TOWARD IMPROVED CLASSIFICATION OF IMBALANCED DATA

A Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

of Doctor of Philosophy

By

Bassam Almogahed

December 2014

TOWARD IMPROVED CLASSIFICATION OF IMBALANCED DATA

Bassam Almogahed

APPROVED:

Ioannis Kakadiaris, Chairman
Dept. of Computer Science

Shishir Shah
Dept. of Computer Science

Christoph Eick
Dept. of Computer Science

Ricardo Vilalta
Dept. of Computer Science

Panagiotis Tsiamyrtzis
Dept. of Statistics,
Athens University of Economics and Business

Dean, College of Natural Sciences and Mathematics

Acknowledgements

This dissertation would not have been possible without the support of many people. First and foremost, I would like to thank my advisor and role model, Dr. Ioannis Kakadiaris. Dr. Kakadiaris helped me see the pros and cons of so many decisions, small and large. I truly admire his ability to understand the nuances in absolutely everything. I have grown a tremendous amount as a researcher and computer scientist by virtue of working with such a brilliant mentor. I would like to extend my deepest gratitude to him for his guidance, and for reading countless drafts of my multiple papers.

I would also like to thank all of my friends and supportive team members in the UH, Computational Biomedicine Lab group—in particular, Panagiotis Moutafis and Yen Le: Panagiotis for being a source not only of statistical prowess, but also of unfailing positivity and humorous yet enlightening discussions.Yen for being such a fantastic officemate and for all the advice and lively dialogues.

A special thank you to Ariel K. Salzer, Esq. for her expert writing advice and insight, and for proofreading numerous iterations of my papers (often at the last minute). Your attention to detail is unparalleled. Without your support and encouragement, this work would not have been impossible.

Finally, I would not be where I am today without the amazing support, encouragement and love from my parents, Dr. Abdullah Almogahed and Katiba Al-Harazi. This thesis is for the two of you.

TOWARD IMPROVED CLASSIFICATION OF IMBALANCED DATA

An Abstract of a Dissertation

Presented to

the Faculty of the Department of Computer Science

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

of Doctor of Philosophy

By

Bassam Almogahed

December 2014

Abstract

There is an unprecedented amount of data available. This has caused knowledge discovery to garner attention in recent years. However, many real-world datasets are imbalanced. Learning from imbalanced data poses major challenges and is recognized as needing significant research. The problem with imbalanced data is the performance of learning algorithms in the presence of underrepresented data and severely skewed class distributions. Models trained on imbalanced datasets strongly favor the majority class and largely ignore the minority class. Several approaches introduced to date present both data-based and algorithmic solutions. However, both types of approaches have been criticized for their lack of generalization, tendency to forfeit information, and likelihood of resulting in over-fitting difficulties.

The goal of this thesis is to develop algorithms to balance imbalanced datasets to allow each classifier to reach optimal predictions. The specific objectives are to: (i) develop sampling methods for imbalanced data, (ii) develop a framework capable of determining which sampling method to use, (iii) evaluate performance of these methods on a variety of imbalanced datasets, and (iv) develop a new machine learning risk-prediction framework for cardiovascular events.

We propose a method for filtering over-sampled data using non-cooperative game theory. It addresses the imbalanced data issue by formulating the problem as a non-cooperative game. The proposed algorithm does not require any prior assumptions and selects representative synthetic instances while generating only a very small amount of noise. We also propose a technique for addressing imbalanced data using semi-supervised learning. Our method integrates under-sampling and semi-supervised learning (US-SSL) to tackle the imbalance problem. The proposed

algorithm, on average, significantly outperforms all other sampling algorithms in 67% of cases, across three different classifiers, and ranks second best for the remaining 33% of cases. Finally, we propose a novel framework based on the US-SSL algorithm to select the appropriate semi-supervised algorithm to balance and refine a given dataset in order to establish a well-defined training set.

Contents

1 Introduction **1**

1.1 Motivation 1

1.2 Challenges 2

1.3 Goal 3

1.4 Contributions 4

1.5 Dissertation Outline 5

1.6 Publications 5

2 Literature Review **7**

2.1 Imbalanced Data 7

2.1.1 Overview 7

2.1.2 Algorithmic Approaches 8

2.1.3 Data-based Approaches 10

2.1.4 Performance Measurement Analysis 17

2.1.5 Statistical Tests 21

2.2 Non-Cooperative Game Theory 23

2.2.1 Concepts and Notation 24

2.2.2 Nash Equilibrium 25

2.2.3 Replicator Dynamics 26

2.3 Semi-Supervised Learning (SSL) 28

2.4 Disease-Prediction Models 29
2.5 Risk Assessment for Cardiovascular Disease 33
2.5.1 Risk Factors 34
2.5.2 Calcium Scoring 34
2.5.3 Follow-up 35
2.5.4 Cardiovascular Events 35
2.5.5 Methods 36
2.5.6 Datasets 38

3 NEATER **40**
3.1 Filtering of Over-Sampled Data Using Non-Cooperative Game Theory 40
3.1.1 Overview 40
3.1.2 Computing Nash Equilibria Using Replicator Dynamics 43
3.1.3 NEATER Implementation Details 46
3.1.4 Computational Complexity Analysis 48
3.2 Experimental Design and Results 49
3.2.1 Datasets 49
3.2.2 Experimental setup 53
3.3 Discussion 54

4 US-SSL **69**
4.1 A Novel Framework for Handling Imbalanced Data in Supervised Learning: A Semi-Supervised Learning Approach 69
4.1.1 Data Pre-processing Procedure 70
4.1.2 Empowering Imbalanced Data in Supervised Learning: A Semi-Supervised Learning Approach (US-SSL) 70
4.1.3 US-SSL Implementation Details: 72
4.1.4 Framework for US-SSL Method Selection 74

4.2 Experimental Design and Results 78
4.2.1 Datasets 78
4.2.2 Experimental Setup 85
4.3 Discussion 87

5 CardioRS **104**
5.1 Framework for Predicting Cardiovascular Events 104
5.1.1 Overview 104
5.1.2 Decision Tree Classifier (J48) 106
5.1.3 CardioRS Framework 108
5.2 Experimental Design and Results 110
5.2.1 Experiments Setup 110
5.2.2 Statistical Analysis 110
5.2.3 CardioRS Analysis 114
5.2.4 Comparison and Evaluation 116

6 Conclusion **119**
6.1 Summary of Contributions 119
6.2 Future Work 120

Bibliography **122**

List of Figures

2.1 (a) Example of the K-nearest neighbors for the x_i example under consideration ($K = 5$). (b) Data created based on the Euclidian distance. (c) The new generated instance is labeled as minority in the new post-processed dataset. 14

2.2 Pie-chart representation of the synthetic instance proportions for SMOTE and ADASYN. 16

3.1 Comparison of different synthetic data generation mechanisms. (a) Original imbalanced data distribution (359 majority examples and 39 minority examples); Data distribution after: (b) SMOTE method; (c) ADASYN method; (d) NEATER-SMT method; and (e) NEATER-ADA method. The red color indicates instances that belong to the majority class while blue indicates the minority class. 47

3.2 Friedman's average ranks for the three classifiers. 58

4.1 Illustration of the US-SSL algorithm. 73

4.2 Comparison of different US mechanisms (the blue circles represent the majority group and the red squares represent the minority group): (a) original imbalanced data distribution; (b) data distribution after OSS method; (c) data distribution after RUS method; (d) data distribution after US-TSVM method; (e) data distribution after US-LGC method; and (f) data distribution after US-GTAM method. 77

4.3 Illustration of the US-SSL framework. 78

4.4 Two-dimensional projections of g241c (left) and g241d (right). Black circles, class +1; gray crosses, class -1. 82

4.5 First data point from Digit1 dataset. Original image (left), and after rescaling, adding noise, and masking dimensions (x) (right). 83

4.6 Fourth data point from the USPS dataset. Original image (left), and after rescaling, adding noise, and masking dimensions (x) (right). . . 84

4.7 Friedman's average ranks for the three classifiers. 89

4.8 Friedman's average ranks for the three classifiers (IR = 18%). 94

4.9 Friedman's average ranks for the three classifiers (IR = 09%). 95

5.1 Depiction of a decision tree classifier. 107

5.2 Depiction of attributes and decisions. 108

5.3 Depiction of the CardioRS framework. 110

List of Tables

2.1 Cost matrix 9

2.2 Confusion matrix for performance evaluation 19

2.3 Summary of established disease prediction models in heart and cancer studies using statistical techniques. 31

2.4 summary of established disease prediction models in heart and cancer studies using AI techniques. 32

2.5 Number of CHD events and CVD events in MESA data based on gender. 36

2.6 Cardiovascular events and base models. 37

3.1 Summary of imbalanced datasets used. 51

3.2 Summary of imbalanced high-dimensional datasets used. 51

3.3 Average AUC, GM and AGF values for three different classifiers for the SMOTE-based algorithms 55

3.4 Average AUC, GM and AGF values for three different classifiers. . . . 56

3.5 Results obtained with the Holm test for $\alpha = 0.05$. 59

3.6 Summary of the Wilcoxon statistic for the over-sampling algorithms with C4.5 classifier. Upper and lower diagonal halves are generated for $\alpha = 0.01$ and $\alpha = 0.05$, respectively. 61

3.7 Summary of the Wilcoxon statistic for the over-sampling algorithms with the Random Forest classifier. Upper and lower diagonal halves generated are for $\alpha = 0.01$ and $\alpha = 0.05$, respectively. 61

3.8 Summary of the Wilcoxon statistic for the over-sampling algorithms with SVM classifier. Upper and lower diagonal halves are generated for $\alpha = 0.01$ and $\alpha = 0.05$, respectively. 63

3.9 Summary of how many times the over-sampling techniques have been significantly-better/same/significantly-worse. 63

3.10 Average AUC, GM and AGF values on high-dimensional datasets for three different classifiers. 64

4.1 Imbalanced datasets. 79

4.2 Basic properties of imbalanced benchmark datasets. 81

4.3 First row: number of components kept in the dimensionality reduction; second row: estimate of the manifold dimension according to Hein and Audbert algorithm 86

4.4 Average AUC, GM and AGF values for three different classifiers . . . 88

4.5 US-SSL results obtained from the Wilcoxon signed-rank test. 89

4.6 Average AUC, GM and AGF values for the three different classifiers (IR = 18%). 92

4.7 Average AUC, GM and AGF values for the three different classifiers (IR = 09%). 93

4.8 US-SSL results obtained from the Wilcoxon signed-rank test (IR = 18%). 93

4.9 US-SSL results obtained from the Wilcoxon signed-rank test (IR = 09%). 93

4.10 AUC, GM, and AGF results for 3 different classifiers (IR = 18%). . . 97

4.11 AUC, GM, and AGF results for 3 different classifiers (IR = 09%) . . 99

4.12 Summary of how well the selected US-SSL method performs in comparison to the actual reported results for the imbalanced data (IR $<$ 20%). This result is based on the highest reported performance across all three classifiers: C4.5, Random Forests and SVM. 100

4.13 Summary of how well the selected US-SSL method performs in comparison to the actual reported results for highly imbalanced data (IR < 10%). This result is based on the highest reported performance across all three classifiers: C4.5, Random Forests and SVM. 100

5.1 CHDH, CHDA and CVDA events classified as "low risk" (<10%) and "high risk" (10% ≥) and the standard FRS of patients 112

5.2 CHDH-MEN, CHDH-WOMEN events by Coronary Artery Calcium (CAC) score in men and women in the two risk categories: "Low Risk" (<10%) and "High Risk" (10% ≥) and the Standard FRS of Patients 113

5.3 Sensitivity and specificity under FRS for each event type: CVDA, CHDA and CHDH 114

5.4 Sensitivity and specificity under CRS for men for each event type: CVDA, CHDA and CHDH. NEATER as the base algorithm 115

5.5 Difference of sensitivity, specificity between base models and CRS for all events - NEATER is the base algorithm 116

Chapter 1

Introduction

1.1 Motivation

In recent years, the class-imbalance problem has received much attention in both academia and industry. Many real-world datasets are imbalanced, meaning that they are composed of a large number of "normal" negative examples and only a small percentage of "abnormal" or "interesting" positive examples [20]. Formally, the class-imbalance problem occurs when the samples from one or several classes significantly outnumber the samples from other classes in a dataset.

In cases of imbalanced datasets, most learning systems will be greatly biased. Specifically, models trained from imbalanced datasets tend to strongly favor the majority class and largely ignore the minority class [20]. For instance, in a dataset in which only 1% of the instances are positive, the accuracy of the classifier will be

99% when the classifier classifies all the instances as negative.

The problem lies in the fact that most traditional Machine Learning classification algorithms are based on the assumption that a dataset will have a balanced class distribution [15, 22]. In addition, these algorithms are designed to generalize from sample data and output the simplest hypothesis which fits the data. This principle is embedded in the inductive bias of many classification Machine Learning algorithms, including Decision Trees, Nearest Neighbor, and Support Vector Machine (SVM). Therefore, when they are used on complex imbalanced datasets, these algorithms are inclined to be overwhelmed by the majority class and disregard the minority class. This can cause classification errors for the minority class [58].

