concern is to have an accurate inductive model, many efforts have been spent on noise tolerant inductive learners. C4.5 decision tree learner [91], for example, uses a tree pruning step intended to remove statistically insignificant portions of the induced decision trees. Similarly, several boosting methods have been developed for being robust to class noise, either by ignoring potentially noisy instances [66] or by preventing extreme increases in instance weights [84]. A comprehensive survey on the different types of label noise, their consequences and the algorithms that consider label noise can be found in the work of Frénay and Verleysen [51].

Altidor et al. [19] investigated the impact of class noise on the stability (robustness of outputs in the face of perturbation) of feature ranking techniques using datasets from different application domains. Eleven threshold-based feature rankers were compared, and their chosen feature lists were compared both between clean and noise-injected data and among the multiple runs of noise injection. The authors found that Geometric Mean, Mutual Information, Deviance, Area Under Precision-Recall Curve, and Kolmogorov-Smirnov Statistic were generally the most stable rankers, while Gini Index, Odds Ratio, and Probability Ratio generally show low stability, and a ranker's stability performance tends to improve when learning from large size datasets.

Pechenizkiy et al. [85] analyzed the impact of class noise on supervised learning using datasets from the medical domain. Specifically, different levels of class noise were injected (from 0 to 20% with a 2% step), and then feature extraction and classification techniques were used to build models. The results showed that feature extraction ameliorated the effects of noise injection, and that some learners (such as Naïve Bayes and C4.5) were less tolerant to noise than others (such as k-NN).

To the best of our knowledge this is the first study to compare three approaches for utilizing feature selection and data sampling using real-world bioinformatics datasets that exhibit both high dimensionality and class imbalance in the context of data quality.

## 8.4 METHODOLOGY

While bioinformatics datasets are commonly characterized by their high dimensionality and class imbalance, most studies have focused on high dimensionality but have completely ignored the class imbalance. In this work we investigate three approaches that are used to deal with both high dimensionality and class imbalance. All approaches combine feature selection and data sampling; the difference between one approach and another is the order (whether sampling takes place before or after feature selection) and the dataset (original or sampled) used for classification. The three approaches are discussed in more detail in Section 2.6.

In this study we employ ten feature ranking techniques (considering four choices for feature subset size: 25, 50, 100, and 200) from three different families: Gini Index (GI), Kolmogorov-Smirnov statistic (KS), Mutual Information (MI), Probability Ratio (PR), Area Under the Receiver Operating Characteristic Curve (ROC), Area Under the Precision Recall Curve (PRC), Signal-to-Noise Ratio (S2N), Wilcoxon Rank Sum (WRS), Significance Analysis of Microarrays (SAM), and Chi Squared (CS). These feature ranking techniques are all discussed in greater detail in Section 2.4.1. We apply three common sampling techniques (discussed in Section 2.5): Random undersampling (RUS), Random oversampling (ROS), and Synthetic minority oversampling technique (SMOTE). These three techniques have been shown to be effective at improving classification performance in previous research [104]. Additionally, we performed sampling to obtain a balanced class ratio: 50:50 majority:minority.

Six learners (discussed in Section 2.7) were used to build predictive models: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). We used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8. All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively.

## 8.5 EXPERIMENTAL RESULTS

The primary objective of this work is to compare three approaches for utilizing feature selection and data sampling using real-world bioinformatics datasets that exhibit both high dimensionality and class imbalance, in the context of data quality. All experiments were conducted with ten feature rankers from three different families (considering four choices for feature subset size), three sampling techniques were applied (RUS, ROS, and SMOTE), and six learners (NB, MLP, 5-NN, SVM, RF100, and LR), and performance was measured using AUC. To avoid any validity problems related to overfitting we used four runs of five-fold cross-validation to build and test our models, presenting the average values across all folds and runs.

The results are presented in Table 8.1, which considers the different values for the choice of approach, data quality level, and sampling technique (presenting the AUC performance of each combination) while averaging together the results across the learners, feature rankers, and subset sizes). In addition, the columns labeled "Average" present the results averaged across all sampling techniques (within a given data quality level), while the row labeled as "Average" presents the results averaged across all choices of approach. Within each column, **bold** values represent the best-performing choice of approach, and *italics* values represent the worst-performing choice of approach.

In Table 8.1, we see the results broken down for each approach. As can be seen, the best AUC results across all choices of approach and sampling technique for Low-Quality and Average-Quality datasets were obtained when FS-DS was used with RUS, while for High-Quality datasets using FS-DS with RUS was still quite competitive (giving an AUC value less than 0.0025 away from the top-performing combination). When considering the oversampling techniques (ROS and SMOTE) DS-FS-UnSam is the top performing approach regardless of the data quality, while for undersampling (RUS) DS-FS-UnSam is the second best (after FS-DS) except on High-Quality data (where DS-FS-UnSam and FS-DS switch places). Looking at the averages, we find that FS-DS is the best approach across all three sampling techniques on Low-Quality datasets, the middle approach on Average-Quality datasets, and the worst approach on High-Quality datasets; DS-FS-UnSam is the best approach on Average-Quality and High-Quality datasets. This appears to result from FS-DS's poor performance with the oversampling techniques (ROS and SMOTE) on the less-noisy datasets. However, as RUS outperforms both of these sampling techniques on the Average-Quality and High-Quality datasets (based on the average results across all approaches), we can give less weight to these results. Overall, because RUS has the best performance for two data quality levels (and the especially poor results with RUS and DS-FS-Sam on Low-Quality data explains the third), and because FS-DS works best with RUS on the Low-Quality and Average-Quality data (where the differences between results matter most), we recommend the combination of these two strategies when working with somewhat challenging data. For particularly High-Quality data, DS-FS-UnSam with either RUS or ROS is an effective approach.

We also performed an ANalysis Of VAriance (ANOVA) test [23] to find statistically significant patterns in these data. The results (using an $\alpha$ value of 0.05) showed that for all combinations of data quality level and choice of sampling technique, the "choice of approach" factor was significant (e.g., had a $Prob > F$ value of less than 0.05). That is to say, when the data are grouped by this factor, at least two of those groups will have significantly different means.

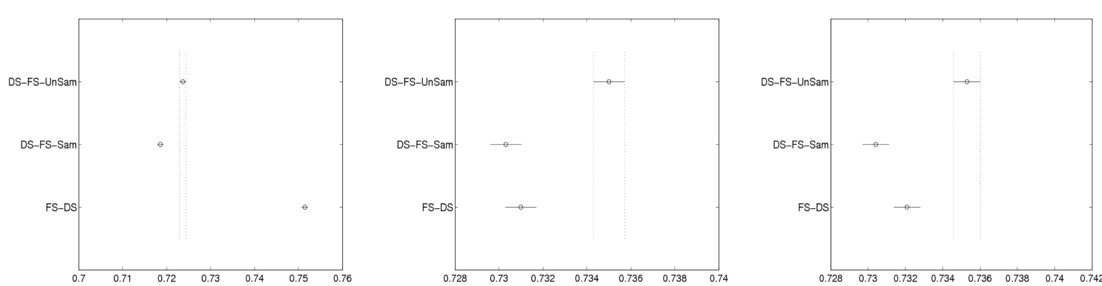
In order to find out which pairs of means (of the three approaches) are significantly different, we conducted a multiple pairwise comparison by using Tukeys Honestly Significant Difference (HSD) criterion [23]. The significance level for Tukeys HSD test is $\alpha = 0.05$. Figures 8.5 through 8.5 show the comparison results of the Low-Quality, Average-Quality, and High-Quality levels, respectively. Each figure contains three subfigures: the one on the left shows the results for RUS, while the one in the middle,

| Approach | Low-Quality | | | | Average-Quality | | | | High-Quality | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RUS | ROS | SMOTE | Average | RUS | ROS | SMOTE | Average | RUS | ROS | SMOTE | Average |
| DS-FS-UnSam | 0.72368 | **0.73500** | **0.735230** | 0.73133 | 0.83694 | **0.84359** | **0.84287** | **0.84113** | **0.93741** | **0.93765** | **0.93560** | **0.93688** |
| DS-FS-Sam | *0.71855* | *0.73031* | *0.73041* | *0.72642* | *0.83423* | 0.83586 | 0.83546 | *0.83519* | *0.93217* | 0.93244 | 0.93119 | 0.93193 |
| FS-DS | **0.75146** | 0.73099 | 0.73209 | **0.73818** | **0.84602** | *0.83506* | *0.83476* | 0.83861 | 0.93524 | *0.93033* | *0.93009* | *0.93189* |
| Average | 0.73123 | 0.73210 | 0.73260 | 0.73198 | 0.83906 | 0.83817 | 0.83770 | 0.83831 | 0.93494 | 0.93347 | 0.93229 | 0.93357 |

**Table 8.1**: Average AUC Values For The Approaches to Combining Feature Selection and Data Sampling

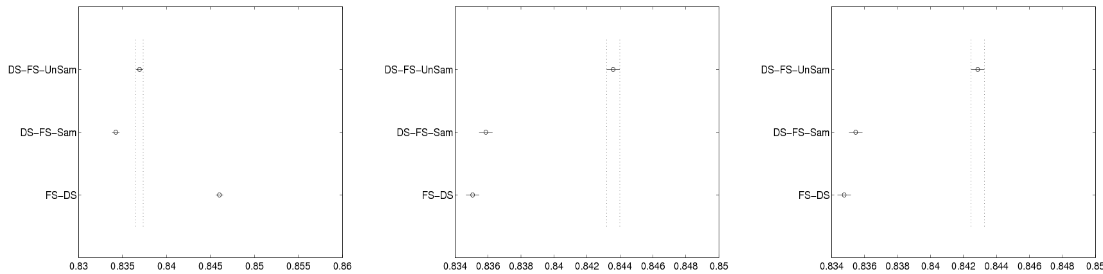
those for ROS, and the one on the right shows the results for SMOTE. The figures display graphs within each group mean represented by a symbol (○) and the 95% confidence interval as a line around the symbol. Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap. We used Matlab to perform the ANOVA and multiple comparison tests.

Figures 8.5 and 8.5 support our conclusion that when using RUS with Low-Quality or Average-Quality data, performing feature selection prior to data sampling (FS-DS) significantly improved classification performance compared performing data sampling prior to feature selection. In addition, we see that RUS is second-best when considering High-Quality datasets. Furthermore, figures 8.5 through 8.5 reveal that for oversampling techniques (ROS, SMOTE), regardless of the data quality, performing data sampling prior to feature selection and then building inductive models using the selected features and the unsampled data (DS-FS-UnSam) is significantly the best performing approach. The classification performance of the three different approaches and their distinctions are clearly depicted in the figures.



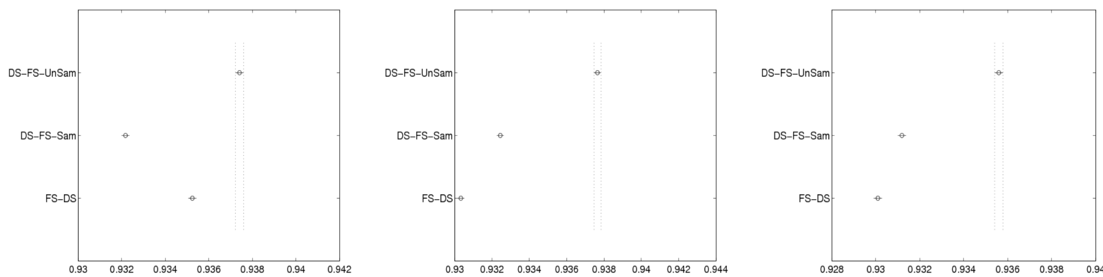

(a) RUS (b) ROS (c) SMOTE

**Figure 8.1**: Tukey's HSD Results: Low-Quality




(a) RUS (b) ROS (c) SMOTE

**Figure 8.2**: Tukey's HSD Results: Average-Quality




(a) RUS (b) ROS (c) SMOTE

**Figure 8.3**: Tukey's HSD Results: High-Quality

### 8.5.1 Robustness of Classification Algorithms and Rankers to Class Noise

The goal of this section is to investigate the robustness of six classifiers and ten feature rankers (considering four choices for feature subset size) to class noise. We only use the recommended approach (i.e. FS-DS) with RUS to investigate this robustness. The results are presented in Tables 8.2 through 8.4. Within each column, **bold** values represent the best AUC value, and *italics* values represent the worst value.