1.2 Challenges

Classifying data from the minority class using a standard classification algorithm is a challenging task, mainly due to the imbalanced nature of the data. Many researchers have studied this problem and consequently, many ideas have emerged.

The primary challenges that arise when handling datasets with imbalance problems are:

1. Different imbalanced datasets have different characteristics:

   (a) the degree of bias between class imbalance could be in the order of 100:1, 1,000:1, or 10,000:1, and in all cases, one class is out-represented by the other,

(b) the number of features varies from just a few features to hundreds, and

(c) the size of the dataset varies from less than a hundred to thousands of instances.

2. Standard classification algorithms seek to minimize the overall classification error rate by producing a biased hypothesis, which regards almost all instances as members of the majority class. Adjustment of classifiers may provide quality results for a specific dataset but does not provide a generalized solution to the imbalance problem.

3. Changing the class distribution of a dataset may result in either removing potentially important data (under-sampling), or over-fitting (over-sampling).

4. Adjusting the costs of the various classes to counter the class imbalance, adjusting the probabilistic estimate at the tree leaf when working with the class distribution may cause over-fitting.

1.3 Goal

The goal of this research is to develop new automated methods to balance imbalanced datasets in order for the classifier to reach optimal predictions.

The specific objectives are to:

1. Develop sampling methods for imbalanced datasets

2. Develop a framework which can help determine which sampling method should be used given an imbalanced data

3. Evaluate the performance of these methods on a variety of imbalanced datasets

4. Develop of a new risk prediction framework for cardiovascular events using machine learning sampling methods.

Although we consider only imbalanced two-class problems in our evaluation, our approaches can be extended to multi-class datasets as well.

1.4 Contributions

To date, we have made the following contributions:

1. Developed a new over-sampling filtering method for imbalanced datasets. The method is based on non-cooperative game theory, which ultimately produces synthetic data belonging to the minority group—and as a result, helps the classifier to produce a hypothesis which can better identify new data that belongs to the minority class.

2. Developed a new under-sampling (US) method to address the imbalanced data problem using semi-supervised learning (SSL).

3. Developed a novel framework based on the US-SSL algorithm to select the appropriate semi-supervised algorithm to balance and refine a given dataset to establish a well-defined training set.

4. Developed a cardiovascular events prediction framework—CardioRS—which has stronger prediction power than the models currently in use today.

1.5 Dissertation Outline

The organization of the remainder of this dissertation is as follows: Chapter 2 presents a literature review of existing approaches in the application domain and a literature review of methods related to this work. In addition, it sets forth a brief introduction to non-cooperative game theory and the various concepts and notations that will be used in our approach. Chapter 3 describes the proposed methodology and frameworks for filtering over-sampled data, accompanied by the results and discussion. Chapter 4 presents our semi-supervised approach framework for handling imbalanced data. In addition, it discusses the experiment setup and results. Chapter 5 presents the framework for predicting cardiovascular events (CardioRS). Finally, Chapter 6 concludes with our contribution highlights, planned future work and the timeline of the thesis.

1.6 Publications

1. I. A. Kakadiaris, B. A. Almogahed, H. S. Hecht, C. T. Sibley, M. Naghavi, M. Budoff. CardioRS: Toward a New Methodology for Predicting Cardiovascular Events. *Journal American College of Cardiology*, 2013;61: doi:10.1016/S0735-1097(13)60163 - 1.

2. Ioannis A. Kakadiaris, Bassam A. Almogahed, Harvey S. Hecht, Christopher T. Sibley, Susan R. Heckbert, Morteza Naghavi, Matthew Budoff. CardioRS (CRS): Toward a New Methodology for Predicting Cardiovascular Events. *Journal of the American College of Cardiology.* 2014 (In Preparation). Impact Factor: 14.086.

3. Bassam A. Almogahed, Ioannis A. Kakadiaris: NEATER: Filtering of Over-sampled Data Using Non-cooperative Game Theory. *Soft Computing.* 2014

4. B. A. Almogahed and I. A. Kakadiaris: Empowering imbalanced data in supervised learning: A semi-supervised learning approach. In *Proc. International Conference on Artificial Neural Networks*, Hamburg, Germany, Sep 15-19 2014.

5. B. A. Almogahed and I. A. Kakadiaris. NEATER: filtering of over-sampled data using non-cooperative game theory. In *Proc. International Conference of Pattern Recognition*, Stockholm, Sweden, Aug 2014.

6. B. A. Almogahed and I. A. Kakadiaris. A Novel Framework for Handling Imbalanced Data in Supervised Learning: A Semi-Supervised Learning Approach. *Machine Learning.* 2014 (Under Review).

Chapter 2

Literature Review

2.1 Imbalanced Data

2.1.1 Overview

The class-imbalance problem occurs when the samples from one or several classes significantly outnumber the samples from other classes in a dataset. In cases of imbalanced datasets, most learning systems will be greatly biased. Specifically, models trained from imbalanced datasets are intended to strongly favor the majority class and largely ignore the minority class [20]. The problem lies in the fact that most traditional classification algorithms are based on the assumption that a dataset will have a balanced class distribution [22]. To overcome the challenge of imbalanced data, many approaches have been introduced that present several solutions at both the algorithmic and data levels [57, 58].

2.1.2 Algorithmic Approaches

At the algorithmic level, the most commonly employed method is to adjust the operation of the existing algorithm to make the classifier more conducive to the classification of the minority class [9, 97, 95].

For instance, Drummond and Holte [38] observed that over-sampling neither significantly improved performance, nor resulted in a change in classification. On the contrary, over-sampling prunes less than under-sampling using the default parameters for the C4.5 algorithm. A modification of the parameter settings of C4.5 improved classification performance and avoided the over-fitting problem during over-sampling. Therefore, while sampling methods attempt to balance distributions by considering their representative proportions of class examples, other approaches, such as cost-sensitive learning method (CSL), consider the costs associated with misclassifying examples [40, 119].

Instead of creating artificial data to balance data distributions, CSL directly targets the imbalanced data by using different cost matrices which describe the costs for misclassifying any particular data example.

In classical machine learning and data-mining settings, classifiers seek to minimize the number of misclassified new instances: false negatives and false positives. Furthermore, most classifiers assume that misclassification costs are equal. However, in many real-world applications, the costs of misclassifications are often different. For example, as described in medical diagnoses, the cost of incorrectly diagnosing a patient to be healthy when in fact he or she is unhealthy may be potentially much

Table 2.1: Cost matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	$C(1,1)$	$C(0,1)$
	Negative	$C(1,0)$	$C(0,0)$

more devastating than diagnosing a healthy person as being ill. In this way, diagnosing a false negative—a Type II error—could potentially be much more serious than diagnosing a false positive—Type I error.

As shown in the cost matrix table above, we denote the positive class 1 as the minority, and the negative class 0 as the majority, where $C(i, j)$ represents the cost of misclassifying an instance belonging to class i when, in actuality, it belongs to class j.

The objective of this strategy is to build a model with minimum total misclassification cost (TC) which is defined as: $TC = C(0,1) \times \#FN + C(1,0) \times \#FP$, where $\#FN$ and $\#FP$ represent the number of false negatives and false positives, respectively and in some applications; cost sensitive techniques have performed better than sampling methods [85, 80].

MetaCost [35] is another cost-sensitive learning algorithm which estimates class probabilities using bagging and then re-labels the training instances with their expected classes. Finally, it rebuilds the model using the modified training set. Some classifiers use a scoring method to show the degree to which an instance belongs to a class. The advantage of this approach is that this ranking can be used in various classifiers by changing the threshold for an instance belonging to a class [129].

One-class learning is an alternative to binary classification where the model is created based on the instances of the target class alone. The basic idea is that the boundaries between two classes are estimated from the data of one class (the target class) so that this approach is not sensitive to the class distribution in the training set. A boundary around the target class is defined such that most of the target objects are included and at the same time the chance of accepting outlier objects is minimized. Many studies have shown that one-class learning is useful for extremely imbalanced datasets with a high dimensional noisy feature space [72, 105].

Besides the risk of over-fitting and the additional complexity, the main drawback of this class of approaches is that they require special knowledge of both the corresponding classifier and the application domain.

2.1.3 Data-based Approaches

At the data level, the most typical approach is to modify the datasets used. These data preprocessing methods can be grouped into two categories: over-sampling [9, 73, 138] and under-sampling [21, 54, 10, 47]. The data-based approach has garnered more investigation since it is classifier independent and can be easily implemented for any problem [47]. However, the main drawback of this solution is that it artificially alters the original class distribution where over-sampling may result in the loss of important information and under-sampling may cause an over fitting problem.

2.1.3.1 Under-sampling (US)

Under-sampling techniques aim to balance a dataset by removing instances that will not cause the classifier to miss important concepts pertaining to the majority class. Since US may lead to loss of potentially useful data, some heuristic under-sampling methods seek to remove superfluous instances which will not affect the classification accuracy of the training set. Some of the classic US methods for balancing class distribution are: Random Under-Sampling (RUS), One-Sided Selection (OSS) [73], Class Purity Maximization (CPM) [139], NearMiss-2 [83], and Under-Sampling Based on Clustering (SBC) [137].

A successful US technique retains all minority examples and prunes only unreliable majority examples which are: (i) negatively impacted by class-label noise; (ii) *redundant*, such that their part can be taken over by other examples; (iii) *borderline* (i.e., close to the boundary between the minority and majority regions); and (iv) borderline examples from the majority class in the *overlapping* regions between classes (in particular for non-linear decision boundaries). This latter category can also plague synthetic examples created by the over-sampling techniques mentioned above. Categories (i) through (iii) above are not as harmful to classifier performance as category (iv); they can easily be detected and eliminated by the *Tomek links* concept [120]. Any performance degradation is caused mainly by the overlap between the imbalanced classes. More recent experiments on artificial data with different degrees of overlapping have demonstrated that overlapping is more important than the overall imbalance ratio [103].

2.1.3.2 Over-sampling (OS)

There are several over-sampling techniques. The simplest is called random over-sampling. Random over-sampling is a mechanism for adding a set E of additional instances (i.e., instance duplicates) randomly sampled from the minority class to the original dataset, D. Consequently, the number of total instances of the minority class is increased, which results in a more balanced class distribution. This provides a mechanism for varying the degree of class distribution balance to any desired balance level. This over-sampling method does not increase information for the minority class. Instead, by replication, it raises the weight of its samples. This then causes the classification rule to become too specific. Therefore, an over-fitting problem will generally occur. Even though the accuracy for the training set is high, the classification performance for new test datasets will likely be worse. By duplicating data and adding it to the original dataset, some of the copied data becomes too specific and classifiers will produce multiple clauses for the duplicate data [73]. Hence, a more effective over-sampling technique is needed.

The most common over-sampling method is Synthetic Minority Over-Sampling Technique (SMOTE), proposed by Chawla *et al.* [20]. SMOTE synthesizes new minority class examples using several neighboring minority examples rather than simply duplicating them as is done in ROS [63]. The SMOTE algorithm is very effective, but may cause an over-generalization problem due to the way it creates synthetic data. This results in increased overlap between the two classes [124]. To overcome this challenge, recent literature has proposed modifying the original SMOTE algorithm in several ways. Most of these modifications seek to determine the region in

which the positive examples should be generated [57].

The most widely-known of these proposed ameliorations are the Borderline SMOTE (B-SMOTE) algorithm [54], which uses only positive examples that are close to the decision boundary since these are more likely to be misclassified, and the Safe-Level-SMOTE (SMOTE-SL) algorithm [14], which defines a 'safe-level' for each positive instance and uses only safe instances to generate synthetic instances.

State-of-the-art over-sampling algorithms include Adaptive Synthetic Sampling (ADASYN) [56] and hierarchical clustering (AHC) [26]. ADASYN uses a density distribution as a criterion to adaptively determine the number of synthetic examples to be generated for each minority class instance according to its level of difficulty in learning. The AHC over-sampling method generates synthetic positive examples by forming a dendrogram using single and complete linkage. Next, it gathers clusters from all dendogram levels and computes their centroids. These centroids are added to the original data as synthetic positive examples.

These approaches have improved learning with respect to data distributions on imbalanced datasets by reducing the bias of class distribution and adaptively shifting the decision boundary to put more attention on instances which are difficult to learn. On the other hand, though, over-sampling increases the size of the data, and may thereby worsen the computational burden of the learning algorithm. Currently, there is no single approach that has emerged to solve the imbalanced data problem. However, studies which have investigated both methods have reported that over-sampling generally performs better than under sampling [10, 122].

SMOTE: Synthetic Minority Over-Sampling Technique:

SMOTE is based on the assumption that the examples close to the minority examples also belong to the minority class. Unlike traditional copy-based oversampling methods, SMOTE is an over-sampling method in which the minority class over-samples by creating synthetic examples based on the feature space similarities between existing minority examples, rather than by over-sampling with replacement [20]. SMOTE can control the number of examples and distributions to achieve the purpose of balancing the dataset through synthetic new examples. When SMOTE deals with nominal (or discrete) and continuous attributes, it creates artificial data differently [20, 140].