The results in Tables 8.2 and 8.3 demonstrate that RF100 is most frequently the top performing learner regardless of the data quality level, ranker, and subset size for nearly 99.6% of the scenarios. The best AUC across all learner, rankers, and subset size was always obtained when RF100 was used with subset size 200 for High-Quality, Average-Quality, and Low-Quality datasets (0.979167, 0.905144, and 0.793777, respectively). The worst AUC was always obtained when LR was used

| Data Quality | Ranker | NB | | | | MLP | | | | 5NN | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 25 | 50 | 100 | 200 | 25 | 50 | 100 | 200 | 25 | 50 | 100 | 200 |
| High Quality | CS | 0.950706 | 0.950044 | 0.945405 | 0.941463 | 0.946412 | 0.951559 | 0.958141 | 0.963081 | 0.957057 | 0.965200 | 0.969706 | 0.972111 |
| | PR | *0.929128* | *0.923349* | *0.908789* | *0.895432* | *0.933088* | *0.938453* | *0.945297* | *0.951283* | *0.932378* | *0.940611* | *0.945671* | *0.948149* |
| | GI | 0.936862 | 0.933591 | 0.925347 | 0.917075 | 0.937576 | 0.944243 | 0.949602 | 0.956268 | 0.939035 | 0.947058 | 0.951712 | 0.954786 |
| | MI | 0.955240 | 0.952248 | 0.948115 | **0.945373** | 0.949624 | 0.953140 | 0.958292 | 0.962942 | 0.957193 | 0.964667 | 0.969472 | 0.972553 |
| | KS | 0.955460 | 0.953014 | **0.948605** | 0.943190 | 0.952383 | 0.955894 | 0.959478 | 0.963607 | **0.959004** | **0.966572** | **0.970727** | **0.972763** |
| | ROC | **0.955992** | **0.953174** | 0.947832 | 0.942320 | 0.952468 | **0.956871** | **0.960415** | **0.963850** | 0.957556 | 0.964330 | 0.969180 | 0.970184 |
| | PRC | 0.953044 | 0.949630 | 0.943573 | 0.938398 | 0.948914 | 0.953424 | 0.957006 | 0.962329 | 0.955232 | 0.961544 | 0.966647 | 0.969061 |
| | S2N | 0.951124 | 0.948077 | 0.944734 | 0.941928 | 0.948830 | 0.951864 | 0.955857 | 0.962092 | 0.953262 | 0.958945 | 0.963074 | 0.965290 |
| | WRS | 0.954676 | 0.951312 | 0.945573 | 0.940035 | **0.952475** | 0.956570 | 0.960015 | 0.963363 | 0.957446 | 0.963960 | 0.968868 | 0.970328 |
| | SAM | 0.948520 | 0.945165 | 0.940726 | 0.936912 | 0.943363 | 0.946617 | 0.953424 | 0.959639 | 0.952517 | 0.959434 | 0.964466 | 0.967030 |
| Average Quality | CS | 0.868216 | 0.873367 | 0.867995 | 0.856471 | 0.844200 | 0.850294 | 0.858376 | 0.870005 | 0.856439 | 0.868974 | 0.876378 | 0.882934 |
| | PR | *0.835467* | *0.831865* | *0.816617* | *0.796337* | *0.821067* | *0.832483* | *0.842929* | *0.854491* | *0.815090* | *0.826489* | *0.830050* | *0.835035* |
| | GI | 0.836322 | 0.835087 | 0.828386 | 0.816984 | 0.826589 | 0.835035 | 0.845489 | 0.858636 | 0.817500 | 0.831399 | 0.839997 | 0.844865 |
| | MI | 0.870992 | 0.875896 | 0.872435 | 0.865831 | 0.844510 | 0.850781 | 0.861277 | 0.869683 | 0.856238 | 0.870272 | 0.880283 | 0.886230 |
| | KS | 0.875731 | 0.879927 | **0.877206** | **0.867655** | 0.847959 | 0.854259 | 0.864445 | 0.871402 | **0.861930** | **0.876619** | **0.885976** | **0.890566** |
| | ROC | **0.877535** | **0.880650** | 0.874980 | 0.867011 | **0.852964** | 0.858373 | **0.864658** | **0.872814** | 0.860748 | 0.874773 | 0.881465 | 0.885895 |
| | PRC | 0.876713 | 0.877980 | 0.871008 | 0.861496 | 0.849006 | 0.850607 | 0.860684 | 0.869282 | 0.859874 | 0.865301 | 0.873807 | 0.878717 |
| | S2N | 0.874772 | 0.873336 | 0.862353 | 0.855412 | 0.852255 | 0.855390 | 0.859395 | 0.870030 | 0.854779 | 0.864405 | 0.872657 | 0.874042 |
| | WRS | 0.872820 | 0.875539 | 0.868121 | 0.858640 | 0.852620 | **0.858594** | 0.864209 | 0.872028 | 0.860416 | 0.875159 | 0.882311 | 0.886387 |
| | SAM | 0.875180 | 0.875928 | 0.868088 | 0.857287 | 0.843762 | 0.850889 | 0.859285 | 0.868513 | 0.859953 | 0.870464 | 0.875981 | 0.879781 |
| Low Quality | CS | 0.736360 | 0.736724 | 0.726920 | 0.716032 | 0.728515 | 0.738161 | 0.743775 | 0.747262 | 0.732193 | 0.744471 | 0.747542 | 0.748292 |
| | PR | 0.709115 | 0.707864 | *0.700795* | *0.685042* | *0.715860* | *0.727092* | *0.736813* | *0.742838* | *0.694639* | *0.707026* | *0.717857* | *0.721855* |
| | GI | *0.707666* | *0.705963* | 0.704189 | 0.693515 | 0.718520 | 0.731919 | 0.742881 | 0.750158 | 0.700459 | 0.712566 | 0.724300 | 0.730512 |
| | MI | 0.740049 | 0.745101 | 0.745426 | 0.743528 | 0.729202 | 0.740096 | 0.748822 | 0.759811 | 0.733468 | 0.744196 | 0.756879 | 0.761594 |
| | KS | 0.750142 | 0.754068 | 0.753286 | 0.748503 | 0.740669 | 0.745524 | 0.754620 | 0.764826 | 0.743309 | 0.757735 | **0.766286** | **0.770031** |
| | ROC | 0.751836 | **0.759902** | **0.760449** | **0.753659** | **0.744710** | **0.749024** | **0.756086** | **0.765007** | 0.741065 | 0.756991 | 0.763330 | 0.768544 |
| | PRC | 0.750009 | 0.750381 | 0.748862 | 0.742240 | 0.737618 | 0.739155 | 0.749741 | 0.757788 | 0.735715 | 0.747180 | 0.756388 | 0.756973 |
| | S2N | 0.740390 | 0.743115 | 0.743298 | 0.737069 | 0.738361 | 0.743303 | 0.755182 | 0.762580 | 0.741524 | 0.748500 | 0.755240 | 0.759280 |
| | WRS | 0.742389 | 0.747792 | 0.749950 | 0.741552 | 0.743959 | 0.748308 | 0.754829 | 0.762428 | 0.736180 | 0.753059 | 0.763510 | 0.769320 |
| | SAM | **0.754933** | 0.753874 | 0.751316 | 0.746918 | 0.738518 | 0.745666 | 0.753335 | 0.762214 | **0.749465** | **0.758150** | 0.764032 | 0.764337 |

**Table 8.2**: Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality): NB, MLP, and 5NN

with subset size 100 for High-Quality, Average-Quality, and Low-Quality datasets (0.818935, 0.715061, and 0.646862, respectively).

Although no ranker consistently outperforms the others, ROC, KS, and WRS are most frequently the top performing rankers across all of the scenarios (41.7%, 18.1%, and 16.7%, respectively). On the other hand, PR is consistently the worst performing for nearly 95% of the scenarios. The only exceptions to this are when considering subset sizes 25 and 50 with RF100 (Average-Quality Datasets) and NB (Low-Quality Datasets), in the case of these exceptions PR is the second worst. In addition, we observe that 5NN learner shows the best performance when combined with KS ranker. KS was among the top performing rankers for 5NN (at worst second best) across all data quality levels. Thus, we recommend using KS when the 5NN learner is used regardless of the data quality level. Additionally, we see that all rankers (except PR and GI) are particularly robust and are able to ameliorate the difficulty of Low-Quality datasets for all learners except LR. As can be seen, all performance values for Low-Quality datasets (except the values for LR regardless of the ranker, as well

| Data Quality | Ranker | SVM | | | | RF100 | | | | LR | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 25 | 50 | 100 | 200 | 25 | 50 | 100 | 200 | 25 | 50 | 100 | 200 |
| High Quality | CS | 0.938690 | 0.932837 | 0.935356 | 0.940056 | 0.964249 | 0.972325 | 0.976654 | 0.978691 | 0.860531 | 0.842555 | 0.838354 | 0.842475 |
| | PR | *0.929274* | *0.921887* | *0.920660* | *0.928411* | *0.949356* | *0.959364* | *0.965385* | *0.969993* | *0.841770* | *0.820811* | *0.818935* | *0.831245* |
| | GI | 0.933517 | 0.927360 | 0.926717 | 0.934046 | 0.954275 | 0.963559 | 0.969217 | 0.972981 | 0.850165 | 0.830950 | 0.830208 | 0.838243 |
| | MI | 0.941172 | 0.933714 | 0.935841 | 0.939905 | **0.966643** | **0.973272** | 0.976467 | **0.979167** | 0.865045 | 0.844613 | 0.837392 | 0.840510 |
| | KS | 0.944241 | 0.937090 | 0.937533 | 0.941230 | 0.965429 | 0.972451 | 0.976254 | 0.978314 | 0.872274 | 0.854205 | 0.842534 | 0.844918 |
| | ROC | **0.945787** | 0.938601 | **0.939496** | **0.942378** | 0.966530 | 0.972964 | 0.976566 | 0.978675 | 0.875355 | 0.859522 | 0.849665 | **0.848098** |
| | PRC | 0.942462 | 0.936502 | 0.935335 | 0.940687 | 0.964703 | 0.971481 | 0.975443 | 0.977812 | 0.864789 | 0.847279 | 0.840319 | 0.843272 |
| | S2N | 0.943440 | 0.935599 | 0.935529 | 0.939707 | 0.962070 | 0.968649 | 0.973256 | 0.975322 | 0.864524 | 0.845548 | 0.841661 | 0.842641 |
| | WRS | 0.945592 | **0.938922** | 0.939236 | 0.942022 | 0.965962 | 0.973025 | **0.976881** | 0.978241 | **0.875449** | **0.860073** | **0.850722** | 0.847882 |
| | SAM | 0.936681 | 0.928565 | 0.930594 | 0.937370 | 0.960925 | 0.968427 | 0.972848 | 0.974862 | 0.854360 | 0.836786 | 0.832809 | 0.838747 |
| Average Quality | CS | 0.832057 | 0.824262 | 0.823920 | 0.835182 | 0.870380 | 0.887709 | 0.896390 | 0.901407 | 0.746383 | 0.735245 | 0.726280 | 0.735801 |
| | PR | *0.817671* | *0.806379* | *0.806986* | *0.820881* | 0.850381 | 0.870005 | *0.880784* | *0.887886* | *0.724450* | *0.717772* | *0.715061* | *0.729356* |
| | GI | 0.820026 | 0.810230 | 0.812570 | 0.826556 | *0.850106* | *0.869156* | 0.881867 | 0.890674 | 0.733928 | 0.726781 | 0.725985 | 0.737170 |
| | MI | 0.834776 | 0.826262 | 0.826602 | 0.834420 | 0.873238 | 0.888567 | 0.896816 | 0.902266 | 0.744274 | 0.736015 | 0.726576 | 0.732039 |
| | KS | 0.839253 | 0.829298 | 0.828985 | 0.837142 | 0.875056 | 0.890353 | 0.897952 | 0.902642 | 0.754388 | 0.740574 | 0.727855 | 0.733714 |
| | ROC | 0.845824 | 0.832784 | 0.832200 | **0.838823** | 0.878964 | **0.893015** | **0.901189** | **0.905144** | **0.761275** | 0.745509 | 0.730305 | **0.737453** |
| | PRC | 0.843890 | 0.828526 | 0.827616 | 0.834818 | **0.879072** | 0.892777 | 0.900742 | 0.904053 | 0.748602 | 0.734830 | 0.727325 | 0.729879 |
| | S2N | **0.846486** | 0.833215 | 0.827551 | 0.834845 | 0.873381 | 0.887244 | 0.897462 | 0.901492 | 0.755185 | 0.742484 | 0.730605 | 0.731306 |
| | WRS | 0.845061 | **0.834811** | **0.833983** | 0.838420 | 0.877521 | 0.892508 | 0.900839 | 0.904471 | 0.759743 | **0.746001** | 0.733072 | 0.736100 |
| | SAM | 0.835931 | 0.824799 | 0.829208 | 0.836415 | 0.873082 | 0.887917 | 0.896014 | 0.899772 | 0.745790 | 0.737731 | **0.734188** | 0.735819 |
| Low Quality | CS | 0.721585 | 0.719030 | 0.718860 | 0.723905 | 0.741398 | 0.762613 | 0.774479 | 0.778345 | 0.670013 | 0.665833 | 0.652395 | 0.662022 |
| | PR | *0.705848* | *0.709260* | *0.709486* | *0.716093* | *0.723026* | *0.744430* | *0.760782* | *0.768549* | *0.657634* | *0.647663* | *0.646862* | *0.653463* |
| | GI | 0.709272 | 0.712561 | 0.717707 | 0.728937 | 0.724414 | 0.747007 | 0.760892 | 0.770225 | 0.662020 | 0.655109 | 0.656586 | **0.668318** |
| | MI | 0.721588 | 0.719371 | 0.720448 | 0.733420 | 0.746479 | 0.764826 | 0.778321 | 0.785066 | 0.672523 | 0.663758 | 0.652569 | 0.658029 |
| | KS | 0.732593 | 0.725033 | 0.728698 | 0.738401 | 0.752278 | 0.770256 | 0.782645 | 0.790650 | 0.680851 | 0.666520 | 0.661024 | 0.659794 |
| | ROC | **0.735575** | 0.730191 | **0.731986** | 0.738999 | 0.758272 | 0.772903 | 0.785567 | 0.792202 | **0.690974** | 0.671871 | 0.664141 | 0.662710 |
| | PRC | 0.728713 | 0.721135 | 0.726235 | 0.729624 | 0.756155 | 0.773447 | 0.783500 | 0.789066 | 0.675456 | 0.661335 | 0.662059 | 0.657404 |
| | S2N | 0.729775 | **0.727397** | 0.731641 | 0.737580 | 0.754496 | 0.772614 | 0.786219 | 0.793600 | 0.677228 | 0.668218 | **0.668713** | 0.664141 |
| | WRS | 0.732263 | 0.730479 | 0.731870 | **0.740452** | 0.753168 | 0.771880 | 0.784665 | 0.788601 | 0.687499 | **0.672113** | 0.665326 | 0.663265 |
| | SAM | 0.730361 | 0.725985 | 0.729878 | 0.737232 | **0.763075** | **0.777191** | **0.788299** | **0.793777** | 0.677325 | 0.666701 | 0.659808 | 0.662054 |

**Table 8.3**: Average AUC Values For The Three Data Quality Levels (High-Quality, Average-Quality, Low-Quality): SVM, RF100, and LR

as PR and GI values regardless of the learner) are above 0.7 AUC. This is notable because “Average-Quality” was defined as those datasets which have a performance above 0.7 AUC; thus, feature rankers (except PR and GI) were generally able to improve performance of this group of datasets. There was no ranker that was able to ameliorate the difficulty of Low-Quality datasets for LR, where the average AUC remained below 0.7 (i.e. Low-Quality). By the same token, no ranker was able to ameliorate the difficulty of Average-Quality datasets for LR, where the average AUC remained below 0.8 (i.e. Average-Quality).

Looking at the results in terms of subset size we find the unifying trend that as the subset size increases, the performance of MLP, 5NN, and RF100 improves regardless of the ranker being used. The best performance was always obtained when the largest subset size was used (i.e. 200). There was no clear pattern for the other learners when considering the different subset sizes.

The average results for the three factors under study (Learner, Ranker, Subset Size) are presented in Table 8.4. Looking at the average results on a per-learner basis, it can be seen that RF100 constantly outperformed the other learners regardless of all other aspects of our experiments (feature ranker, subset size, and data quality level). This indicates that the RF100 is the is best candidate when learning from bioinformatics datasets that suffer from the two problems (high dimensionality and class imbalance) simultaneously. On the other hand, LR constantly shows the worst performance regardless of all other aspects of our experiments. As for other learners of note, we see that NB is less sensitive to class noise and is able to withstand the impact of class noise. NB was in the middle of the pack for High-Quality datasets, while for Average-Quality and Low-Quality datasets comes in second and fourth(with no significant difference between the third and the fourth), respectively. Additionally, we see that 5NN is among the top performing learners across all levels of data quality (i.e. comes in second or at worst third) giving us confidence that it is a safe choice.

The average results of rankers demonstrate that ROC is the best performing ranker on average in addition to being most frequently the top performing ranker. This gives us confidence that ROC is a good choice for feature selection regardless of data quality level. On the contrary, GI and PR were consistently at the bottom of the pack across all data quality levels. Furthermore, the performance of feature ranking techniques generally deteriorates with decreasing data quality. The decrease in performance with decreasing data quality is generally apparent on all rankers. As for other rankers of note, we see that SAM is less sensitive to class noise. SAM was in the middle of the pack for High-Quality datasets, while for Average-Quality and High-Quality datasets it was among the top performing rankers. Additionally, a general trend that appears is that rankers that perform well (or poorly) in one data quality level perform as well (or as poorly) in the other levels.

The results in terms of subset size show improved performance as feature subset size grows across all data quality levels (except subset size 50 for High-Quality datasets), and the best average performance is always obtained when the largest

| Factor | Choice | High-Quality | Average-Quality | Low-Quality |
|---|---|---|---|---|
| Learner | NB | 0.943052 | 0.871100 | 0.760579 |
| | MLP | 0.953811 | 0.859684 | 0.761287 |
| | 5NN | 0.960901 | 0.870340 | 0.764700 |
| | SVM | 0.937287 | 0.837202 | 0.745810 |
| | RF100 | **0.971533** | **0.893767** | **0.787904** |
| | LR | *0.844870* | *0.744019* | *0.688478* |
| Ranker | CS | 0.937622 | 0.850379 | 0.744903 |
| | PR | 0.915396 | 0.813797 | 0.713058 |
| | GI | *0.915166* | *0.810876* | *0.712385* |
| | MI | 0.940564 | 0.855002 | 0.758525 |
| | KS | 0.944240 | 0.858092 | 0.766820 |
| | ROC | **0.945079** | **0.859145** | **0.770073** |
| | PRC | 0.938038 | 0.850512 | 0.751255 |
| | S2N | 0.940210 | 0.853047 | 0.762210 |
| | WRS | 0.944679 | 0.858503 | 0.768858 |
| | SAM | 0.931427 | 0.850831 | 0.766510 |
| Subset Size | 200 | **0.937585** | **0.850028** | **0.754777** |
| | 100 | 0.935139 | 0.846262 | 0.754112 |
| | 50 | *0.933298* | 0.844687 | 0.750505 |
| | 25 | 0.934947 | *0.843096* | *0.746445* |

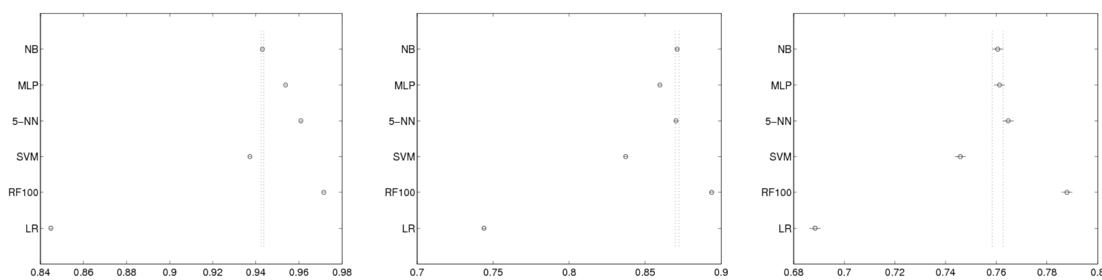
**Table 8.4**: Average AUC Values: Learner, Ranker, and Subset Size

subset size in our experiment (i.e. 200) was used. The worst performance was obtained when the smallest subset size in our experiment (i.e. 25) was used except for High-Quality datasets.

We also performed an ANalysis Of VAriance (ANOVA) test [23] (not shown) to find statistically significant patterns in these data. The results (using an $\alpha$ value of 0.05) showed that three factors (learner, ranker, and subset size) are significant for all data quality levels. That is to say, when the data are grouped by this factor, at least two of those groups will have significantly different means.

Due to this statistical significance, we performed a multiple pairwise comparison by using Tukey's Honestly Significant Difference (HSD) criterion [23]. The significance level for Tukey's HSD test is $\alpha = 0.05$. Figures 8.5.1 through 8.6 show the comparison results of the learners, rankers, and subset sizes factors, respectively. Each figure contains three subfigures: the one on the left shows the results for High-Quality, while the one in the middle, those for Average-Quality, and the one on the right shows the results for Low-Quality.

Figures 8.5.1 through 8.6 support our conclusion that RF100 is statistically better than all other learners, while LR is statistically the worst performing learner across all data quality levels. The figures also show that regardless of the data quality, GI and PR are always showing the worst performance with no significant difference between them. Furthermore, ROC is consistently the best performing rankers for all data quality levels, however, the differences among the top performing rankers are insignificant. Figure 8.6 shows that subset size 200 began significantly better than the other subset sizes for High-Quality and Average-Quality datasets, but not significantly better than subset size 100 for Low-Quality datasets.

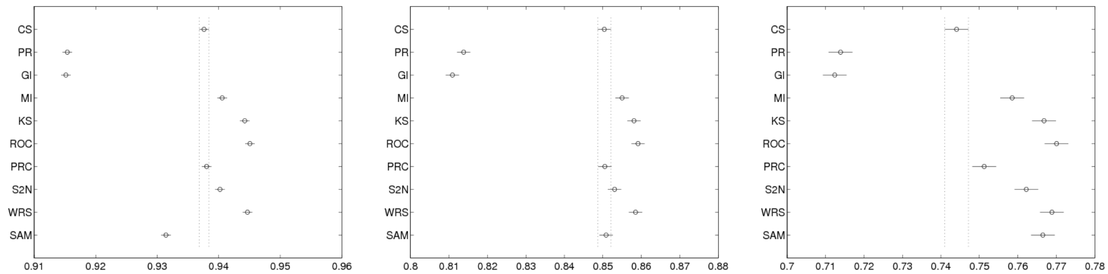



(a) High-Quality (b) Average-Quality (c) Low-Quality

**Figure 8.4**: Tukey's HSD Results: Learners

## 8.6 CHAPTER SUMMARY

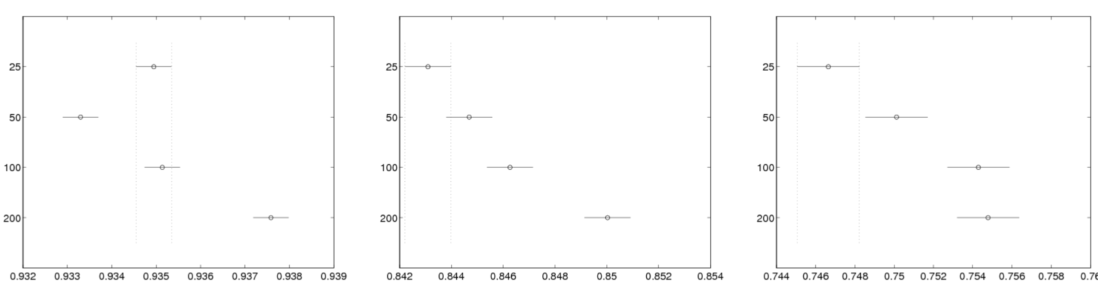
Working with bioinformatics datasets is often challenging due to the high abundance of features and the skewed class distribution. This chapter explored three approaches for utilizing feature selection and data sampling using real-world datasets that exhibit both high dimensionality and class imbalance in the context of data quality. The experiment was carried out on ten high-dimensional, class-imbalanced bioinformatics




(a) High-Quality (b) Average-Quality (c) Low-Quality

**Figure 8.5**: Tukey's HSD Results: Rankers

datasets which was first determined to be noise-free. We injected noise into these datasets to create three levels of data quality (High-Quality, Average-Quality, and Low-Quality), and used ten feature ranking techniques from three families and three sampling techniques. We applied the three approaches (sampling then feature selection and building a learner with the unsampled data (DS-FS-Unsam), sampling then feature selection and built a learner with the sampled data (DS-FS-Sam) and feature selection then sampling and building a learner with the sampled data (FS-DS)) to evaluate their effectiveness in dealing with high dimensionality and class imbalance in the context of data quality. The evaluation was carried out using the area under the ROC curve (AUC) classifier performance metric.