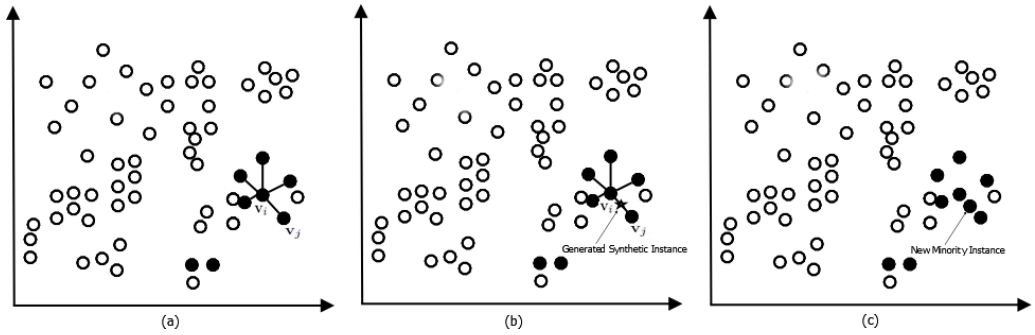


Figure 2.1: (a) Example of the K-nearest neighbors for the x_i example under consideration ($K = 5$). (b) Data created based on the Euclidian distance. (c) The new generated instance is labeled as minority in the new post-processed dataset.

As illustrated in Fig. 2.1, first let us consider a dataset D and two subsets $D_m \subset D$ and $D_M \subset D$, where D_m is the data associated with the minority class in D, and D_M is the set of data associated with the majority class in D, so that the two sets are mutually exclusive and their union is the universal set D, and every example $\mathbf{v}_i \in D$ is an instance in the n-dimensional feature space V. For a subset

D_m, consider the K-nearest neighbors for each example $\mathbf{v}_i \in D_m$, for some specified integer K (which is set to 5 in SMOTE). To create a synthetic sample, we randomly select one of the K-nearest neighbors ($\mathbf{v}_j$), then multiply the feature vector difference with a random number between 0 and 1 and add this vector to $\mathbf{v}_i$: $\mathbf{v}_y = \mathbf{v}_i + \delta(\mathbf{v}_j - \mathbf{v}_i)$, where $\mathbf{v}_i \in D_m$ is the minority example that we are considering and $\mathbf{v}_j$ is one of the K-nearest neighbors for $\mathbf{v}_i : \mathbf{v}_j \in D_m$, and δ is a random number $\delta \in [0, 1]$. The resulting synthetic instance is generated along the line between the minority sample $\mathbf{v}_i$ and the randomly selected K-nearest neighbor $\mathbf{v}_j$ [24]. All synthetic data form a new subset D_y and the dataset D consists of three mutually exclusive subsets $\{D = D_m \cup D_M \cup D_y \}$.

ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning:

Similarly to SMOTE, ADASYN also uses feature interpolation to generate synthetic instances. The difference is, instead of applying a uniform distribution for data generation like SMOTE, ADASYN uses a density distribution $\hat{\mathrm{r}}_i$ as a criterion to automatically decide the number of synthetic samples that need to be generated for each minority example.

The density distribution criteria is defined as the normalized number of majority cases within the k-nearest neighbor of each minority example. Figure 2.2 depicts the proportion of synthetic instances generated by SMOTE and ADASYN based on three-nearest neighbors for minority examples N_1 through N_4. From Fig. 2.2, it is clear that: (i) SMOTE uniformly assigns the number of synthetic instances to be generated for N_1 to N_4, and (ii) ADASYN uses a different distribution to

determine the number of synthetic instances for N_1 to N_4. ADASYN computes the number of majority cases within the three-nearest neighbors of N_1 to N_4 first, which is {3,0,1,2} in this example, and normalizes this into the density distribution {3/3, 0, 1/3, 2/3} $\Rightarrow$ {1/2, 0 , 1/6, 1/3}, which is used to bias the data generation process. ADASYN does not generate instances of minority examples without majority cases in their k-nearest neighbors, such as N_2 in Fig 2.2.

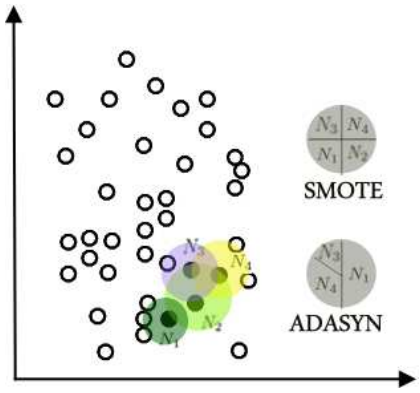


Figure 2.2: Pie-chart representation of the synthetic instance proportions for SMOTE and ADASYN.

2.1.3.3 Hybrid Techniques

Hybrid techniques have also been developed. These hybrids combine over- and under-sampling methods to over-sample and then clean the majority and minority class data. For example, the SOMTE + ENN (SMOTE-ENN) [10] hybrid relies on SMOTE to over-sample the minority class and then ENN uses cleaning methods to remove instances from both classes. Similarly, SMOTE + Tomek (SMOTE-TK) [10] uses SMOTE to over-sample and then Tomek to clean. Other hybrid algorithms such

as SMOTE-RSB [104] use SMOTE to generate the synthetic samples for the minority class and apply cleaning techniques to the newly generated instances. Rough Set Theory (RST) is the cleaning approach used in the SMOTE-RSB method.

The main drawbacks of these algorithms are that SMOTE and ADASYN both create representative synthetic data, but do so while generating a high degree of noise, and their decision boundary is greatly influenced. B-SMOTE, SMOTE-TK, SMOTE-ENN and SMOTE-SL, on the other hand, are able to keep noise levels down while only slightly influencing the decision boundary. They do not pay attention to the interior instances and create only a few representative synthetic instances. Hence, they do well when using the SVM classifier, but poorly with the other classifiers. SMOTE-RSB performs competitively as compared to other leading algorithms when used with the C4.5 classifier when the dataset is not severely imbalanced. However, along with SMOTE-TK, SMOTE-RSB runtime costs are generally higher than other comparative algorithms, especially when the size of the dataset is large.

2.1.4 Performance Measurement Analysis

Many measures have been developed for performance evaluation on imbalanced classification problems. The most commonly used metric for measuring the performance of learning systems is the overall accuracy. Generally, for a two-class classification problem, classification performance is evaluated by a confusion matrix table in which each column of the matrix represents the instances in a predicted class. Each row represents the instances in an actual class.

In classical machine learning and data-mining settings, classifiers seek to minimize the number of misclassified new instances: false negatives and false positives. Furthermore, most classifiers assume that misclassification costs are equal. However, when addressing the class of imbalance problems, the overall classification accuracy metric is not appropriate because it does not consider mis-classification costs and is strongly biased in favor of the majority class [42].

$$Accuracy = \frac{(TP + TN)}{(TP + FN + FP + TN)}$$

For this reason, other metrics are considered for such evaluation, namely, precision, recall, F-measure of the accuracy on the majority class and the minority class. These metrics are extracted from the confusion matrix. Throughout this document, positive instance corresponds to the minority class and negative instance to the majority class. Precision and recall are defined as:

$$precision = \frac{TP}{TP + FP}$$

and

$$recall = \frac{TP}{TP + FN}$$

Here, *precision* -positive predictive value- is a measure of how many instances were correctly labeled as positive. Recall, which is also referred to as sensitivity or True Positive Rate, is a measure of how many instances of the positive class were labeled correctly.

Table 2.2: Confusion matrix for performance evaluation

		Predicted Outcome		
		p	**n**	**total**
Actual Class	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Precision and recall are both less sensitive to changes in data distributions. They are therefore preferable measurements to accuracy. As an assessment of the accuracy for the positive class, precision is somewhat sensitive to data distributions, while recall is not. Recall gives no insight into how many instances are incorrectly classified as positive. Similarly, precision does not tell us how many positive instances are incorrectly classified. Nevertheless precision and recall can effectively evaluate classification performance in imbalanced learning scenarios.

The F-measure, or balanced F-score, is a performance metric which is based on the harmonic mean for the classifiers precision and recall. It measures the effectiveness of classification with respect to the user's coefficient, which determines the weighted importance on either recall or precision.

$$F - measure = \frac{(1 + \beta^2) \times recall \times precision}{\beta^2 \times recall + precision}$$

where β is an adjustable coefficient to the relative importance of recall versus precision. Consequently, this provides more insight into the functionality of a classifier than the accuracy metric.

We will also report the geometric mean (GM) metric which is used for evaluating classifiers in imbalanced domains [9] to objectively estimate a model's prediction capabilities [46, 77]. Finally, recent studies have highlighted the unsuitability of the F-measure in processing highly-imbalanced problems, since it is designed to focus on the positive (minority) class without taking into account the proper effect of the negative (majority) class.

In other words, two classifiers may have the same F-score, even though they have two different true negative rates (TNR) [84, 64]. An alternative measure is the Adjusted F-measure (AGF). The AGF is a generalization of the F-measure which has been proven to be more robust in measuring a classifier's performance on imbalanced case data. We will report AGF in this study [84].

The last metric that we will consider for imbalanced data learning is the Receiver Operating Characteristic (ROC) analysis from signal detection theory. The area under the ROC curve (AUC) assesses overall classification performance [13]. The advantage of such a metric is that AUC does not place more emphasis on one class over the other. Thus, it is not biased against the minority class.

2.1.5 Statistical Tests

Non-parametric tests are generally preferred over parametric methods. This is because the non-parametric nature of the problems typically results in violation of the usual assumptions (e.g., independence, normality, and homogeneity of the variance). Here, the AUC results have been further tested for statistically significant differences using non-parametric tests [33, 45].

Both pairwise and multiple comparisons have been used in this paper. First, to determine whether there are statistically significant differences among the oversampling techniques, we have employed the Iman-Davenport statistic. This technique begins by computing the Friedman's ranking of the algorithms for each dataset independently according to the AUC results: when there are twelve competing strategies, the ranks for each dataset range from 1 (best) to 12 (worst); in case of ties, average ranks are assigned. Then the average rank of each algorithm across all datasets is computed. Under the null-hypothesis, which states that all the algorithms are equivalent, the Friedman's statistic can be computed as follows:

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_j R_j^2 - \frac{K(K+1)^2}{4} \right] \tag{2.1}$$

where N denotes the number of datasets, K is the total number of algorithms, and R_j is the average rank of the algorithm j.

The χ_F^2 is distributed according to the Chi-square distribution with $K-1$ degrees of freedom, when N and K are big enough. However, as compared to each other, the Iman-Davenport statistic is more useful than the Friedman statistic for

our purposes. The Friedman statistic produces a conservative effect, which is undesirable [34]. The Iman-Davenports statistic constitutes a better alternative. This is distributed according to the F-distribution with $K-1$ and $(K-1)(N-1)$ degrees of freedom:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1)-\chi_F^2}. \tag{2.2}$$

If the null-hypothesis of equivalence is rejected, we can then proceed with a post-hoc test. In this work, the Holm post-hoc test has been employed to ascertain whether the best (control) algorithm performs significantly better than the remaining techniques [45].

Subsequently, the Wilcoxon paired signed-rank test has been used to to assess the statistically significant differences between each pair of over-sampling algorithms. This statistic ranks the differences in performances of two algorithms for each dataset, ignoring the signs, and compares the ranks for the positive and the negative differences. Let d_i be the difference between the performance scores of the two algorithms on i^{th} out of N datasets. The differences are ranked according to their absolute values. Let R^+ be the sum of ranks for the datasets on which the first algorithm outperforms the second, and R^- the sum of ranks for the opposite. The cases where the ranks of $d_i = 0$ are omitted from consideration:

$$R^+ = \sum_{d_i>0} rank(d_i) + \frac{1}{2}\sum_{d_i=0} rank(d_i),$$

and

$$R^{-} = \sum_{d_i<0} rank(d_i) + \frac{1}{2} \sum_{d_i=0} rank(d_i).$$

Specifically, the null hypothesis is that the median values of the two methods are equal while the alternative is that the median AUC value for our method is higher. The Bonferonni correction ensures that the overall statistical significance level is not overestimated due to the multiple tests performed.

2.2 Non-Cooperative Game Theory

Game theory is the study of strategic decision-making, or interactive decision theory. A vast array of economic, political and biological games have been discussed [71, 96, 113]. During the 1950s, many scholars extensively developed this field of study. Since then, it has been widely recognized as an important tool in many fields.

A game consists of a set of players, a set of actions (or strategies) available to those players, and a specification of payoffs for each combination of strategies. There are many types of games (e.g., cooperative, non-cooperative, symmetric, asymmetric, perfect information). Discussing all game types is outside the scope of this research. Rather, we limit our discussion to non-cooperative games, which address the interaction between individual rational decision makers. We will focus on the normal form representation of the game and only consider binary-class problems.

2.2.1 Concepts and Notation

We begin by introducing some game theory terminology and follow the notations used in Weibull [128]. The normal form of a game G is given by (I, S, π), where I denotes the set of players, S is the pure strategy action space and π its combined payoff function. More precisely, let $I = \{1, 2, ..., n\}$ where n is a positive number, each player $i \in I$ can have k pure strategies. We call S_i the set of pure strategies available to player i, which can be written as $S_i = \{1, 2, ..., k_i\}$, for some $k_i \geq 2$. A pure strategy profile s is an association of strategies to players, which is an n-tuple: $s = (s_1, s_2, ..., s_n)$, where s_i is a pure strategy for player i. The set of all possible ways in which players may choose their strategies is thus the Cartesian product $S = \times_i S_i$. For any player i and for any strategy profile s, let $\pi_i(s) \in R$ be the associated numerical payoff for player i. This payoff depends not only on the player's own strategy, but also on the strategies chosen by other players. The collection of $\pi_i(s)$ defines the i^{th} player payoff function $\pi_i : S \to R$ and the combined pure strategy payoff function is $\pi : S \to R^n$. For two players i and j with finite possible strategies, the payoff function can be represented as a $k_i \times k_j$ matrix. A strategy tuple is a unique choice of actions by each player. The best response is an action that maximizes a player's i payoff function for a given action tuple of the other players denoted by $-i$ [92]. A strategy s_i of player i is called a best response to a strategy profile s_{-i} of his opponent if: $\forall s'_i \in S_i; \pi_i(s_i, s_{-i}) \geq \pi_i(s'_i, s_{-i})$.

2.2.2 Nash Equilibrium

We call a strategy profile Nash Equilibrium [90] if each s_i is a best response to s_{-i}, that is, if: $\forall i \in \{1, 2, ..., n\} \forall s'_i \in S_i; \pi_i(s_i, s_{-i}) \geq \pi_i(s'_i, s_{-i})$. In terms of payoff function, we can re-write this simply as: $\pi_i(s_i, s_{-i}) \geq \pi_i(s'_i, s_{-i})$. This solution is self-enforcing where no player i can change his chosen strategy from s_i to s'_i to improve his payoff. This assumes that all other players remain with the strategies they have chosen in s.

Mixed strategies are a combination of strategies from which one is randomly chosen with specified probability. Players independently select strategies using a probability distribution, which leads to a probability distribution over a strategy vector. A mixed strategy for player i is the probability distribution over his set S_i of pure strategies. We can represent any mixed strategy x_i for player i as a vector x_i in k_i-dimensional Euclidean space R^{k_i}, its d^{th} coordinate $x_i^d \in R$ being the probability assigned by x_i to the player's d^{th} pure strategy, and k is the pure strategy set of any player i. Since all probabilities x_i^d for all pure strategies are positive and they sum to one, the vector $x \in R^{k_i}$ belongs to unit simplex Δ_i in k_i-space $R_+^{k_i}$ defined as:

$$\Delta_i = \{R_+^{k_i} : \sum_{d=1}^{k_i} x_i^d = 1\}.$$

A mixed strategy profile is a vector $x = (x_1, x_2, ..., x_n)$ where $x_i \in \Delta_i$ is a mixed strategy for player $i \in I$. Each mixed strategy profile is a point in the mixed strategy space of the game given by Cartesian product $\times_{i \in I} \Delta_i$. The expected value of the

payoff to player i associated with mixed strategy profile $x_i \in \Delta_i$ is given by:

$$u_i(x) = \sum_{s \in S} x(s)\pi_i(s).$$

The real number $u_i(x)$ is the i^{th} player's payoff from strategy profile x. A pure strategy can be considered as a mixed strategy that assigns probability 1 to s_i and probability 0 to all other strategies of player i. This is said to be a degenerate or extreme mixed strategy denoted by e_i^d which is a vector of length k_i. We can write $u_i(e_j^d, x_{-j})$ as the payoff that player i obtains when player j plays her d^{th} pure strategy. Hence, we can rewrite the above equation as:

$$u_i(x) = \sum_{d=1}^{k_j} u_i(e_j^d, x_{-j})x_j^d.$$

Therefore, the payoff $u_i(x)$ is the weighted sum of the payoffs that player i obtains for each j's pure strategies where the weights x_j^d are the probability assigned by x_i to the player's d^{th} pure strategy. Similar to pure strategy, a mixed strategy space $x = (x_1, x_2, ..., x_n)$ is said to be Nash Equilibrium if it is the best reply to itself, that is: $u_i(x_i, x_{-i}) \geq u_i(x_i', x_{-i})$. Unlike pure strategies where not all games have a Nash Equilibrium, a normal form game with a finite set of players and a finite set of strategies has at least one Nash Equilibrium of mixed strategies [90].

2.2.3 Replicator Dynamics

Replicator Dynamics is a process that determines how populations playing specific strategies evolve. There are different replicator dynamics depending on the evolutionary model being used [11, 12, 13]. This section will discuss replicator dynamics

equations for symmetric games and only two-player games will be considered. The fitness notion, $w_i(p)$, specifies how successful each sub-population is and must be defined for each component of p. Hence, a differential equation that governs the growth of frequencies for a symmetric game can then be defined as:

$$\dot{p} = p_i(t)(w_i\,(p) - w\,(p)). \tag{2.3}$$

Here, p_i is the fraction of members of type i in the population, $p = (p_1, p_2, ..., p_n)$ is the vector of the percentage distribution of types in the population, $w_i(p)$ is the fitness of this type, and $w(p)$ is the average payoff in the whole population.

The equation above rewards strategies that outperform the average by increasing their frequency, and penalizes poorly performing strategies by decreasing their frequency. In many situations it is not appropriate to model the frequencies as continuous functions of time. Using a discrete model, allows for the prevention of mixing between generations. The discrete dynamic must play the same role as the continuous version. Frequencies corresponding to fit strategies must increase, and those that correspond to unfit strategies must diminish. In our experiment, we applied the discrete-time version of the replicator equation derived by Weibull [128].

$$p_i(t+1) = \frac{\alpha + w_i\,(p)}{\alpha + w(p)} p_i(t) \tag{2.4}$$

where α is a small constant controlling the growth rate. Clearly, the i^{th} sub-population will grow whenever $w_i(p) > w(p)$.

2.3 Semi-Supervised Learning (SSL)

SSL is a special form of classification. Traditional classifiers use only labeled data (features/labels) to train. Labeled instances, however, are often difficult, expensive, or time consuming to obtain, as they require the efforts of experienced human annotators. Meanwhile, unlabeled data may be relatively easy to collect, but its use has proven to be fairly limited. SSL addresses this problem by using a large amount of unlabeled data, together with the labeled data, to build better classifiers. Because semi-supervised learning requires less human effort and yields higher accuracy, it is of great interest both in theory and in practice.

Semi-supervised learning methods utilize unlabeled data to either modify or reprioritize hypotheses obtained from labeled data alone. Although not all methods are probabilistic, it is easier to look at methods that represent hypotheses by $p(y|x)$, and unlabeled data by $p(x)$. There are many SSL methods, including: Expectation Maximization (EM) with generative mixture models, self-training, co-training, transductive support machines and graph-based methods. SSL is a vast and heavily researched field. While a complete survey of SSL methods is outside the scope of this paper, more information can be found in Chapelle's book [19].

Traditionally, SSL is of interest when a dataset contains both labeled and unlabeled data. SSL can potentially utilize both labeled and unlabeled data to achieve better performance than supervised learning. The key to SSL is that it allows for exploitation into the geometric structure of the data distribution. Close data points should belong to the same class and decision boundaries should lie in regions of low

data density [142].

2.4 Disease-Prediction Models

Disease-predictive modeling is the process by which a model is created in an attempt to predict the probability that a disease will occur in a given subject. In the medical field, disease- prediction models are also referred to as risk-prediction models. Most of these models are based on detection theory, which attempts to estimate the probability of an outcome given a set of collected input data from patients. Currently, there are many established disease-prediction models based on statistical regression models (e.g. logistic regression, the Weibull proportional hazard model [28], nonlinear Poisson regression [127], and the Cox proportional hazard model [18]. We will refer to these as traditional statistical methods. Traditional statistical methods are used to predict many diseases, especially cancer (e.g. colon cancer [27], endometrial cancer [23], and rectal cancer [82], as well as cardiovascular disease [28, 134, 61] and coronary heart disease [7, 8].

The baseline statistical models used in this study are based on the Cox proportional hazards model, and all risk prediction equations are derived from the Framingham Heart Study. The risk calculators tools $B^1_{m,w}$, $B^5_{m,w}$ and $B^6_{m,w}$ are available at the Framingham Heart Study webpage [43] and the risk calculator for $B^3_{m,w}$, $B^4_{m,w}$, and $B^7_{m,w}$ were developed by the University of Edinburgh [93]. Table 2.6 represents the baseline statistical models used, various risk factors and sources.

Rapid growth in technology and computer power over recent years has spurred

the emergence of a new field, machine learning (ML). Machine learning is a branch of the field of Artificial Intelligence. It is concerned with the design and development of algorithms which allow computers to evolve behaviors based on empirical data, e.g., from sensor data or databases [130]. ML can be used to capture characteristics of interest from data, including unknown underlying probability distribution. Thus, data is used as examples to illustrate relationships between observed variables.

Many machine learning algorithms have been developed throughout the years. Some of these have been used to build predictive models. The medical community has begun taking advantage of these new possibilities to create and improve existing predictive models. There have been many ML-based disease predictive models for predicting heart disease and cancer [110, 118, 114, 112, 78, 17, 31, 79, 69, 55, 99, 98].

Table 2.3: Summary of established disease prediction models in heart and cancer studies using statistical techniques.

Model	**Disease Name**	**Method**	**Year**
Harvard Cancer Risk Index [27]	Colon Cancer	Logistic Regression	2000
HCCRA [27]	Colorectal Cancer	Logistic Regression	2000
Framingham Risk Score [7, 131, 39]	Cardiovascular Disease Coronary Heart Disease Cardiovascular Disease Mortality	Cox Regression	2002
PROCAM [8]	Coronary Heart Disease Myocardial Infraction	Cox Proportional Hazard Model	2002
Imperiale [65]	Colorectal Cancer	Logistic Regression	2003
MMRpro [23]	Endometrial Cancer Colorectal Cancer	Bayesian segregation analysis	2006
SCORE [28]	Cardiovascular Disease Coronary Heart Disease Cardiovascular Disease Mortality	Weibull Proportional Hazard Model	2007
ASSIGN [134]	Cardiovascular Disease Coronary Heart Disease	Cox Proportional Hazard Model	2007
QRISK1 [60]	Cardiovascular Disease Bayes Information Criterion	Cox proportional hazard	2007
Reynolds Risk Score [106]	Cardiovascular Disease Bayes Information Criterion	Cox proportional hazard	2007
Drive [37]	Colorectal Cancer	Logistic regression	2007
QRISK2 [61]	Cardiovascular Disease Bayes Information Criterion	Cox proportional hazard	2008
Wei [127]	Colon Cancer	Nonlinear Poisson Regression	2009
Freedman [44]	Colorectal Cancer	Logistic Regression	2009
Ma *et al* [82]	Colon Cancer, Rectal Cancer	Cox Proportional Hazard Model	2010

Table 2.4: summary of established disease prediction models in heart and cancer studies using AI techniques.

Model	**Disease Name**	**Method**	**Year**
Seiwerth *et al* [110]	Throat Cancer	Decision Trees	2000
Tewari *et al* [118]	Prostate Cancer	Genetic Algorithms	2001
Snow *et al* [114]	Colorectal Cancer	Artificial Neural Network	2001
Shipp et al [112]	Lymphoma Cancer	Clustering	2002
Linkens *et al* [78]	Ovarian Cancers Childhood Leukemia Cancer Lung Cancer	Decision Trees	2003
Catto *et al* [17]	Bladder Cancer	Fuzzy Logic	2003
Delen [31]	Breast Cancer	Decision Trees, Neural Network	2004
Listgarten *et al* [79]	Breast Cancer	Support Vector Machines	2004
Kaiserman *et al* [69]	Skin Cancer	Artificial Neural Network	2005
Hayashida *et al* [55]	Esophageal	Support Vector Machines	2005
Parthiban [99]	Heart Disease	Neuro-fuzzy, genetic Algorithm	2007
Palaniappan [98]	Heart Disease	Nave Bayes	2008
Zhang [141]	Echocardiograms PimaIndians Breast Cancer Hepatitis	Support Vector Machines	2009
Patil [100]	Heart Attacks	Neural Network	2009
Subbalakshmi [116]	Heart Disease	Nave Bayes	2011

2.5 Risk Assessment for Cardiovascular Disease

Current guidelines recommend the use of risk prediction models, such as the Framingham Risk Score (FRS) to generate an estimate of cardiovascular risk based on an individual's Coronary Artery Disease (CAD) risk factors. These risk factors include: age, sex, smoking habits, Low-Density Lipoprotein (LDL) cholesterol, High-Density Lipoprotein (HDL) cholesterol, and Systolic Blood Pressure (SBP) [94, 49]. However, the limitations of such risk factors has been increasingly recognized.