(a) High-Quality (b) Average-Quality (c) Low-Quality

**Figure 8.6**: Tukey's HSD Results: Subset Sizes

The experimental results suggest that when considering RUS sampling technique with Low-Quality and Average-Quality datasets, performing feature selection prior to data sampling (FS-DS) was significantly better than performing data sampling prior to feature selection. For RUS and the High-Quality datasets, FS-DS was the second-best approach. Thus, based on RUS generally giving the best (or close to the best) classification performance while reducing the size of datasets (leading to reduced computation when analyzing such datasets), we recommend using FS-DS with RUS as the approach for combining feature selection and data sampling, especially when dealing with Low-Quality and Average-Quality datasets.

We also investigated the robustness of the classifiers and feature rankers to class noise using only the recommended approach (i.e. FS-DS) with RUS. The results suggest that Random Forest 100 is the best learner across all data quality levels. However, Naïve Bayes, performed well on Low-Quality datasets but less so on the other datasets. This suggests that Naïve Bayes is unusually robust to Low-Quality datasets, and that its performance does not degrade as much as other learners. Logistic Regression, on the other hand, shows the worst performance across the board. As for the rankers, the best across all datasets was generally Area Under the Receiver Operating Characteristic Curve (ROC) and this would be the recommended ranker regardless of the data quality level. However, Significance Analysis of Microarrays, performed well on Low-Quality datasets but not as good on the other datasets. This suggests that Significance Analysis of Microarrays is robust to class noise, and that it does not lose performance on these datasets as much as other rankers. Gini Index and Probability Ratio, on the other hand, were the worst performing rankers across all data quality levels. Furthermore, the results for evaluating the subsets sizes show improved performance as feature subset size grows across all data quality levels, and the best performance is always obtained when the largest subset size in our experiment (i.e. 200) was used. These results allow us to recommend with high confidence the use of RF100 learner with the ROC ranker (using 200 features) for building optimal classification models for bioinformatics data.

# CHAPTER 9

# HOW TO OPTIMALLY COMBINE UNIVARIATE AND MULTIVARIATE FEATURE SELECTION WITH DATA SAMPLING FOR CLASSIFYING NOISY, HIGH DIMENSIONAL, CLASS IMBALANCED DNA MICROARRAY DATA

## 9.1 INTRODUCTION

In the previous chapter (Chapter 8) we utilized one form of filter-based feature selection (i.e. filter-based feature ranking) to compare three approaches developed for classification problems on datasets that exhibit both problems (high dimensionality and class imbalance) simultaneously [15]. In this chapter [11], we compare the same approaches, however, we utilize the three major forms of feature selection: ranker-based techniques, filter-based subset selection, and wrapper-based subset selection. We perform experiments using ten gene expression datasets that were first determined to be relatively free of noise, which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality), and we build our final models using six different classification algorithms.

## 9.2 CONTRIBUTIONS

The primary contribution of this chapter is to provide a comprehensive empirical analysis to give guidance to researchers and practitioners on best practices when classifying bioinformatics data that exhibit both high dimensionality and class imbalance in the context of data noise. In the chapter we: (1) compare three approaches to combining feature selection and data sampling to determine whether the order in which they are applied is important or not; (2) simulate real-world scenarios by injecting class noise into ten real-world gene-expression datasets (after having been determined to be relatively free of noise) creating three data quality tiers (High-Quality, Average-Quality, and Low-Quality); (3) examine three major forms of feature selection techniques (rankers, filter-based subset selection, and wrapper subset selection) and (4) employ six classifiers that are commonly-used in the literature.

The remainder of this chapter will be organized as follows: Section 9.3 presents related works on the topics of high dimensionality, class imbalance, and data noise. Section 9.4 outlines the methods used in this work. In Section 9.5, we present our results. Finally, Section 9.6 concludes this chapter.

## 9.3 RELATED WORK

The overabundance of features makes the process of analyzing bioinformatics datasets more challenging (requiring extensive computation and degrading the predictive performance of inductive models). Feature selection is commonly used for handling high-dimensional data, which tries to choose the best features for performing classification and eliminate redundant and useless features. There are a number of advantages when those redundant and irrelevant features are removed, including: enhanced generalization capability of models, improved model interpretability, and a faster learning process. Accordingly, feature selection has received a lot of attention. A broad survey of feature selection is presented by Guyon and Elisseeff [57], who divide feature selection techniques into two broad categories: filters and wrappers. In 2013, our research group conducted a comprehensive study [41] to investigate the effectiveness of 25 different feature ranking techniques and 6 classification algorithms when predicting patient response to a drug treatment. The result showed that the Random Forest classifier is the best performing classifier regardless of the feature selection being used, and it improved classification performance as feature subset size increased.

In the context of subset-based feature selection, Khoshgoftaar et al. [68] investigated the problem of subset-based selection stability (robustness of outputs in the face of perturbation), including the importance of stability as well as various stability measures. The authors investigated the previous studies on stability analysis of feature subset selection techniques within the domain of bioinformatics and have identified the shortcomings of these works to explore possible opportunities for future work. Wald et al. [68] investigated the stability of two filter based subset selection techniques (Consistency feature subset evaluator and Correlation-Based Feature Selection). They found that Consistency has the greatest stability overall, while Correlation-Based Feature Selection shows moderate stability.

Wrappers, received little attention because they can be very computationally expensive and can result in an overfitted inductive model. Inza et al. [62] compared filter-based feature ranking and wrapper-based subset selection. The authors used six feature ranking techniques along with four choices of learner on two bioinformatics datasets. They showed that wrapper feature selection outperforms filter-based ranking, however, it is computationally more expensive. A comparative study on all three forms of feature selection was conducted by Wang et al. [114]. Experiments were conducted using four filter-based rankers, one filter-based subset evaluator, and three classifiers for both wrapper selection and final classification. The authors found that both subset selection approaches (filter-based and wrapper-based) can give good performance while selecting a smaller subset of features.

Class imbalance occurs when positive class instances (that is, those which belong to the most important class) are outnumbered by instances of the other class(es). Many real-world bioinformatics datasets are characterized by class imbalance [92, 97, 61]. Traditional classifiers applied to class-imbalanced datasets often result in suboptimal classification performance [108]. Data sampling is the most popular technique for handling class imbalanced data [73], where the dataset is transformed into a more balanced one by adding or removing instances.

Relatively, little work focused on both challenges (high dimensionality and class imbalance) together, particularly in the bioinformatics domain. Blagus and Lusa [25] employed three sampling techniques (oversampling, downsizing, and multiple downsizing) as well as variable selection on class imbalanced data. The authors considered only one possible order of feature selection and data sampling (named DS-FS-Sam in this work). In a more recent study, Blagus and Lusa [26] performed a study using data sampling on high-dimensional data. However, some of the datasets used in this were not particularly imbalanced, with the minority class being as high as 45% of the instances. In these cases, data sampling will have little effect as the classes are fairly balanced to begin with. Al-Shahib et al. [16] used undersampling as well as a wrapper based-feature selection to build classifiers to predict protein function from amino acid sequence features. This study only considers one possible order of feature selection and sampling, without examining the importance of this order.

Another challenge encountered when analyzing real-world data is noise. Noise is random error or variance in a measured variable. All kinds of noise can lead to suboptimal classification performance, and class noise has a more harmful effect on classification problems than attribute noise [117]. Unfortunately, many data mining techniques are sensitive to data noise, thus, low quality data can result in suboptimal predictive classification performance and can also impact the effectiveness of feature selection. Therefore, it is important to understand how low quality data can impact data mining techniques (feature selection techniques and classification models). Thus, all empirical investigations presented in this study were performed on data which was first determined to be free of noise and then had artificial class noise added in a controlled fashion. This way, the results can be used to simulate real-world scenarios.

## 9.4 METHODOLOGY

In this chapter we compare the same three approaches discussed in the previous chapter (Chapter 8). These approaches are discussed in more detail in Section 2.6. Instead of examining only ranker-based techniques, we examine all forms of feature selection: ranker-based techniques, filter-based subset selection, and wrapper-based subset selection. In particular, we employ three feature rankers (Chi Squared (CS), Area Under the Receiver Operating Characteristic (ROC) Curve, and Wilcoxon Rank Sum (WRS)) each coupled with three feature subset sizes (25, 50, and 100), a filter-based subset evaluator (Correlation-Based Feature Selection (CFS)), and wrapper-based feature selection using Naïve Bayes inside the wrapper. All feature selection techniques are discussed in more detail in Section 2.4. We apply Random undersampling (RUS) to obtain a balanced class ratio: 50:50 majority:minority. RUS (discussed in Section 2.5) reduces the dataset size, which makes subsequent analysis computationally more efficient compared to oversampling techniques. Additionally, prior research showed its effectiveness [104] and was the recommended sampling technique in the previous chapter (Chapter 8).

Additionally, six classifiers (discussed in Section 2.7) were used to build predictive models: Naïve Bayes (NB), Multilayer Perceptron (MLP), 5-Nearest Neighbor (5NN), Support Vector Machines (SVM), Random Forest with 100 trees (RF100), and Logistic Regression (LR). We used four runs of five-fold cross-validation to build and test our models, and AUC was used as the performance metric. These are discussed in more detail in Section 2.8. All experiments were performed on 10 bioinformatics data (discussed in Section 2.2) which were first determined to be free of noise, and which then had artificial class noise injected in a controlled fashion creating three levels of data quality (High-Quality, Average-Quality, and Low-Quality). Data quality and noise injection are discussed in more detail in Sections 2.1 and 2.3, respectively.

## 9.5 EXPERIMENTAL RESULTS

This study is a comparison of three approaches to utilize both feature selection and data sampling (DS-FS-UnSam, DS-FS-Sam, and FS-DS). The three approaches differ in the order (whether feature selection takes place before or after data sampling) and the dataset (unsampled or sampled) used to build the training dataset. We employed 11 feature selection strategies representing the three major types of feature selection (ranker-based techniques, filter-based subset selection, and wrapper-based feature selection). We utilized random undersampling based on the fact that it is commonly used in the literature and prior research showed its effectiveness (Chapter 8). We created three levels of data quality (High-Quality, Average-Quality, and Low-Quality) by injecting class noise into ten gene expression datasets (relatively free of noise) in a controlled fashion. We build our final models using six commonly-used classification algorithms.