Many putative novel risk factors have been proposed to improve risk discrimination beyond traditional risk scoring. These include measures of inflammation (high sensitivity C-reactive protein) [107], endothelial function (flow mediated dilation), assessment of peripheral atherosclerosis by ankle-brachial index and measurement of subclinical carotid atherosclerosis (carotid-intimal medial thickness). Coronary Artery Calcium, or CAC, is an indicator of subclinical coronary atherosclerosis detectable by non-contrast CT scan. Measurement of CAC has been demonstrated to predict cardiovascular events, and to be superior at risk reclassification beyond the FRS, as compared with other ancillary cardiac risk assessment methods [50, 51, 68].

We predicted that a novel machine learning approach to risk factor assessment would improve on the limitations of traditional risk scoring methods, such as the FRS. Additionally, we assessed the additive value of CAC measurement to the machine learning approach, and compared this to the proven risk reclassification benefit of CAC above traditional methods.

2.5.1 Risk Factors

Besides the CT scans, and as part of the baseline examination, participant centers collected information on cardiovascular risk factors, including but not limited to: family history of CHD, history of smoking, LDL, HDL, diabetes, body mass index (BMI) and hypertension. Patients were classified as having diabetes if they had ever been diagnosed as having diabetes during a hospital admission, were taking medication for diabetes, or had a history of medical treatment for diabetes.

2.5.2 Calcium Scoring

The details of the methods utilized and interpretations of the CT scanning conducted by the Multi-Ethic Study of Atherosclerosis (MESA) have been previously reported [16].

Certified technologists scanned all participants twice. All scans were read by a radiologist or cardiologist at the Los Angeles Biomedical Research Institute at Harbor-UCLA Medical Center in Los Angeles, California, to identify and quantify coronary artery calcification using a system similar to that used by Yaghoubi et. al [135]. The coronary artery calcium measurements (Agatston scores) [1] were adjusted with a standard calcium phantom that was scanned at the same time that each participant was scanned [91]. Each participant was scanned twice, and the Agatston score was calculated for each scan. The mean of the two scans was then analyzed. A detailed description of the reading protocol used in MESA has been presented elsewhere [16].

2.5.3 Follow-up

At intervals of 9 to 12 months, MESA followed up with the entire cohort via phone interviews. MESA recorded new cardiovascular events as well as any deaths that had taken place. To verify self-reported diagnoses, MESA requested copies of all medical records for hospitalization and outpatient cardiovascular diagnoses. To verify deaths, MESA obtained death certicates in addition to conducting next-of-kin interviews for all out-of-hospital cardiovascular deaths. Two physician members of the MESA mortality and morbidity review committee independently classified events and assigned incident dates. In the event of disagreement between physician members, the full committee collaborated to make the final classification. A more detailed description of the MESA follow-up methodology is available online [88].

2.5.4 Cardiovascular Events

MESA followed the cohort yearly for up to 8.5 years from baseline (median, 7.1 years) and monitored for incidence of cardiovascular events. In our study, we considered two types of recorded events: Coronary Heart Disease (CHD) events and Cardiovascular Disease (CVD) events. A CHD event was defined as: (i) myocardial infarction; (ii) angina, which includes definite angina or probable angina if followed by revascularization; (iii) resuscitated cardiac arrest; and (iv) CHD death. A CVD event was defined as a (i) CHD event; (ii) stroke; (iii) stroke death; (iv) other atherosclerotic death; or (v) other CVD death.

Table 2.5: Number of CHD events and CVD events in MESA data based on gender.

	MEN (n=3,011)		WOMEN(n=3,463)	
	No. with Events	No. without Events	No. with Events	No. without Events
CHDH	122	2,889	70	3,393
CHDA	211	2,800	108	3,355
CVDA	273	2,738	172	3,291

2.5.5 Methods

Currently, ten-year estimated risk score calculators are widely used in the mainstream cardiovascular events. These scoring systems are based on a regression model, or point scores. They are computed for each participant and available free of charge on the internet for public use. They are built given the standard Framingham risk factors, mainly age, sex, smoking, SBP, treatment for hypertension, and HDL and total cholesterol. In addition to these factors, diabetes and body mass index are also used as factors in CHDA and CVDA, respectively. A list of risk factors for each event and as well as widely used methods for prediction is summarized in Table 2.6 below.

Recently, for CHDH events, Coronary Artery Calcium, or CAC, was suggested to be a strong indicator of subclinical coronary atherosclerosis detectable by non-contrast CT scan. Measurement of CAC has been demonstrated to predict cardiovascular events, and to be superior at risk reclassification beyond the FRS, as

Table 2.6: Cardiovascular events and base models.

Baseline	Events	Risk Factors	#Factors	Source	ML Model
$B^1_{m,w}$	CHDH	Age Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) Treatment for Hypertension	6	FRS: Based on Cox Regression	$CRS^1_{m,w}$
$B^2_{m,w}$	CHDH-C	Age, Calcium Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) Treatment for Hypertension	7	FRS: Cox Regression Based	$CRS^2_{m,w}$
$B^3_{m,w}$	CHDH	Age, Diabetes Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) IVH	7	FRS: Based on Cox Regression	$CRS^3_{m,w}$
$B^4_{m,w}$	CHDA	Age, Diabetes Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) IVH	7	FRS: Based on Cox Regression	$CRS^4_{m,w}$
$B^5_{m,w}$	CVDA	Age, Diabetes Smoking Habits Treatment for Hypertension Systolic Blood Pressure (SBP) BMI	6	FRS: Based on Cox Regression	$CRS^5_{m,w}$
$B^6_{m,w}$	CVDA	Age, Diabetes Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) Treatment for Hypertension	7	FRS: Based on Cox Regression	$CRS^6_{m,w}$
$B^7_{m,w}$	CVDA	Age, Diabetes Smoking Habits Low-Density (LDL) Cholesterol High-Density (HDL) Cholesterol Systolic Blood Pressure (SBP) IVH	7	FRS: Based on Cox Regression	$CRS^7_{m,w}$

compared with other ancillary cardiac risk assessment methods. Many studies recommend that CAC be added to the list of standard risk factors [50, 51, 68].

There are multiple proposals to show the advantage of using both FRS and CAC together for prediction. First, stratifying the rates by the two levels of FRS and then evaluating the following preselected four categories of CACS: 0, 1 to 100, 101 to 300, and 301 and more. Second, re-computing the ten-year estimated risk for each participant by adding calcium to the existing risk factors and producing new scores which are also grouped into the two prediction categories: 0% to 9% and 10% and higher. We will refer to these methods for evaluations and comparison in Chapter 5 of this proposal.

2.5.6 Datasets

MESA was initiated in July, 2000 to investigate the prevalence, correlation, and progression of subclinical cardiovascular disease in individuals who were free of clinical cardiovascular disease at study entry. Details of the design and organization of MESA have been reported previously [11].

Recruitment was from six communities in the United States (Baltimore, Maryland; Chicago, Illinois; Forsyth County, North Carolina; Los Angeles, California; New York City, New York; and St. Paul, Minnesota). Self reported race and ethnicity was used to consider potential racial and ethnic differences in atherosclerosis. When the study began, MESA participants were 38% White, 28% Black, 22% Hispanic and 12% Chinese. Participants with diabetes were excluded as a CHD risk

equivalent. Diabetes was defined as a self-reported history of diabetes, hypoglycemic medication use, or fasting glucose >126 mg/dL. The full cohort included 6,814 men and women between 45 and 84 years of age. Data from 331 of these patients have been excluded from this report due to missing values for calcium scores or other risk factors. In conducting the study, we computed the Framingham Risk Scores (FRS) for all records in which they were missing and included these in our experiments. All participants gave their written informed consent and the institutional review boards at all participating centers approved the study.

Chapter 3

NEATER

3.1 Filtering of Over-Sampled Data Using Non-Cooperative Game Theory

3.1.1 Overview

We developed an approach based on game theory to filter over-sampled data. We employ SMOTE and ADASYN. SMOTE creates samples which are, for the most part, more closely related to the minority class. This causes the classifier to create larger decision regions. We use ADASYN because, in practice, this algorithm results in a balanced dataset which is more focused on those positive instances that are harder to learn. In principle, though, any oversampling method could have been used. First, we over-sample using two state-of-the-art data generation mechanisms,

SMOTE and ADASYN. Then, all data, including the original data, are considered players, and the possible class memberships are considered strategies available to all game players. However, only the synthetic data play the game to determine their class membership. In other words, every player i corresponds to a particular record in our dataset. Each player can choose between two available strategies $S_i = \{m, M\}$, where m stands for minority and M for majority. The mixed strategy profile for each player i, will lie in two-dimensional space S_i.

Our approach does not consider the synthetic data - created by SMOTE or ADASYN - as part of the minority class. Rather, it keeps the synthetic samples "unlabeled." These samples then participate in a non-cooperative game to determine their class membership. We keep all the synthetic data that end up belonging to the minority class and eliminate the data that remains, since we already have enough real majority samples. We will designate the application of NEATER to samples created by SMOTE as NEATER-SMT and to samples created by ADASYN as NEATER-ADA.

There are two types of players: (i) I_c, which denotes players that already belong to a class ($I_c = D_m \cup D_M$); they correspond to the real minority and majority samples; and (ii) I_u, which denotes unlabeled players or synthetic samples $I_u = D_y$. The set I_c is given by $\{I_{c|1}, I_{c|2}, \ldots, I_{c|k}\}$ where $I_{c|d}$ are the set of players who will always play their d^{th} pure strategy and k is the number of all possible strategies. Thus, we can say that each player of this type is playing his degenerate or extreme mixed strategy $e_i^d \in S_i$. In our settings, all datasets have only two possible classes. If the original sample belongs to the minority class $d = 1$, it will always play $e_i^m = e_i^1 = (1, 0)$ and

$e_i^M = e_i^2 = (0,1)$ if it belongs to the majority class $d = 2$. These I_c players play the game not to maximize their own payoffs, but to participate in the process where the unlabeled players I_u try to maximize their payoffs. Each I_u player will interact with a number of its neighbors I_ϕ, one neighbor at a time. Neighbors I_ϕ can be from both types of players $I_\phi \subseteq (I_c \cup I_u)$. The payoff will be the sum of all payoffs gained from each game played with each of the neighbors [66]. Each sub-game between two neighbors is called a bimatrix game [76]. In other words, the payoff of u is the sum of the payoffs that a player receives from all the two-player games that player plays with its neighbors.

In general, we specify the two-player game between two players i and j by providing a pair of payoff matrices: a $k_i \times k_j$ matrix A_{ij} and another $k_j \times k_i$ matrix A_{ji} specifying the payoffs u_i and u_j of the players for different choices of strategies followed by the two players. Usually, we can write A_{ji} as the partial payoff function between i and j. Hence, for a mixed strategy profile $x = (x_1, x_2, ..., x_n)$, the payoffs can be computed as:

$$u_i\,(x) = \sum_{j=1}^{n} (x_i^T A_{ij} x_j).$$

Since each unlabeled player can interact with neighbors of both types and there are only two strategies, minority $d = 1$ and majority $d = 2$, the payoff function for player i is:

$$u_i(x) = \sum_{j \in I_\phi \cap I_u} (x_i^T A_{ij} x_j) + \sum_{d=1}^{2} \sum_{j \in I_\phi \cap I_{c|d}} (x_i^T A_{ij} e_j^d).$$

Each player will play with number of neighbors I_ϕ, which are not necessarily of a specific type. A player may play a game with all/some/none of the players in both

sets I_u and I_c. The partial payoff matrix A_{ij} is computed as: $A_{ij} = q_{ij} \times \mathbb{I}_2$, where q_{ij} is an inverse distance weighted function, and $\mathbb{I}_2$ is the identity matrix. This can be extended to a multi-class imbalanced problem as follows:

$$u_i(x) = \sum_{j \in I_\phi \cap I_u} (x_i^T A_{ij} x_j) + \sum_{d=1}^{k} \sum_{j \in I_\phi \cap I_{c|d}} (x_i^T A_{ij} e_j^d),$$

where k represents the number of classes, and $A_{ij} = q_{ij} \times \mathbb{I}_k$ is the partial payoff matrix, where $\mathbb{I}_k$ is the identity matrix. Nash Equilibrium is computed using replicator dynamics [128, 29, 62]. Our game falls under the symmetric game type where all payers have the same set of strategies. In the next section, we will briefly discuss replicator dynamics equations for the games of this type. We will consider only two-player games.

3.1.2 Computing Nash Equilibria Using Replicator Dynamics

The fitness notion, $w_i(p)$, specifies how successful each sub-population is and must be defined for each component of p. The most studied differential equation governing the growth of a population is defined as:

$$\dot{p} = p_i(t)(w_i(p) - w(p)), \tag{3.1}$$

where p_i is the fraction of members of type i in the population, $p = (p_1, p_2, ..., p_n)$ is the vector of the percentage distribution of types in the population, $w_i(p)$ is the fitness of this type, and $w(p)$ is the average payoff in the whole population. The advantage of Eq. (3.1) is that it rewards the sub-population that outperforms the

average by increasing their percentage, and penalizes the poorly performing ones by decreasing their percentage.

In our work, we consider the discrete time version of the equations for two-player symmetric games. For each population representing a player, a vector x is constructed with the i^{th} component equal to the percentage of the corresponding sub-population. In our method, the percentages correspond to the probability of playing a strategy s from all possible available strategies. In many situations, as in our game, it is not appropriate to model the percentages as continuous functions of time. Using a discrete model allows us to prevent mixing between generations. However, the discrete time replicator dynamics equation must play the same role as the continuous version. The percentage of members which corresponds to fit strategies must increase. Those which correspond to unfit strategies must diminish. We apply the discrete-time version of the replicator equation [128] to I_u:

$$p_i(t+1) = \frac{\alpha + w_i(p)}{\alpha + w(p)} p_i(t) \tag{3.2}$$

where α is a small constant controlling the growth rate.

Clearly, the i^{th} sub-population will grow whenever $w_i(p) > w(p)$. Thus, the fitness for $w_i(x)$ is $(e_i^d A_{ij} x_j)$. This is simply the expected utility of playing the pure strategy s_i against a player with a mixed strategy defined by the vector x is $u_i(e_i^d)$. The average fitness of the population is then $w(x) = x_i^T A_{ij} x_j$ which is $u_i(x)$, as derived earlier:

$$x_i^m(t+1) = \frac{\alpha + u_i(e_i^m)}{\alpha + u_i(x(t))} x_i^m(t). \tag{3.3}$$

Since there are only two classes, minority and majority, it is sufficient to study the evolution of the minority class m, since the majority percentage necessarily falls immediately from it.

To visually observe the effect of NEATER on both SMOTE and ADASYN, we provide an example of a dataset with 359 majority examples and 39 minority examples. Fig. 3.1(a) depicts the original imbalanced data distribution, Figs. 3.1(b) and 3.1(c) show the post-SMOTE data distribution and post-ADASYN data distribution, respectively. Figures 3.1(d) and 3.1(e) show the effect of applying NEATER to the post-generated data for SMOTE and ADASYN, respectively - NEATER-SMT and NEATER-ADA. In these figures, the red x-marks represent the majority group and the blue x-marks represent the minority group (original and synthetic). From Fig. 3.1(b), we observe that SMOTE generates an equal number of instances for each minority example, which may result in generating noisy instances as well. ADASYN in Fig. 3.1(c) is very aggressive in learning from the boundary. It generates synthetic data examples very close to the decision boundary. Each approach could have two potential effects on the learning performance. It may increase the sensitivity of the minority data, as it provides a good representation of the minority data distribution. However, it may also decrease the classification performance of the majority class and thereby deteriorate the overall classification performance. NEATER aims to remove newly generated synthetic data which has a low probability of belonging to the minority class. NEATER thereby enhances sensitivity without negatively impacting

specificity (Fig. 3.1(d, e)).

3.1.3 NEATER Implementation Details

Input. Synthetic Data generated by SMOTE or ADASYN $D_y = I_u$; Number of nearest neighbors b; Number of iterations needed to converge h.

Step 1: Initialize all players i in I_u to minority class (Pr = 0.5) and majority class (Pr = 0.5).

Step 2: For each i, compute its b nearest neighbors.

Step 3: For each i interacting with each of its b neighbors j, compute the utility functions FOR $i = 1, \ldots, I_u$

$$u_i(x) = \sum_{j \in I_\phi \cap I_u} (x_i^T A_{ij} x_j) + \sum_{d=1}^{2} \sum_{j \in I_\phi \cap I_{c|d}} (x_i^T A_{ij} e_j^d),$$

where $d = 1$ is playing the minority class and $d = 2$ is the majority class.

Step 4: Compute the average payoff in the whole population $u(x) = x_i^T A_{ij} x_j$.

Step 5: Apply discrete-time replicator dynamic to study the evolution of the minority strategy probability x_i^m only, since the majority probability falls immediately from it $1 - x_i^m$. For t =1 to $t = h$

$$x_i^m(t+1) = \frac{\alpha + u_i(e_i^m)}{\alpha + u_i(x(t))} x_i^m(t)$$

if $t = h$, then stop; otherwise, increase t by 1 and go to STEP 2.

Step 6: For each player i in I_u, assign the strategy-class membership with the highest probability.

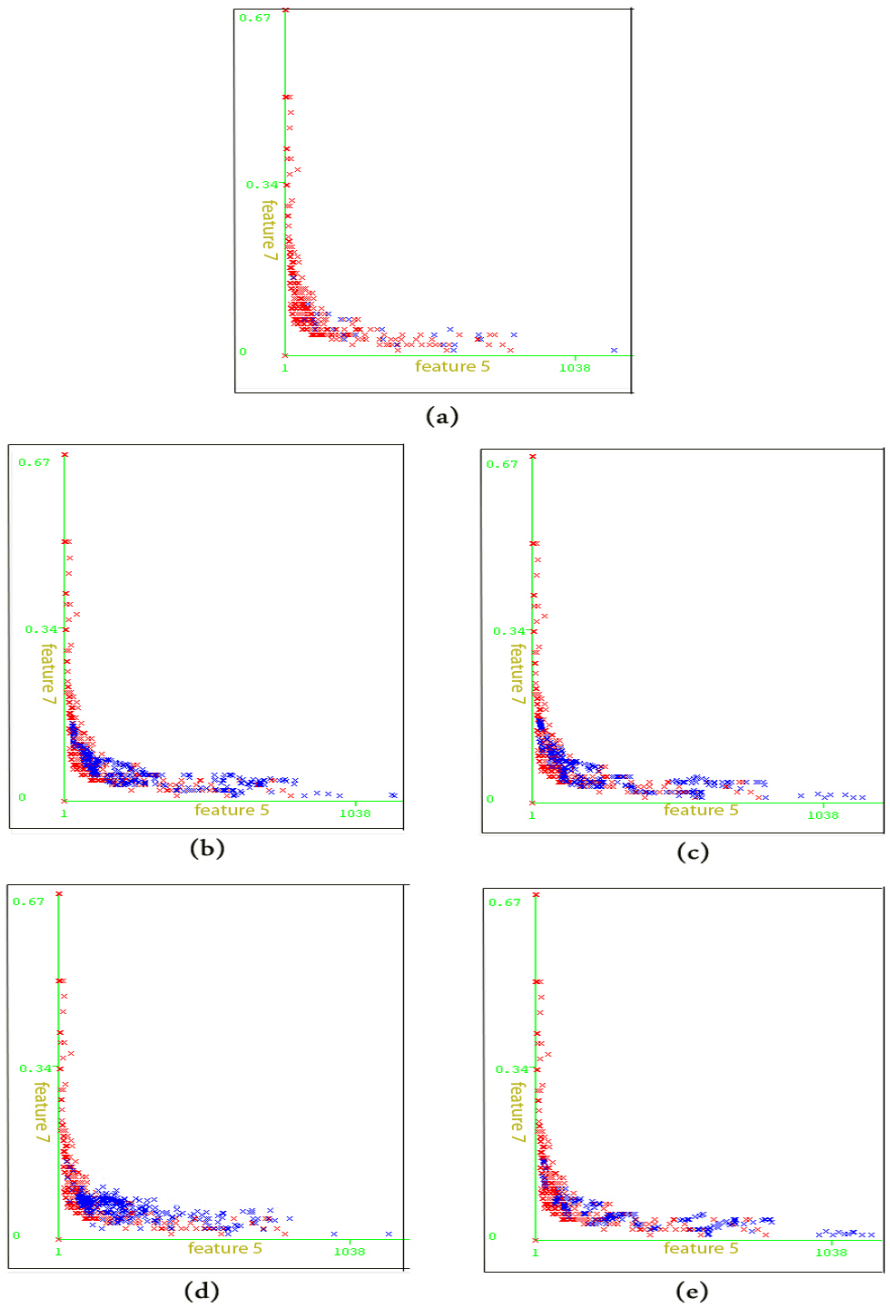


Figure 3.1: Comparison of different synthetic data generation mechanisms. (a) Original imbalanced data distribution (359 majority examples and 39 minority examples); Data distribution after: (b) SMOTE method; (c) ADASYN method; (d) NEATER-SMT method; and (e) NEATER-ADA method. The red color indicates instances that belong to the majority class while blue indicates the minority class.

3.1.4 Computational Complexity Analysis

We analyze the computational complexity of the two stages of the approach: (i) generating the initial synthetic data either by SMOTE or ADASYN, and (ii) finding Nash Equilibrium of the polymatrix game using Eq. (3). The notation that we will follow is the following: m: the dimension of the feature space; n: the number of training examples; k: the number of neighbors considered; p: the number of players; c: the number of pure strategies; and h: the number of iterations needed to converge.

The time complexity of generating the synthetic instances can be decomposed into three steps: (i) computing the Euclidean distance from the minority example under consideration to all the other examples which have a complexity $O(mn)$; (ii) sorting all current Euclidean distance calculations in ascending complexity $O(nlog(n))$; and (iii) retrieving the first k examples corresponding to the first k items in the sorted Euclidean distance set complexity $O(k)$. Thus, the time complexity for this step is $O(mn + nlog(n) + k)$. In typical situations, k and m are both significantly smaller than n, which simplifies the time complexity to $O(nlog(n))$.

The complexity of finding the equilibrium of the game using Eq. (3) is $O(chp^2)$. Since the number of classes is binary and the number of iterations can be at most $h = p$, the complexity of the game is $O(p^3)$.

NEATER's computational time in seconds cost is comparable to other existing methods during simulation. Generally, though, the runtimes costs for SMOTE-RSB and SMOTE-TK are higher than comparative algorithms. This is especially true when dealing with large datasets (i.e., "Letter"). This longer runtime is likely

attributable to the fact that SMOTE-TK repeatedly iterates across the entire data space until all Tomex links have been cleared. Similarly, SMOTE-RSB locates and removes all instances which do not belong to the rough lower approximation.

3.2 Experimental Design and Results

In our experiments, we considered two sets of data. The first contained many datasets which have been used as a benchmark in multiple sampling algorithm studies and which vary in size and imbalance ratio (Dataset Group A). In addition to the imbalanced data problem, it has become increasingly common for the number of attributes (features) to grow very large and greatly exceed the number of samples (high-dimensional data). For that reason, we also evaluated the performance of NEATER on high-dimensional data (Dataset Group B).

The few studies which have dealt with the class imbalance problem for high-dimensional data focused mostly on developing methods for feature selection [136, 86] and proposing strategies for adjusting the classifiers [2, 117], rather than balancing the data. To our knowledge the joint effect of high-dimensionality and class imbalance on classification has not been thoroughly investigated [81, 12].

3.2.1 Datasets

Dataset Group A: The 22 datasets used in our study were obtained from the UCI repository [87] and the KEEL Dataset Repository [4]. All original multi-class

datasets were first transformed into two-class problems. Specifically, if the original dataset had multiple classes, we combined all classes except one, which became the minority class (see appendix). The data from the rest of the classes were considered the majority. The percentage of minority examples varies from 2.10% (highly imbalanced) to almost 41.70% (slightly imbalanced).

Table 3.1 summarizes the main characteristics of the datasets, including the imbalance ratio (IR), and the number of attributes. It is sorted by imbalanced rates from low to high. The smallest dataset has 214 total examples (Glass-3), while the largest dataset contains 20,000 observations. These datasets vary extensively in number of attributes and class proportions, thus offering different aspects for the proposed approach.

Dataset Group B: Group B encompasses high-dimensional imbalanced data. We considered five datasets, four of which involved cancer gene expression microarray datasets [48, 6, 115, 25] and a fifth - the Madelon dataset [52] - which is a synthetic dataset. A brief description of each dataset is provided in Table 3.2. All original datasets contained two classes, with the exception of Sorlie, which contained five. The Sorlie dataset was first transformed into a two-class problem where we combined all classes except one, which became the minority class. The data from the remaining classes are considered the majority. The percentage of minority examples varies from 17.1% to almost 36.5%. The number of attributes varies from 456 to 12,533. Table 3.2 summarizes the main characteristics of the datasets, including the imbalance ratio, and the number of attributes. It is sorted by the number of attributes from low to high.

Table 3.1: Summary of imbalanced datasets used.

Dataset	#Examples	Minority	IR (%)	#Attributes
PC2	745	16	2.10	37
vehicle1	846	19	2.52	18
LetterA	20,000	789	3.95	17
Ecoli-2	336	18	5.46	7
Glass-6	214	17	6.38	9
PC1	1,109	77	6.94	21
Glass-3	214	17	7.94	10
Thyroid	7,200	576	8.00	21
yeast-3	1,484	120	8.11	8
Ecoli-3	336	27	8.19	7
SAT-4	6435	626	9.73	37
CM1	498	49	9.83	22
PC3	1,077	134	12.40	38
Segment-5	2,310	330	14.29	19
KC1	2,109	326	15.45	22
KC3	194	36	18.50	40
KC2	522	105	20.50	22
ILPD	583	167	28.60	10
Yeast-2	1,484	429	28.91	8
BCWC	569	212	37.00	32
Spambase	4,601	1,813	39.4	57
Liver Disorder	345	144	41.7	7

Table 3.2: Summary of imbalanced high-dimensional datasets used.