The results are presented in Table 9.1. Each value represents the average AUC performance across four runs of five-fold cross-validation when applying the given combination of feature selection technique, feature-selection/data-sampling strategy, and classifier to the datasets which match that data quality level. In the "Feature Selection Technique" column, the rankers (CS, ROC, and WRS) are followed by a number, which represents the number of features chosen from that ranked list, and the wrapper-based selection approach which uses the NB learner inside the wrapper is abbreviated as "WrapNB" for space considerations. The table includes six sub-tables: one for each classifier (NB, MLP, 5-NN, SVM, RF100, and LR, respectively). The sub-tables also present the average performance (last row of the sub-tables) of each of the approaches over the 11 feature selection strategies and datasets which match that data quality level for that specific learner. The last row of the table represents the overall average performance of each of the approaches for that specific data quality level. The best and worst choices of approach for each combination of learner and

| Learner | Feature Selection Technique | High Quality | | | Average Quality | | | Low Quality | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | DS-FS-UnSam | DS-FS-Sam | FS-DS | DS-FS-UnSam | DS-FS-Sam | FS-DS | DS-FS-UnSam | DS-FS-Sam | FS-DS |
| NB | CS25 | 0.956922 | 0.939667 | 0.955531 | 0.862794 | 0.856155 | 0.885698 | 0.739856 | 0.714112 | 0.755110 |
| | CS50 | 0.959370 | 0.938579 | 0.952183 | 0.871306 | 0.860189 | 0.888606 | 0.742820 | 0.718590 | 0.748761 |
| | CS100 | 0.955167 | 0.934014 | 0.947035 | 0.866736 | 0.855428 | 0.881821 | 0.725347 | 0.712826 | 0.742586 |
| | ROC25 | 0.958692 | 0.946576 | 0.962710 | 0.870029 | 0.867871 | 0.894704 | 0.746643 | 0.722051 | 0.786812 |
| | ROC50 | 0.958067 | 0.943315 | 0.958141 | 0.875069 | 0.872118 | **0.894763** | 0.751881 | 0.731324 | 0.796500 |
| | ROC100 | 0.952748 | 0.937655 | **0.953093** | 0.869952 | 0.866031 | 0.888957 | 0.748590 | 0.734069 | **0.798689** |
| | WRS25 | 0.957530 | 0.945056 | 0.961441 | 0.864825 | 0.862119 | 0.891515 | 0.733065 | 0.711782 | 0.782320 |
| | WRS50 | 0.956223 | 0.941191 | 0.956522 | 0.869192 | 0.866374 | 0.891053 | 0.734966 | 0.720141 | 0.788267 |
| | WRS100 | 0.950764 | 0.934775 | 0.951179 | 0.862180 | 0.858238 | 0.883945 | 0.736548 | 0.723336 | 0.789965 |
| | CFS | 0.937945 | 0.924392 | 0.950497 | 0.811599 | 0.817122 | 0.855433 | 0.669044 | 0.676202 | 0.733283 |
| | WrapNB | 0.835792 | *0.830064* | 0.868756 | 0.711503 | *0.707852* | 0.734158 | 0.596321 | *0.593213* | 0.626226 |
| | Average | 0.944836 | 0.929815 | 0.947846 | 0.851993 | 0.847645 | 0.874833 | 0.722660 | 0.707039 | 0.760637 |
| MLP | CS25 | 0.949837 | 0.940655 | 0.948744 | 0.848082 | 0.834246 | 0.850272 | 0.734229 | 0.707881 | 0.743435 |
| | CS50 | 0.953861 | 0.948902 | 0.951914 | 0.852838 | 0.840810 | 0.857234 | 0.740538 | 0.715005 | 0.758941 |
| | CS100 | 0.960967 | 0.955446 | 0.958011 | 0.858286 | 0.850620 | 0.866224 | 0.746738 | 0.724627 | 0.759960 |
| | ROC25 | 0.953674 | 0.945432 | 0.958299 | 0.853856 | 0.843059 | 0.861977 | 0.745979 | 0.720473 | 0.767679 |
| | ROC50 | 0.958871 | 0.952182 | 0.959560 | 0.858792 | 0.850617 | 0.865710 | 0.750513 | 0.726702 | 0.769858 |
| | ROC100 | 0.961491 | 0.956587 | **0.963166** | 0.867104 | 0.855439 | **0.871431** | 0.750899 | 0.735905 | **0.781452** |
| | WRS25 | 0.954065 | 0.945599 | 0.957761 | 0.854462 | 0.841916 | 0.861483 | 0.745195 | 0.716835 | 0.769848 |
| | WRS50 | 0.958447 | 0.951893 | 0.959371 | 0.858027 | 0.851084 | 0.866670 | 0.749886 | 0.724741 | 0.770297 |
| | WRS100 | 0.961224 | 0.956002 | 0.962819 | 0.867409 | 0.854301 | 0.870917 | 0.750887 | 0.735094 | 0.778505 |
| | CFS | 0.949134 | 0.946058 | 0.961668 | 0.837805 | 0.833801 | 0.864029 | 0.740287 | 0.712072 | 0.749076 |
| | WrapNB | 0.843205 | *0.834045* | 0.880463 | 0.737496 | *0.722565* | 0.763017 | 0.617948 | *0.600128* | 0.635755 |
| | Average | 0.947003 | 0.940450 | 0.951743 | 0.847120 | 0.836561 | 0.856021 | 0.735393 | 0.712320 | 0.754453 |
| 5-NN | CS25 | 0.953704 | 0.956645 | 0.960823 | 0.847929 | 0.851715 | 0.869672 | 0.710428 | 0.734122 | 0.752029 |
| | CS50 | 0.965272 | 0.964150 | 0.966178 | 0.861466 | 0.865179 | 0.880277 | 0.720440 | 0.744309 | 0.768665 |
| | CS100 | 0.970302 | 0.968320 | 0.970497 | 0.872175 | 0.872211 | 0.884749 | 0.726869 | 0.746708 | 0.769050 |
| | ROC25 | 0.953293 | 0.955219 | 0.964155 | 0.848046 | 0.856520 | 0.877677 | 0.712116 | 0.736981 | 0.774097 |
| | ROC50 | 0.962445 | 0.962259 | 0.968287 | 0.863585 | 0.872026 | 0.888707 | 0.733277 | 0.750572 | 0.787125 |
| | ROC100 | 0.968997 | 0.966409 | **0.972136** | 0.874404 | 0.876468 | 0.893524 | 0.741672 | 0.755795 | 0.792522 |
| | WRS25 | 0.953941 | 0.954880 | 0.963517 | 0.847733 | 0.855716 | 0.877799 | 0.705651 | 0.727978 | 0.774909 |
| | WRS50 | 0.961572 | 0.962071 | 0.968237 | 0.864295 | 0.871577 | 0.889605 | 0.726673 | 0.747163 | 0.785342 |
| | WRS100 | 0.968989 | 0.966003 | 0.971613 | 0.876125 | 0.876875 | **0.893934** | 0.741471 | 0.754707 | **0.794353** |
| | CFS | 0.958922 | 0.957368 | 0.970727 | 0.845771 | 0.846970 | 0.876716 | 0.710403 | 0.732188 | 0.752888 |
| | WrapNB | *0.818756* | 0.832314 | 0.876703 | *0.705934* | 0.717608 | 0.759913 | *0.591255* | 0.600066 | 0.639645 |
| | Average | 0.950087 | 0.950831 | 0.960151 | 0.848828 | 0.853811 | 0.874113 | 0.712529 | 0.731759 | 0.764196 |
| SVM | CS25 | 0.946300 | 0.929197 | 0.940573 | 0.831390 | 0.823077 | 0.841703 | 0.720238 | 0.705382 | 0.739134 |
| | CS50 | 0.939195 | 0.927435 | 0.931881 | 0.824639 | 0.816886 | 0.831260 | 0.719086 | 0.702378 | 0.735627 |
| | CS100 | 0.937636 | 0.934473 | 0.933959 | 0.818697 | 0.823840 | 0.829222 | 0.715291 | 0.705441 | 0.735847 |
| | ROC25 | 0.950437 | 0.936142 | **0.950783** | 0.844368 | 0.834184 | **0.858921** | 0.730131 | 0.712864 | **0.763731** |
| | ROC50 | 0.942167 | 0.932891 | 0.940745 | 0.829304 | 0.824617 | 0.844431 | 0.721214 | 0.715188 | 0.754172 |
| | ROC100 | 0.939026 | 0.936861 | 0.942601 | 0.826409 | 0.827039 | 0.843152 | 0.719614 | 0.716623 | 0.759721 |
| | WRS25 | 0.950232 | 0.935980 | 0.950566 | 0.842009 | 0.834840 | 0.858333 | 0.723477 | 0.709912 | 0.763399 |
| | WRS50 | 0.942757 | 0.933448 | 0.940561 | 0.830201 | 0.827787 | 0.846444 | 0.720455 | 0.714818 | 0.756164 |
| | WRS100 | 0.938757 | 0.936113 | 0.942837 | 0.828340 | 0.828942 | 0.844667 | 0.718393 | 0.718234 | 0.758983 |
| | CFS | 0.927253 | 0.929551 | 0.944706 | 0.808067 | 0.811775 | 0.836028 | 0.717858 | 0.713123 | 0.738035 |
| | WrapNB | *0.817936* | 0.836037 | 0.885095 | *0.712097* | 0.721493 | 0.765021 | *0.582856* | 0.604834 | 0.641082 |
| | Average | 0.931444 | 0.925307 | 0.937245 | 0.819987 | 0.817750 | 0.837661 | 0.709616 | 0.702876 | 0.741618 |
| RF100 | CS25 | 0.969187 | 0.955251 | 0.968309 | 0.874662 | 0.855336 | 0.881142 | 0.744117 | 0.713799 | 0.766278 |
| | CS50 | 0.976990 | 0.965916 | 0.974070 | 0.894274 | 0.872044 | 0.896810 | 0.768187 | 0.734437 | 0.785214 |
| | CS100 | **0.981394** | 0.971171 | 0.977398 | 0.904710 | 0.881168 | 0.903292 | 0.781599 | 0.752352 | 0.789484 |
| | ROC25 | 0.969483 | 0.958572 | 0.971536 | 0.880436 | 0.864444 | 0.892014 | 0.756735 | 0.733108 | 0.784975 |
| | ROC50 | 0.976401 | 0.966331 | 0.976160 | 0.897804 | 0.879363 | 0.901876 | 0.772071 | 0.749486 | 0.797153 |
| | ROC100 | 0.979796 | 0.971035 | 0.978867 | 0.908131 | 0.886637 | **0.908800** | 0.788014 | 0.759363 | 0.809325 |
| | WRS25 | 0.969461 | 0.957710 | 0.970715 | 0.879174 | 0.862433 | 0.890956 | 0.748925 | 0.726660 | 0.783919 |
| | WRS50 | 0.976692 | 0.966586 | 0.975798 | 0.896891 | 0.878118 | 0.902515 | 0.771123 | 0.745215 | 0.799303 |
| | WRS100 | 0.980399 | 0.971477 | 0.978767 | 0.906916 | 0.887170 | 0.908432 | 0.786473 | 0.757678 | **0.809843** |
| | CFS | 0.974927 | 0.963818 | 0.978352 | 0.887516 | 0.862464 | 0.890485 | 0.761195 | 0.731640 | 0.754519 |
| | WrapNB | *0.832597* | 0.838633 | 0.894984 | *0.718018* | 0.724336 | 0.774074 | *0.594230* | 0.600662 | 0.628792 |
| | Average | 0.963800 | 0.954486 | 0.968419 | 0.879991 | 0.861942 | 0.888475 | 0.754062 | 0.729319 | 0.775265 |
| LR | CS25 | 0.866455 | 0.868877 | 0.846261 | 0.743642 | 0.751385 | 0.744122 | 0.655100 | 0.669714 | 0.685226 |
| | CS50 | 0.840325 | 0.853312 | 0.834029 | 0.719469 | 0.745230 | 0.741035 | 0.643623 | 0.664317 | 0.689559 |
| | CS100 | 0.830151 | 0.849953 | 0.834958 | 0.707030 | 0.736769 | 0.735040 | 0.627445 | 0.647898 | 0.681842 |
| | ROC25 | 0.872804 | 0.878459 | 0.874804 | 0.748057 | 0.769015 | 0.766753 | 0.670578 | 0.687764 | 0.714578 |
| | ROC50 | 0.848052 | 0.868016 | 0.862499 | 0.722330 | 0.758921 | 0.755277 | 0.636923 | 0.674371 | 0.704317 |
| | ROC100 | 0.831921 | 0.860648 | 0.856425 | 0.700410 | 0.744720 | 0.745784 | 0.631843 | 0.662669 | 0.697913 |
| | WRS25 | 0.871226 | 0.878883 | 0.876238 | 0.745140 | 0.767975 | 0.766116 | 0.659711 | 0.685611 | 0.717176 |
| | WRS50 | 0.848156 | 0.869280 | 0.862782 | 0.721510 | 0.759055 | 0.757439 | 0.635926 | 0.674705 | 0.705709 |
| | WRS100 | 0.833725 | 0.860085 | 0.858356 | 0.705496 | 0.746030 | 0.747689 | 0.630497 | 0.666091 | 0.699391 |
| | CFS | 0.848489 | 0.900733 | **0.902570** | 0.715437 | 0.796670 | **0.802972** | 0.661949 | 0.696776 | **0.721932** |
| | WrapNB | 0.840103 | *0.824609* | 0.863791 | 0.728094 | *0.707829* | 0.737880 | 0.608909 | *0.597538* | 0.630244 |
| | Average | 0.848400 | 0.864853 | 0.860662 | 0.723389 | 0.753085 | 0.753947 | 0.642275 | 0.666718 | 0.695669 |
| Overall | Average | 0.930928 | *0.927624* | **0.937678** | 0.828551 | *0.828466* | **0.847508** | 0.712755 | *0.708339* | **0.748640** |

**Table 9.1**: Average AUC Values

data quality are printed in **bold** and *italics*, respectively

From the results we can make the general statement that FS-DS is the best approach to utilize feature selection and data sampling when learning from class imbalanced, high dimensional bioinformatics datasets. The overall average performance shows that FS-DS is the best performing approach across the board (regardless of the data quality level). When we look at the "Average" row in each sub-table showing the performance across all feature selection strategies, we find that FS-DS is the best performing approach for all combinations of data quality tiers and learners (except High-Quality with LR). The other two approaches did not perform as well: DS-FS-UnSam was in the middle of the pack on average; for NB, MLP, SVM, and RF100 it was the second best, and was the worst when considering the other learners (5-NN and LR), while DS-FS-Sam was the worst performing approach on average. Furthermore, FS-DS showed itself to be particularly noise tolerant by not being at the bottom of the pack for Average-Quality and Low-Quality data (higher levels of noise), where it was never the worst performing approach and at worst comes in second place. When considering High-Quality data, FS-DS was the worst performing approach for only 3 of the 66 combinations (SVM learner and the CS ranker with 25 features, and the LR learner with the CS ranker utilizing 25 and 50 features).

Looking closely at these results in terms of the different feature selection techniques, it can be seen that for all subset selection techniques (CFS and Wrapper), the best performing approach was consistently FS-DS regardless of the learner and data quality level. The only exception to this is when considering Low-Quality data with the RF100 learner, where FS-DS was the second best. When considering the other category of feature selection (i.e. rankers), we can see that for all but 8 out of 108 combinations of learner and ranker with both Average-Quality and Low-Quality data (Average-Quality with the RF100 learner and CS ranker with 100 features or LR learner and all rankers utilizing 25 and 50 features as well as CS with 100 features), the best approach was FS-DS. This is especially important as these two tiers of data quality represent higher levels of noise. When considering High-Quality data, FS-DS was at the top of the pack for 39 out of 66 combinations. On the other hand, DS-FS-UnSam was the best choice for 20 of 198 combinations, and was the worst for 66 combinations, while DS-FS-Sam was only the best approach for 16 combinations and was at the bottom for 129 of the 198 combinations.

Looking at these results on a per-data quality level basis, we see that FS-DS is particularly robust and is able to improve the classification performance for all learners regardless of the feature selection technique when Low-Quality datasets (AUC less than 0.7 due to noise injection) are used. In particular, FS-DS improved the performance of classifiers enough to result in AUC values greater than 0.7 (which is our metric for Average-Quality) for all combinations of learner and feature selection, except for Wrapper regardless of the learner, and LR with all rankers utilizing 100 features as well as CS with 25 and 50 features. Additionally, it should be noted that FS-DS was the only approach that was able to improve the classification performance for LR (when combined with ROC25, ROC50, WRS25, WRS50, and CFS feature selection), resulting in AUC values greater than 0.7. FS-DS combined with the RF100 learner helped improve the classification performance on Low-Quality datasets significantly (when combined with ROC100 or WRS100), resulting in AUC values greater than 0.8 (i.e. our metric for High-Quality). By the same token, FS-DS and RF100 improved the performance on Average-Quality datasets, achieving AUC values greater than 0.9.

### 9.5.1 Statistical Tests

We performed a set of one-factor ANalysis Of VAriance (ANOVA) tests [23] to validate the classification results and find statistically significant patterns. The ANOVA analysis and subsequent statistical tests were performed within MATLAB. Since a significance factor of 5% was chosen, the $Prob > F$ value must be less than this value (i.e. 0.05) for the result to be significant.

In this analysis, we considered only one factor: the choice of strategy for combining feature selection and data sampling, with three different levels of this factor (DS-FS-UnSam, DS-FS-Sam, and FS-DS). The tests performed were across all datasets and factors, and for each level of data quality. For the ANOVA tests, the AUC results across all six learners were used as the response variable. The results are presented in Table 9.2. These results show that the choice of approach for combining feature selection and data sampling is significant across all data quality levels as well as each level of data quality; that is to say, when the data are grouped by the choice of approach, at least two of those groups will have significantly different means.

We wanted to find out which pairs of means are significantly different, and which are not. We conducted a multiple pairwise comparison by using Tukey's Honestly Significant Difference (HSD) criterion [23]. The significance level for Tukey's HSD test is $\alpha = 0.05$. Figure 9.6 shows the comparison results of the three choices of approach for combining feature selection and data sampling for all data quality levels, and for each of the different levels of data quality; The results for all datasets, High-Quality datasets only, Average-Quality datasets only, and Low-Quality datasets only are shown in Figures 9.1(a), 9.1(b), 9.1(c), and 9.1(d), respectively. The figures display graphs within each group mean represented by a symbol ($\circ$) and the 95% confidence interval as a line around the symbol. Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap.

Figure 9.6 supports our conclusion that the top performing choice of approach for combining feature selection and data sampling is always FS-DS. The difference between the top performing approach and the other approaches (i.e. DS-FS-UnSam and DS-FS-Sam) is statistically significant across all data quality levels. Furthermore, DS-FS-Sam was significantly the worst performing approach across all data

| Datasets | Source | Sum Sq. | d.f. | Mean Sq. | F | Prob>F |
|---|---|---|---|---|---|---|
| All Data Quality Levels | FS/DS Strategy | 50.4 | 2 | 25.1952 | 1073.41 | 0 |
| | Error | 21727.6 | 925677 | 0.0235 | | |
| | Total | 21777.9 | 925679 | | | |
| High Quality | FS/DS Strategy | 9.56 | 2 | 4.78028 | 437.57 | 1.31E-190 |
| | Error | 5966.15 | 546117 | 0.01092 | | |
| | Total | 5975.71 | 546119 | | | |
| Average Quality | FS/DS Strategy | 19.58 | 2 | 9.79016 | 424.17 | 1.27E-184 |
| | Error | 5633.49 | 244077 | 0.02308 | | |
| | Total | 5653.07 | 244079 | | | |
| Low Quality | FS/DS Strategy | 44.16 | 2 | 22.0823 | 621.47 | 2.14E-269 |
| | Error | 4813.84 | 135477 | 0.0355 | | |
| | Total | 4858.01 | 135479 | | | |

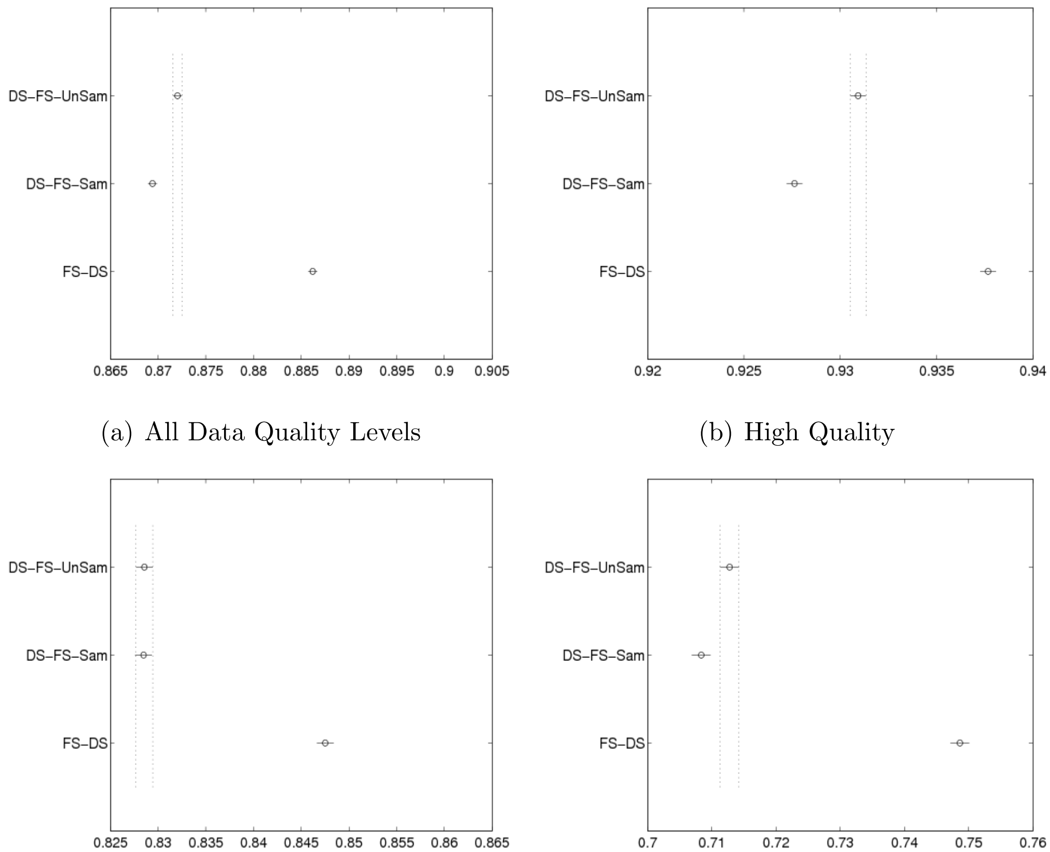
**Table 9.2**: ANOVA Results: Feature-Selection/Data-Sampling Strategies Across All Learners

quality levels, except when considering Average-Quality datasets, where the difference is statistically insignificant. DS-FS-UnSam, on the other hand, shows average performance on High-Quality and Low-Quality datasets, while being second worst on Average-Quality datasets but not statistically distinguishable.

## 9.6 CHAPTER SUMMARY

While many studies investigated feature selection and data sampling in bioinformatics, utilizing them in conjunction has received little attention. In this work, we compare three approaches to combine feature selection and data sampling (DS-FS-UnSam, DS-FS-Sam, and FS-DS). We employed three major forms of feature selection (feature ranking, filter-based subset selection, and wrapper-based feature selection) as well as a commonly used data sampling technique. We created three categories of datasets (High-Quality, Average-Quality, and Low-Quality) by injecting artificial class noise in a controlled fashion into ten gene expression datasets which were first determined to be relatively free of noise. We build our final models using six different classification algorithms.