Dataset	#Examples	#Minority	Minority (%)	#Attributes
Sorlie	85	15	36.5	456
Madelon	1,950	650	33.6	500
Christensen	217	85	39.1	1,413
Alon	62	15	17.6	2,000
Gordon	181	31	17.1	12,533

Sorlie: Sorlie et al. (2001) examined 85 experimental samples gathered from cDNA microarrays to identify breast carcinoma based on variations in gene expression levels. The data consist of 456 cDNA clones from 427 unique genes for 78 carcinomas, three benign tumors, and four normal tissues [115].

Madelon: Madelon (2003) is a synthetic dataset containing data points grouped in 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. A number of distractor features which have no predictive power were added. The order of the features and patterns was randomized. This is a two-class classification problem with continuous input variables. The difficulty lies in that the problem is multivariate and highly non-linear [52].

Christensen: Christensen *et al.* (2009) surveyed over 200 carefully annotated human tissue samples from 10 anatosites at 1,413 CpGs associated with 773 genes to investigate tissue-specific differences in DNA methylation and to discern how aging and exposures contribute to normal variation in methylation [25].

Alon: Alon et al. (1999) have presented a dataset that contains gene expression levels of 40 tumors and 22 normal colon tissues for 6,500 human genes obtained with an Affymetrix oligonucleotide array [6].

Gordon: Gordon et al. (2002) contains 31 malignant pleural mesothelioma (MPM) and 150 lung adenocarcinoma (ADCA) tumors, a total of 181 samples. Expression levels are provided for 12,533 genes [48].

3.2.2 Experimental setup

For Group A, three different well-known classifiers were used: C4.5, Random Forest and SVM. We used the Waikato Environment for Knowledge Analysis (WEKA) implementation for the three algorithms [53].

The no-free-lunch theorem states that there is no prior superiority for any classifier system over other classifiers, so the best classifier for a particular task is itself task-dependent. However, there is a more compelling theory for the SVM that suggests it is likely to be better choice than many other approaches for many problems [133].

These different classification algorithms allowed us to compare our approach to other methods which are able to handle misclassification costs directly. We have adopted a five-fold cross validation technique to estimate the AUC measure. Each classifier has been applied to the original (imbalanced) training datasets and also to datasets that have been preprocessed by NEATER and nine state-of-the-art over-sampling techniques. ROS, SMOTE, B-SMOTE, ADASYN, and AHC are among the most widely-used methods. However, four more hybrid over-sampling algorithms have been employed for comparison purposes. These are: SMOTE-RSB, SMOTE-SL, SMOTE-ENN and SMOTE-TK. These four (along with B-SMOTE) all rely on SMOTE to generate new instances. They all also utilize cleaning methods to enhance instance quality. First, we evaluate and compare one of the NEATER versions (NEATER-SMT) to these SMOTE-based methods. Next, we evaluate and compare both versions of NEATER with the five most widely-used algorithms above. Finally,

we statistically analyze all algorithms. The Euclidean distance has been used as the distance metric with all algorithms. The number of minority neighbors has been set to five for SMOTE, B-SMOTE, ADASYN, NEATER-SMT and NEATER-ADA. The datasets have been balanced to the 50% distribution where synthetic minority data are generated until the two class distributions are approximately equal. For the implementations of these approaches, we used the KEEL data-mining software tool [4].

As to the high-dimensional datasets (Group B), we have adopted a similar experimental setup to that of Group A. We used three classifiers: C4.5, Random Forest, and SVM. Each classifier has been applied to the original (imbalanced) training datasets, and also to datasets that have been preprocessed by SMOTE and ADASYN. Next, we applied NEATER using NEATER-SMT and NEATER-ADA as the base over-sampling techniques (NEATER-SMT and NEATER-ADA). The performance of the classifiers was evaluated by the AUC, GM and AGF measures.

3.3 Discussion

Analysis for Group A: The first aim of the Group A experiments was to evaluate NEATER-SMT and compare it to the other SMOTE-based over-sampling techniques. Second, we wanted to assess whether NEATER (both versions) will properly handle the class imbalance problem, as compared to the most commonly employed state-of-the-art techniques. Third, we wanted to determine NEATER's robustness across various classifiers. Finally, we investigate which of the two versions (NEATER-SMT

or NEATER-ADA) yields the best performance in terms of the AUC metric.

Table 3.3 summarizes the average AUC, GM and AGF values across all datasets obtained with the three classifiers using the SMOTE-based over-sampling approaches. The imbalanced datasets yielded the poorest results, regardless of classifier used. For C4.5, NEATER-SMT performs significantly better than all other algorithms based in all metrics recorded. The performance of Random Forest is ranked the highest in both AUC and AGF when used with NEATER-SMT as the sampling method and second highest after SMOTE-TK in terms of GM metric. However, as per SVM, NEATER-SMT does not perform as well as other SMOTE-based algorithms.

Table 3.3: Average AUC, GM and AGF values for three different classifiers for the SMOTE-based algorithms

Algorithm	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.734	0.659	0.672	0.745	0.678	0.691	0.628	0.4	0.431
SMOTE	0.764	0.740	0.705	0.782	0.761	0.730	**0.788**	**0.781**	0.682
B-SMOTE	0.758	0.720	0.707	0.772	0.732	0.720	0.779	0.769	0.663
SMOTE-RSB	0.769	0.743	0.705	0.783	0.759	0.735	0.782	0.773	0.682
SMOTE-SL	0.732	0.706	0.700	0.769	0.735	0.742	0.784	0.777	0.687
SMOTE-ENN	0.773	0.762	0.718	0.787	0.763	0.735	0.782	0.777	**0.691**
SMOTE-TK	0.769	0.754	0.713	0.790	**0.776**	0.737	0.784	0.761	0.682
NEATER-SMT	**0.783**	**0.764**	**0.721**	**0.791**	0.768	**0.743**	0.780	0.766	0.683

Table 3.4 summarizes the average AUC, GM and AGF values across all datasets obtained with the three classifiers using both NEATER versions and the most widely used state-of-the-art over-sampling approaches. As expected, classification with the imbalanced datasets yielded the poorest results, regardless of classifier used. However, both versions of NEATER performed better than or comparably to the best

performing algorithms. Thus, it appears that using NEATER to filter over-sampled synthetic minority samples produces balanced datasets with a better representation of the underlying class distribution. This, in turn, contributes to stronger classification results, according to the AUC metric. Specifically, the performance of C4.5 is ranked highest when used with NEATER-SMT as the sampling method. Random Forests and SVM have the better performance results with NEATER-ADA. The average of GM metrics across all datasets comparably followed the AUC metric behavior. However, as per AGF, NEATER-SMT exhibits the better results across all classifiers.

Table 3.4: Average AUC, GM and AGF values for three different classifiers.

Algorithm	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.734	0.659	0.672	0.745	0.678	0.691	0.628	0.4	0.431
ROS	0.750	0.720	0.708	0.769	0.734	0.729	0.787	0.781	0.680
SMOTE	0.764	0.740	0.705	0.782	0.761	0.730	0.788	0.781	0.682
B-SMOTE	0.758	0.720	0.707	0.772	0.732	0.720	0.779	0.769	0.663
AHC	0.759	0.732	0.699	0.780	0.751	0.724	0.785	0.780	0.681
ADASYN	0.774	0.759	0.707	0.787	0.766	0.718	0.783	0.776	0.655
Neater-SMT	**0.783**	**0.764**	**0.721**	0.791	0.768	**0.743**	0.780	0.766	**0.683**
Neater-ADA	0.774	0.756	0.716	**0.792**	**0.774**	0.730	**0.790**	**0.783**	0.679

To evaluate the effect of NEATER on the oversampling method used to generate the synthetic data, we analyze the performance differences between applying NEATER-SMT and NEATER-ADA to SMOTE and ADASYN, respectively. First, when we compare NEATER-SMT to the base method SMOTE, we note that it does improve the performance across all metrics for C4.5 and Random Forest. However, there is no significant improvement for the SVM classifier. Given the computational

cost, our first observation is that NEATER-SMT is not suitable for SVM learners and further analysis is needed. Second, when we compare NEATER-ADA to the base method ADASYN, there is no difference in performance in terms of AUC and GM. However, NEATER-ADA exhibits improved performance as to the AGF measurement. On the other hand, there are improvements in all metrics for the both Random Forest and SVM when applying NEATER to ADASYN, as indicated in Table 3.4. The measurements of these metrics for each dataset used in the experiments are available in the appendix.

The Friedman's average ranks for the three classifiers are depicted in Fig. 3.4. This serves as further confirmation of the findings with regard to the AUC. For the C4.5, NEATER-SMT clearly arises as the over-sampling algorithm with the lowest ranking, that is, the highest performance on average. This is followed by SMOTE-ENN, NEATER-ADA and ADASYN. For the Random Forest and SVM classifiers, NEATER-ADA is the technique with the best ranking, followed by SMOTE-TK, NEATER-SMT and ADASYN for the Random Forest, and SMOTE-SL, ROS and SMOTE for SVM. Imbalanced datasets produced the highest average ranks (worst performance) with all classifiers.

With the aim of investigating whether our first conclusion can be supported by non-parametric statistical tests, the Iman-Davenport statistic has been computed using Eq. 5 to discover whether or not the AUC results are significantly different. This computation yielded $F_F = 15.91$ for C4.5, $F_F = 6.42$ for Random Forest, and $F_F = 3.34$ for SVM. As the critical values for the F-distribution with K - 1 = 12 - 1 = 11 and (K - 1)(N - 1) = (12 - 1)(22 - 1) = 231 degrees of freedom at confidence levels

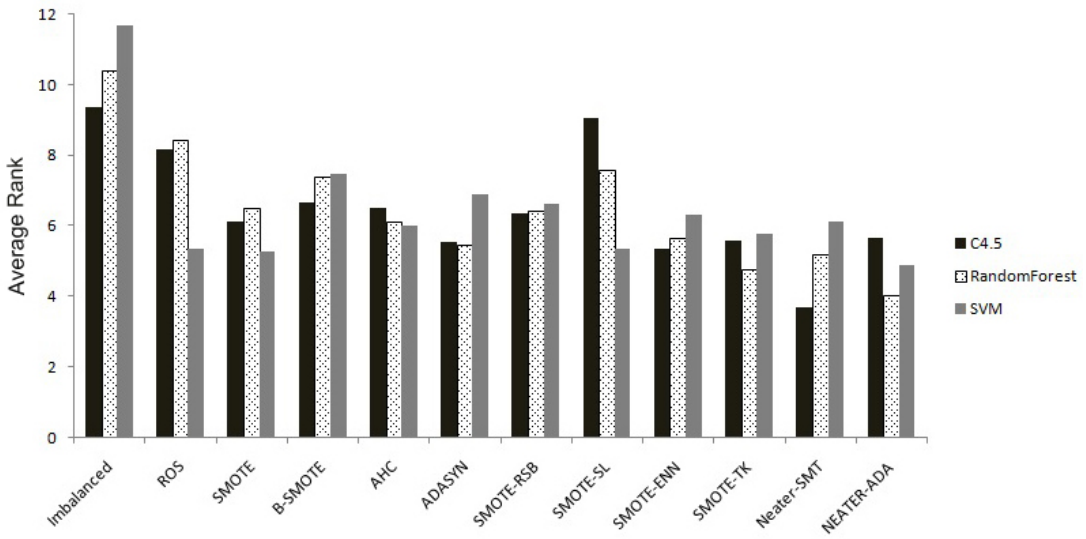


Figure 3.2: Friedman's average ranks for the three classifiers.

of 90 and 95% are $F(11, 231)_{0.90} = 1.60$ and $F(11, 231)_{0.95} = 1.83$, the null-hypothesis that all strategies explored here perform equally well can be rejected. Consequently, we can now continue with a Holm post hoc test, using the best over-sampling method for each classifier as the respective control algorithm.

Table 3.5 presents the z-values, the p-values and the adjusted αs calculated using the Holm procedure, where the symbol "**" indicates that the null-hypothesis of equivalence with the control algorithm is not rejected at a significance level of $\alpha = 0.05$ (no significant difference between the two methods).