The experimental results demonstrate that paying attention to the order when uti-




(a) All Data Quality Levels

(b) High Quality

(c) Average Quality

(d) Low Quality

**Figure 9.1**: Tukey's HSD Results: Feature-Selection/Data-Sampling Strategies Across All Learners

lizing both feature selection and data sampling and the dataset (whether unsampled or sampled) used for classification is extremely important in improving the performance of classification algorithms. We found that the best order to apply feature selection and data sampling is to employ feature selection followed by data sampling (i.e. FS-DS). This approach significantly improved the performance of all classifiers compared to the other approaches. On the other hand, DS-FS-Sam was the worst performing approach, on average, regardless of the data quality level. All of these results are supported by ANOVA and Tukey's Honestly Significant Difference statistical tests. Thus, we recommend FS-DS as the approach when learning from class imbalanced high dimensional bioinformatics datasets, regardless of any implication of noise or the classification algorithm that is going to be used. In particular, we recommend using FS-DS with feature rankers (especially ROC and WRS utilized with 100 features), as they showed superior classification performance compared to subset-based feature selection techniques.

# CHAPTER 10

# CONCLUSION AND FUTURE WORK

In bioinformatics, the huge volumes of biological data make manual human analysis impractical. Thus, methods from the domain of machine learning are designed to handle very large datasets, and can be used to extract interesting, important, and hidden knowledge. Gene expression datasets are of great importance in cancer diagnosis and drug discovery. One particular challenge in handling gene expression datasets is noise. The presence of noise in gene expression datasets is inevitable and can have a harmful effect on machine learning. Additionally, many real-world gene expression datasets suffer from the combined problem of high dimensionality and class imbalance. High dimensionality occurs because there are large numbers of genes which can result in extensive computation, redundant features, irrelevant features, and suboptimal predictive performance. The unequal distribution of instances between classes is commonly known as class imbalance. This is a challenge for traditional classification algorithms because they typically attempt to maximize some performance metrics without regard to the significance of different classes, possibly favoring the majority class at the expense of the minority class. A variety of approaches have been described to counter these two challenges (i.e. high dimensionality and class imbalance), but little work has dealt with both challenges simultaneously (especially in bioinformatics), let alone considered the problem of data noise.

This research focuses on the analysis of machine learning algorithms (classification algorithms, feature selection techniques, and data sampling techniques) to assess their robustness to class noise, and give practitioners guidance on best practices when classifying bioinformatics data that exhibit both high dimensionality and class imbalance in the context of data noise. In the course of this work, we performed many studies, each with its own set of experiments from which many conclusions were drawn. In the sections that follow, conclusions are briefly stated and suggestions for future work are provided.

## 10.1 CONCLUSION

Class noise can adversely impact machine learning techniques. In chapter 3 we evaluated the robustness of ten filter-based feature selection techniques and six classification algorithms, examining the impact of data noise on classification performance. The empirical results showed that CS, MI, KS, ROC, WRS, and SAM are particularly robust and are able to ameliorate the difficulty of Low-Quality datasets for all learners except LR. Among the learners investigated RF100 was the least sensitive to class noise and a good candidate for classification across all rankers and data quality levels, while LR showed the worst performance across all rankers and data quality levels. Finally, there was no ranker that was able to ameliorate the difficulty of Low-Quality and Average-Low datasets for LR.

Subset feature selection techniques have the advantage of selecting unique features (features not highly correlated with other features in the selected set) compared to ranker-based feature selection techniques (which can only target relevancy rather than redundancy). In this chapter 4, we investigated the robustness of subset-based feature selection (both filter-based subset selection and wrapper-based subset selection) using high dimensional bioinformatics datasets in the context of data quality. The experiment demonstrated that CFS outperforms both Consistency feature selection and wrapper-based feature selection across all data quality levels and learners. Furthermore, the results demonstrate that the effectiveness of subset-based feature selection when learning from noisy high dimensional datasets depends on the choice of learner. However, because of the other benefits gained when employing subset-based feature selection (such as finding the most important features which are unique and not highly correlated with other features), we recommend exploring the use of subset-based feature selection (especially CFS) for bioinformatics, specifically if feature reduction (and elimination of redundant features) is more important than raw classification performance.

In chapter 5, we provide, to the best of our knowledge, the first study to compare three major categories of feature selection (feature rankers, filter-based subset selection , and wrapper-based subset selection) when learning from bioinformatics datasets with varying levels of data quality due to noise injection. The experimental results showed that feature rankers outperform the two subset-based approaches for all combinations of learner and data quality level (except High-Quality data with the LR learner). Although CFS (an example of filter-based subset selection) performance was competitive, it was always outperformed by a feature ranker. Wrapper-based subset selection, on the other hand, was significantly the worst performing approach across all data quality levels. All of this analysis was confirmed through ANOVA and Tukey's HSD tests. Thus, based on these results and the fact that feature rankers are computationally much less expensive than the two subset-based approaches, we recommend using feature rankers to reduce high dimensionality when analyzing bioinformatics datasets, regardless of the data quality level (i.e. noise level). In particular, the ROC and WRS rankers are less sensitive to class noise and are good choices for feature selection.

In the context of learning from balanced bioinformatics datasets with varying levels of data quality, we investigate the effectiveness of ensemble classification techniques in chapter 6. We compared three forms of ensemble classification techniques (Select-Bagging, Select-Boosting, and Random Forest), as well as feature selection followed by single classifiers, using ten high-dimensional and balanced gene expression datasets with varying levels of data quality. We employ three feature rankers (with three feature subset sizes) along with Correlation Based Feature selection to alleviate the high-dimensionality, and we utilize five classification algorithms to build the classification models. The experimental results demonstrated that the best strategy to improve the classification performance for models built with balanced bioinformatics datasets is an ensemble classification technique. The results showed the best performing approaches are consistently RF100 and Select-Bagging regardless of the data quality level. These two approaches significantly improved the classification performance compared to no ensemble classification regardless of the data quality level. Thus, we recommend using ensemble classification techniques when learning from balanced high-dimensional bioinformatics datasets regardless of any implication of noise. In particular, we recommend using RF100 as it is significantly the top performing classification approach, on average, regardless of the data quality level. Additionally, RF100 does not rely on the choice of the base classifier, unlike the other ensemble approaches.

Class imbalance is a frequent problem within bioinformatics datasets. Addressing this problem, however, has received little attention in this domain. Data sampling is the most common way to deal with this problem. In chapter 7 we examine the importance of alleviating class imbalance for classification problems on bioinformatics datasets. We compared two approaches (FS and FS-DS). The difference between the two approaches is whether we apply data sampling (commonly used technique to alleviate class imbalance) or not. Our results showed that feature selection alone does not perform as well as when we incorporate data sampling as well. The approach which utilizes data sampling (i.e. FS-DS) was most frequently the top performing approach (regardless of the data quality level). We also performed a series of z-tests and found that the difference between the two approaches is statistically significant across all factors and for each level of data quality. For these reasons we recommend alleviating the class imbalance (e.g. by applying data sampling) to achieve improved classification performance for bioinformatics classification problems.

Many bioinformatics datasets are characterized by their high dimensionality and class imbalance. However, addressing these two problem simultaneously has not had enough attention. In chapter 8, we explored three approaches for addressing these two problems simultaneously. All three approaches utilize feature selection and data sampling. The difference between one approach and another is based on two main questions, whether feature selection or data sampling should come first and whether to use original or sampled data to build the training dataset. The experimental results suggest that when feature selection was performed prior to data sampling (FS-DS) it significantly improved classification performance compared to the cases where data sampling was performed prior to feature selection when considering Random Undersampling (RUS) sampling technique with Low-Quality and Average-Quality datasets, and second best approach when considering RUS with High-Quality datasets. Thus, based on the general performance and the fact that RUS reduces the size of datasets leading to reduced computation when analyzing such datasets we recommend using FS-DS with RUS as the approach for combining feature selection and data sampling, especially when dealing with Low-Quality and Average-Quality datasets.

We also investigated the robustness of the classifiers and feature rankers to class noise using only the recommended approach (i.e. FS-DS) with RUS. The results suggest that Random Forest 100 is the best learner across all data quality levels. Logistic Regression, on the other hand, showed the worst performance across the board. As for the rankers, the best across all datasets was generally Area Under the Receiver Operating Characteristic Curve (ROC) and this would be the recommended ranker regardless of the data quality level. Gini Index and Probability Ratio, on the other hand, were the worst performing rankers across all data quality levels. Furthermore, the results for evaluating the subsets sizes show improved performance as feature subset size grows across all data quality levels, and the best performance is always obtained when the largest subset size in our experiment (i.e. 200) was used. All of this analysis was confirmed through ANOVA and Tukey's HSD tests. These results allow us to recommend with high confidence the use of the RF100 learner with the ROC ranker (using 200 features) for building optimal classification models for bioinformatics data.

In this chapter 9, we compare the same approaches discussed in chapter 8, and instead we utilize the three major forms of feature selection: ranker-based techniques, filter-based subset selection, and wrapper-based subset selection. Our results support our conclusions in the previous chapter, that paying attention to the order when utilizing both feature selection and data sampling and the dataset (whether unsampled or sampled) used for classification is extremely important in improving the performance of classification algorithms, with the best order to apply feature selection and data sampling is to employ feature selection followed by data sampling (i.e. FS-DS). This approach significantly improved the performance of all classifiers compared to the other approaches. We confirmed all of these results using ANOVA and Tukey's Honestly Significant Difference statistical tests. Thus, we recommend FS-DS as the approach when learning from class imbalanced high dimensional bioinformatics datasets, regardless of any implication of noise or the classification algorithm that is going to be used. In particular, we recommend using FS-DS with feature rankers, as they showed superior classification performance compared to subset-based feature selection techniques.

Overall, we examine and compare different machine learning techniques and demonstrate their effectiveness in the context of learning from bioinformatics datasets with varying levels of data quality due to noise injection. Based on these experiments, our recommendations for achieving optimal classification performance would depend on the characteristics of the dataset being analyzed. For high-dimensional bioinformatics data, feature rankers are the best strategy to reduce high dimensionality, regardless of the data quality level (i.e. noise level). In particular, the ROC and WRS rankers are less sensitive to class noise and are good choices for feature selection. If learning from balanced bioinformatics datasets, the best strategy to improve the classification performance is an ensemble classification technique. In particular, RF100 and Select-Bagging were consistently the best performing approaches. If learning from class-imbalanced high-dimensional bioinformatics datasets, the best strategy to significantly improve classification performance is to perform feature selection prior to data sampling (considering Random Undersampling).

## 10.2 FUTURE WORK

Opportunities for future work are listed below:

- In all of these experiments, only the impact of class noise on machine learning algorithms has been studied. Future research, may involve studying the robustness of machine learning algorithms to attribute noise.

- In all of these experiments, only cancer gene classification datasets were considered. Future research may involve using datasets for more specific purposes (e.g. patient response datasets).

- While we considered only binary classification in all of these experiments, future research may consider multi-class classification.

- Finally, while exploring solutions to the combined problem of high dimensionality and class imbalance, only one family of preprocessing techniques for addressing class imbalance was considered. Other methods for addressing class imbalance could be the object of future work (e.g. cost-sensitive learning).

## 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, 2010.

[2] A. Abu Shanab, T. M. Khoshgoftaar, and R. Wald. Impact of noise and data sampling on stability of feature selection. In *Machine Learning and Applications and Workshops (ICMLA), 2011 10th International Conference on*, volume 1, pages 172–177, Dec 2011.

[3] A. Abu Shanab, T. M. Khoshgoftaar, and R. Wald. Robustness of threshold-based feature rankers with data sampling on noisy and imbalanced data. In *Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference*, pages 92–97, 2012.

[4] A. Abu Shanab, T. M. Khoshgoftaar, and R. Wald. Evaluation of wrapper-based feature selection using hard, moderate, and easy bioinformatics data. In *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*, pages 149–155, Nov 2014.

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

[6] 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 *IEEE 13th International Conference on Information Reuse and Integration (IRI)*, pages 415–422. IEEE, August 2012.

[7] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. How ranker and learner choice affects classification performance on noisy bioinformatics data. In *2014 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 277–282, August 2014.

[8] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Comparing feature ranking, filter-based feature subset selection, and wrapper-based feature subset selection for classification of noisy bioinformatics data. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, September 2014, In Submission.

[9] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Comparison of three approaches for combining feature selection and data sampling using hard, moderate, and easy bioinformatics data. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, March 2014, In Submission.

[10] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Filter-based subset selection for easy, moderate, and hard bioinformatics data. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, July 2014, In Submission.

[11] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. How to optimally combine univariate and multivariate feature selection with data sampling for classifying noisy, high dimensional, class imbalanced dna microarray data. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, October 2014, In Submission.

[12] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Is gene selection enough for imbalanced bioinformatics data? Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, November 2014, In Submission.

[13] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Observing the effects of artificially-injected difficulty on the process of building classification models for bioinformatics data. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, April 2014, In Submission.

[14] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and A. Napolitano. Ensemble classification performance on balanced bioinformatics data with varying levels of data quality. Technical report, Department of Computer and Electrical Engineering and Computer Science, Florida Atlantic University, January 2015, In Submission.

[15] A. Abu Shanab, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse. Comparison of approaches to alleviate problems with high-dimensional and class-imbalanced data. In *2011 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 234–239, August 2011.

[16] 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.

[17] 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 of the United States of America*, 96(12):6745–6750, 1999.

[18] W. Altidor, T. M. Khoshgoftaar, and A. Napolitano. Wrapper-based feature ranking for software engineering metrics. In *International Conference on Machine Learning and Applications (ICMLA)*, pages 241–246, dec. 2009.

[19] W. Altidor, T. M. Khoshgoftaar, and A. Napolitano. A noise-based stability evaluation of threshold-based feature selection techniques. In *2011 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 240–245, August 2011.

[20] 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.

[21] D. G. Beer, S. L. R. 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. G. Taylor, M. D. Iannettoni, M. B. Orringer, and S. Hanash. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nat Med*, 8(8):816–824, Aug 2002.

[22] 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.

[23] M. L. Berenson, D. M. Levine, and M. Goldstein. *Intermediate Statistical Methods and Applications: A Computer Package Approach*. Prentice-Hall, Englewood Cliffs, New Jersey, 1983.

[24] M. Berthold and D. J. Hand, editors. *Intelligent Data Analysis*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2nd edition, 2004.

[25] R. Blagus and L. Lusa. Class prediction for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 11(1):523–539, 2010.

[26] 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, Dec 2012.

[27] L. Breiman. *Classification and regression trees*. CRC press, 1993.

[28] L. Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, Aug. 1996.

[29] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[30] 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*, 03(05):1171–1189, 2005.

[31] R. Caruana and D. Freitag. Greedy attribute selection. In *In Proceedings of the Eleventh International Conference on Machine Learning*, pages 28–36. Morgan Kaufmann, 1994.

[32] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

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

[34] Y. Chen and Y. Zhao. A novel ensemble of classifiers for microarray data classification. *Applied Soft Computing*, 8(4):1664 – 1669, 2008. Soft Computing for Dynamic Data Mining.

[35] N. Cristianini and B. Schölkopf. Support vector machines and kernel methods: The new generation of learning machines. *AI Mag.*, 23(3):31–41, Sept. 2002.

[36] M. Dash and H. Liu. Consistency-based search in feature selection. *Artificial intelligence*, 151(1):155–176, 2003.

[37] M. Dash, H. Liu, and H. Motoda. Consistency based feature selection. In T. Terano, H. Liu, and A. Chen, editors, *Knowledge Discovery and Data Mining. Current Issues and New Applications*, volume 1805 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin Heidelberg, 2000.

[38] B. S. Debahuti Mishra. Feature selection for cancer classification: a signal-to-noise ratio approach. *International Journal of Scientific and Engineering Research*, 2009, 2011.

[39] 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.

[40] T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000.

[41] D. Dittman, T. 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.

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

[43] 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.

[44] 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 *Ninth IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 147–152, December 2010.

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

[46] P. Domingos and M. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Mach. Learn.*, 29(2-3):103–130, Nov. 1997.

[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] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th international joint conference on Artificial intelligence - Volume 2*, pages 973–978, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

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

[50] R. Fraiman, A. Justel, and M. Svarc. Pattern recognition via projection-based knn rules. *Computational Statistics and Data Analysis*, 54(5):1390 – 1403, 2010.

[51] B. Frenay and M. Verleysen. Classification in the presence of label noise: A survey. *Neural Networks and Learning Systems, IEEE Transactions on*, 25(5):845–869, May 2014.

[52] Y. Freund and R. E. Schapire. Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[53] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[54] 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.

[55] G. J. Gordon, R. V. Jensen, L.-L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno. Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research*, 62(17):4963–4967, 2002.

[56] X. Guo, Y. Yin, C. Dong, G. Yang, and G. Zhou. On the class imbalance problem. In *Fourth International Conference on Natural Computation, 2008. ICNC '08.*, volume 4, pages 192–201, October 2008.

[57] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[58] M. Hall. *Correlation-based Feature Selection for Machine Learning*. PhD thesis, The University of Waikato, Hamilton, New Zealand, April 1997.

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

[60] D. W. Hosmer and S. Lemesbow. Goodness of fit tests for the multiple logistic regression model. *Communications in Statistics - Theory and Methods*, 9(10):1043–1069, 1980.

[61] N. Iizuka, M. Oka, H. Yamada-Okabe, M. Nishida, Y. Maeda, N. Mori, T. Takao, T. Tamesa, A. Tangoku, H. Tabuchi, K. Hamada, H. Nakayama, H. Ishitsuka, T. Miyamoto, A. Hirabayashi, S. Uchimura, and Y. Hamamoto. Oligonucleotide microarray for prediction of early intrahepatic recurrence of hepatocellular carcinoma after curative resection. *The Lancet*, 361(9361):923–929, 2003.

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

[63] W. Jiang. Some theoretical aspects of boosting in the presence of noisy data. In *In Proceedings of The Eighteenth International Conference on Machine Learning (ICML2001*, pages 234–241. Morgan Kaufmann, 2001.

[64] Y. Jiang, J. Lin, B. Cukic, and T. Menzies. Variance analysis in software fault prediction models. In *Software Reliability Engineering, 2009. ISSRE '09. 20th International Symposium on*, pages 99–108, Nov 2009.

[65] T. Jirapech-Umpai and S. Aitken. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6(1):148, 2005.

[66] A. Karmaker and S. Kwek. A boosting approach to remove class label noise. *Int. J. Hybrid Intell. Syst.*, 3(3):169–177, Aug. 2006.

[67] R. Khardon and G. Wachman. Noise tolerant variants of the perceptron algorithm. *Journal of Machine Learning Research*, 8:227–248, 2007.

[68] T. Khoshgoftaar, A. Fazelpour, H. Wang, and R. Wald. A survey of stability analysis of feature subset selection techniques. In *Information Reuse and Integration (IRI), 2013 IEEE 14th International Conference on*, pages 424–431, Aug 2013.

[69] T. M. Khoshgoftaar, D. Dittman, R. Wald, and A. Fazelpour. First order statistics based feature selection: A diverse and powerful family of feature seleciton techniques. In *11th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 151–157, Dec. 2012.

[70] 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.

[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] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[73] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[74] K. Lakshmi and S. Mukherjee. An improved feature selection using maximized signal to noise ratio technique for tc. In *Third International Conference on Information Technology: New Generations, 2006. ITNG 2006.*, pages 541–546, April 2006.

[75] T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15):2429–2437, 2004.

[76] 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.

[77] H. Liu and R. Setiono. Chi2: feature selection and discretization of numeric attributes. In *Proceedings of the Seventh International Conference on Tools with Artificial Intelligence, 1995.*, pages 388–391, November 1995.

[78] H. Liu and L. Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):491–502, April 2005.

[79] T.-Y. Liu. EasyEnsemble and feature selection for imbalance data sets. In *IJCBS '09: International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009.*, pages 517–520, August 2009.

[80] Y. Lu and J. Han. Cancer classification using gene expression data. *Information Systems*, 28:243–268, 2003.

[81] T. M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[82] F. Model, P. Adorja'n, A. Olek, and C. Piepenbrock.

[83] 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.

[84] N. C. Oza. Aveboost2: Boosting for noisy data. In J. K. Fabio Roli and T. Windeatt, editors, *Fifth International Workshop on Multiple Classifier Systems*, pages 31–40, Cagliari, Italy, June 2004. Springer-Verlag.

[85] M. Pechenizkiy, A. Tsymbal, S. Puuronen, and O. Pechenizkiy. Class noise and supervised learning in medical domains: The effect of feature extraction. In *Computer-Based Medical Systems, 2006. CBMS 2006. 19th IEEE International Symposium on*, pages 708–713, 2006.

[86] S. D. Peddada, E. K. Lobenhofer, L. Li, C. A. Afshari, C. R. Weinberg, and D. M. Umbach. Gene selection and clustering for time-course and dose-response microarray experiments using order-restricted inference. *Bioinformatics*, 19(7):834–841, 2003.

[87] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

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

[89] E. F. Petricoin III, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, and L. A. Liotta. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359(9306):572–577, 2002.

[90] F. Provost and T. Fawcett. Robust classification for imprecise environments. *Mach. Learn.*, 42(3):203–231, Mar. 2001.

[91] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[92] 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.

[93] 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.

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

[95] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26(5):pp. 1651–1686, 1998.

[96] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse. A study on the relationships of classifier performance metrics. In *21st International Conference on Tools with Artificial Intelligence*, pages 59–66, November 2009.

[97] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature medicine*, 8(1):68–74, Jan. 2002.

[98] R. Simon. Roadmap for developing and validating therapeutically relevant genomic classifiers. *J Clin Oncol*, 23:7332–7341, 2005.

[99] R. L. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, 19(12):1484–1491, 2003.

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

[101] A. C. Tan and D. Gilbert. Ensemble machine learning on gene expression data for cancer classification. *Applied bioinformatics*, 2(3):75 – 83, 2003.

[102] 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.

[103] J. Van Hulse and T. M. Khoshgoftaar. Knowledge discovery from imbalanced and noisy data. *Data & Knowledge Engineering*, 68(12):1513–1542, 2009.

[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, and A. Napolitano. A comparative evaluation of feature ranking methods for high dimensional bioinformatics data. In *2011 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 315–320, August 2011.

[106] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald. Feature selection with high-dimensional imbalanced data. In Y. Saygin, J. X. Yu, H. Kargupta, W. Wang, S. Ranka, P. S. Yu, and X. Wu, editors, *IEEE International Conference on Data Mining Workshops, 2009. ICDMW '09.*, pages 507–514. IEEE Computer Society, December 2009.

[107] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald. Threshold-based feature selection techniques for high-dimensional bioinformatics data. *International Journal of Network Modeling and Analysis in Health Informatics and Bioinformatics*, 1(1–2):47–61, 2012.

[108] S. Visa and A. Ralescu. Issues in mining imbalanced data sets – a review paper. In *Proc. 16th Midwest Artificial Intelligence and Cognitive Science Conf.*, pages 67–73, 2005.

[109] R. Wald, T. M. Khoshgoftaar, and A. Abu Shanab. The effect of measurement approach and noise level on gene selection stability. In *Bioinformatics and Biomedicine (BIBM), 2012 IEEE International Conference on*, pages 1–5, Oct 2012.

[110] R. Wald, T. M. Khoshgoftaar, and A. Abu Shanab. Comparison of two frameworks for measuring the stability of gene-selection techniques on noisy class-imbalanced data. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 881–888, Nov 2013.

[111] R. Wald, T. M. Khoshgoftaar, A. Abu Shanab, and A. Napolitano. Comparative analysis on the stability of feature selection techniques using three frameworks on biological datasets. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, pages 418–423, 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 *IEEE 14th International Conference on Information Reuse and Integration (IRI 2013)*, pages 232–238, 2013.

[113] X. Wang and O. Gotoh. Accurate molecular classification of cancer using simple rules. *BMC Medical Genomics*, 2(1):64, 2009.

[114] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. X. Mayer, and H. W. Mewes. Gene selection from microarray data for cancer classification–a machine learning approach. *Computational Biology and Chemistry*, 29(1):37–46, 2005.

[115] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, Burlington, MA, 3rd edition, January 2011.

[116] M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11(11):1878–1887, 2001.

[117] X. Zhu and X. Wu. Class noise vs. attribute noise: A quantitative study. *Artificial Intelligence Review*, 22(3):177–210, Nov 2004.