The results in Table 3.5 of the Holm test reveal the superiority of NEATER-SMT over all approaches with the C4.5 classifier except for SMOTE-ENN, NEATER-ADA and ADASYN. On the results for the Random Forest classifier, one can observe that

Table 3.5: Results obtained with the Holm test for $\alpha = 0.05$.

i	Algorithm	z	p-value	α/i
C4.5 (Neater-SMT is the control method)				
11	Imbalanced	-6.3567	0	0.004545
10	SMOTE-SL	-5.8739	0	0.005000
9	ROS	-4.4375	0.00090	0.005556
8	B-SMOTE	-3.6425	0.00134	0.006250
7	AHC	-3.5760	0.00388	0.007143
6	SMOTE-RSB	-2.7288	0.00481	0.008333
5	SMOTE	-2.9811	0.00732	0.010000
4	SMOTE-TK	-2.7656	0.01184	0.012500
3	SMOTE-ENN**	-2.0130	0.04444	0.016667
2	Neater-ADA**	-1.7379	0.08186	0.025000
1	ADASYN**	-1.5120	0.13104	0.050000
Random Forest (Neater-ADA is the control method)				
11	Imbalanced	-5.9798	0	0.004545
10	ROS	-4.7520	0	0.005000
9	SMOTE-SL	-4.4871	0	0.005556
8	B-SMOTE	-3.2293	0.00124	0.006250
7	SMOTE-RSB	-3.0533	0.00403	0.007143
6	AHC	-2.4361	0.00400	0.008333
5	SMOTE	-2.103	0.00957	0.010000
4	ADASYN**	-1.9786	0.03758	0.012500
3	SMOTE-ENN**	-1.3059	0.04020	0.016667
2	Neater-SMT**	-0.8587	0.38978	0.025000
1	SMOTE-TK**	-0.1678	0.65028	0.050000
SVM (Neater-ADA is the control method)				
11	Imbalanced	-3.9798	0	0.004545
10	B-SMOTE	-2.8028	0.00357	0.005000
9	ADASYN	-2.7121	0.00444	0.005556
8	SMOTE-RSB**	-1.2573	0.21130	0.006250
7	AHC**	-1.2506	0.21410	0.007143
6	Neater-SMT**	-1.1760	0.23800	0.008333
5	SMOTE-TK**	-1.1089	0.26700	0.010000
4	SMOTE-ENN**	-1.0436	0.29834	0.012500
3	ROS**	-0.6952	0.48392	0.016667
2	SMOTE-SL**	-0.2223	0.82588	0.025000
1	SMOTE**	-0.0187	0.98404	0.050000

the control algorithm NEATER-ADA is better than all algorithms, but it is equivalent to NEATER-SMT, ADASYN, SMOTE-ENN and SMOTE-TK. Finally, with SVM, NEATER-ADA performs significantly better than B-SMOTE and ADASYN, but behaves equally as well as all other algorithms. As expected, use of the imbalanced datasets without any preprocessing produces the worst results.

Several algorithms exhibit similar behaviors, especially with the Random Forest and SVM classifiers. We have therefore run Wilcoxon's test between each pair of techniques for each classification model. The upper diagonal halves of Tables 3.6-3.8 summarize this statistic for a significance level of $\alpha = 0.01$ (1% or less chance). The lower diagonal halves correspond to a significance level of $\alpha = 0.05$. The symbol "$\bullet$" indicates that the method in the row significantly outperforms the method in the column. The symbol "$\square$" indicates that the method in the column performs significantly better than the method in the row.

With the C4.5 classifier, ADASYN, SMOTE-ENN and NEATER-ADA perform significantly better than ROS and SMOTE-SL at both significant levels, and better than B-SMOTE at $\alpha = 0.05$. The most notable observation from Table 3.6 is that NEATER-SMT performs significantly better than all the state-of-the-art algorithms at significant level $\alpha = 0.05$, except for ADASYN, and half of the algorithms at significant level $\alpha = 0.01$. This demonstrates this algorithm's ability to consistently produce well-balanced training sets for further classification with the C4.5 decision tree.

In the case of Random Forest, SMOTE-TK and NEATER-SMT are both significantly better than ROS, AHC, SMOTE-RSB and SMOTE-SL for $\alpha = 0.05$. On the

Table 3.6: Summary of the Wilcoxon statistic for the over-sampling algorithms with C4.5 classifier. Upper and lower diagonal halves are generated for $\alpha = 0.01$ and $\alpha = 0.05$, respectively.

		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
(a)	Imbalanced	-		□	□		□	□		□	□	□	□
(b)	ROS	•	-				□			□		□	□
(c)	SMOTE	•	•	-					•			□	
(d)	B-SMOTE	•			-								
(e)	AHC	•				-			•			□	
(f)	ADASYN	•	•		•		-		•				
(g)	SMOTE-RSB	•	•					-	•				
(h)	SMOTE-SL						□		-	□	□	□	□
(i)	SMOTE-ENN	•	•		•	•			•	-			
(j)	SMOTE-TK	•	•		•				•		-		
(k)	NEATER-SMT	•	•	•	•	•		•	•	•	•	-	
(l)	NEATER-ADA	•	•		•				•				-

Table 3.7: Summary of the Wilcoxon statistic for the over-sampling algorithms with the Random Forest classifier. Upper and lower diagonal halves generated are for $\alpha = 0.01$ and $\alpha = 0.05$, respectively.

		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
(a)	Imbalanced	-	□	□	□	□	□	□	□	□	□	□	□
(b)	ROS	•	-	□			□			□	□	□	□
(c)	SMOTE	•	•	-									
(d)	B-SMOTE	•			-								□
(e)	AHC	•				-							□
(f)	ADASYN	•	•		•		-		•				
(g)	SMOTE-RSB	•	•		•			-					
(h)	SMOTE-SL	•							-		□		□
(i)	SMOTE-ENN	•	•		•				•	-			
(j)	SMOTE-TK	•	•		•			•	•		-		
(k)	NEATER-SMT	•	•		•			•	•			-	
(l)	NEATER-ADA	•	•	•	•	•	•	•	•				-

other hand, NEATER-ADA is significantly better than all other algorithms, except for NEATER-SMT, SMOTE-ENN, and SMOTE-TK at $\alpha = 0.05$. Finally, for $\alpha = 0.01$, the NEATER-ADA approach is significantly superior to AHC, B-SMOTE, ROS ans SMOTE-SL. When using the SVM, Table 3.8 indicates that there are less statistically significant differences than in the previous case of the Random Forest algorithm. Nonetheless, the NEATER-ADA algorithm performs significantly better than B-SMOTE and ADASYN at significance level $\alpha = 0.05$. All the oversampling algorithms perform significantly better than the original data at both significance levels.

As a summary of Wilcoxon's tests for an easier analysis, the three values in the cells of Table 3.9 denote how many times each method has been significantly-better / same / significantly-worse than the rest of the over-sampling strategies at significance levels of $\alpha = 0.01$ and $\alpha = 0.05$ for each classifier. The results reported here corroborate the discussion of the previous tables, proving the practical relevance of over-sampling the minority class, irrespective of the classification model (using the imbalanced set is significantly worse than employing a training set that has been preprocessed by some over-sampling algorithm). This summary also allows us to clearly state the overall superiority of the NEATER-SMT and NEATER-ADA algorithms over the remaining methods, especially with the C4.5 and Random Forest classifiers, respectively.

Analysis for Group B: The goals of the Group B experiments were first to assess whether the well-known sampling techniques perform as well on high-dimensional datasets as they do on low-dimensional datasets and, second, to determine which

Table 3.8: Summary of the Wilcoxon statistic for the over-sampling algorithms with SVM classifier. Upper and lower diagonal halves are generated for $\alpha = 0.01$ and $\alpha = 0.05$, respectively.

		(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)	(j)	(k)	(l)
(a)	Imbalanced	-	□	□	□	□	□	□	□	□	□	□	□
(b)	ROS	•	-		•								
(c)	SMOTE	•		-									
(d)	B-SMOTE	•			-								
(e)	AHC	•				-							
(f)	ADASYN	•					-						
(g)	SMOTE-RSB	•						-					
(h)	SMOTE-SL	•			•				-				
(i)	SMOTE-ENN	•			•					-			
(j)	SMOTE-TK	•			•						-		
(k)	NEATER-SMT	•										-	
(l)	NEATER-ADA	•			•		•						-

Table 3.9: Summary of how many times the over-sampling techniques have been significantly-better/same/significantly-worse.

	C4.5		RF		SVM	
	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$	$\alpha = 0.05$	$\alpha = 0.01$
Imbalanced	0/1/10	0/3/8	0/0/11	0/0/11	0/0/11	0/0/11
ROS	1/3/7	0/7/4	1/3/7	1/4/6	2/9/0	1/10/0
SMOTE	3/7/1	1/9/1	2/8/1	2/9/0	1/10/0	1/10/0
B-SMOTE	1/5/5	1/10/0	1/4/6	1/9/1	1/6/4	1/10/0
AHC	2/7/2	0/10/1	1/9/1	1/9/1	1/10/0	1/10/0
ADASYN	4/7/0	3/8/0	4/6/1	2/9/0	1/9/1	1/10/0
SMOTE-RSB	3/7/1	1/10/0	4/6/1	1/10/0	1/10/0	1/10/0
SMOTE-SL	0/3/8	0/6/5	1/5/5	1/8/2	2/9/0	1/10/0
SMOTE-ENN	5/5/1	3/8/0	4/7/0	2/9/0	2/9/0	1/10/0
SMOTE-TK	4/6/1	2/9/0	5/6/0	3/8/0	2/9/0	1/10/0
NEATER-SMT	6/5/0	5/6/0	5/6/0	2/9/0	1/10/0	1/10/0
NEATER-ADA	4/7/0	3/8/0	8/3/0	5/6/0	3/8/0	1/10/0

Table 3.10: Average AUC, GM and AGF values on high-dimensional datasets for three different classifiers.

Algorithm	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.755	0.737	**0.716**	0.762	0.718	0.726	0.820	0.810	0.807
SMOTE	0.754	0.737	0.708	0.765	0.752	0.755	0.821	0.817	0.779
ADASYN	0.738	0.723	0.692	0.795	0.779	0.756	0.818	0.814	0.809
Neater-SMT	**0.756**	**0.738**	0.710	0.788	0.776	0.778	**0.823**	**0.818**	**0.814**
Neater-ADA	0.748	0.732	0.702	**0.802**	**0.786**	**0.793**	0.820	0.815	0.811

classifier benefits the most from these techniques. Finally, we sought to assess the effectiveness of NEATER on high-dimensional datasets and determine which of its two versions yielded the best performance in terms of the AUC, GM and AGF metrics (Table 3.10).

When we applied SMOTE, ADASYN, NEATER-SMT and NEATER-ADA to high-dimensional class imbalanced data, our main findings from the analysis were: (i) in the low-dimensional setting, SMOTE and ADASYN are efficient in reducing the class-imbalance problem for most classifiers; (ii) SMOTE has hardly any effect on most classifiers trained on high-dimensional data, while ADASYN proves beneficial for Random Forest classifier; (iii) applying NEATER to SMOTE and ADASYN results in more well-defined datasets, as demonstrated by the performance improvement in all classifiers with and without NEATER applied; and (iv) NEATER-SMT and NEATER-ADA yield significant improvements on high-dimensional data with the Random Forest classifier, as compared to other classifiers.

Our results are in agreement with the finding that SMOTE and ADASYN had little or no effect on many classifiers when data were high-dimensional. [12]. Even

though one of the NEATER versions has the highest AUC measure of all classifiers, there is no significant effect on the C4.5 and SVM classifiers. In practice, only Random Forest classifiers seem to benefit substantially from the use of NEATER in the high-dimensional setting.

Akbani [3] explains how SVM has been extensively studied and has shown remarkable success in many applications. However, the success of SVM is very rare when it is applied to the problem of learning from imbalanced datasets. SVM performs well with moderately imbalanced data, even without any modifications. Its unique learning mechanism makes it an interesting candidate for dealing with imbalanced datasets, since SVM only takes into account those instances that are close to the boundary, i.e. the support vectors, for building its model. This means that SVM is unaffected by non-noisy negative instances far away from the boundary, even if they are numerous.

SMV is not the best choice when under-sampling is applied to the training data to balance the data. The problem is that SVM is designed to estimate the probability distribution of the target population. Since that distribution is unknown, we try to estimate the population distribution using a sample distribution. As long as the sample is drawn randomly, the sample distribution can be used to estimate the population distribution from where it was drawn. By learning the sample distribution we can learn to approximate the target distribution. However, once under-sampling is performed on the majority class, the sample can no longer be considered random. The second problem with under-sampling is that valid instances form the majority class which contain valuable information are discarded. The nature of the information

these instances contain can be understood in the following way. The problem with imbalanced datasets is that they skew the boundary towards the positive instances. Any hyper-plane can be defined by its orientation, given by the direction of w, and its distance from the origin. The task of SVM is to learn the optimal hyper-plane in the feature space. In order to do this, it takes cues from the dataset about the orientation and distance of the optimal hyper-plane. From a relatively noise-free but imbalanced dataset that is linearly separable in the feature space, SVM will learn to approximate the orientation of the hyper-plane better than using the same dataset after it is under-sampled [3].

Three of the five datasets exhibited only minor changes when the oversampling techniques were applied (Tables 3.13 - 3.15). This could be due to either (i) the dataset already being well defined; or (ii) the sample size being too small to show any major effect.

Remarks: NEATER has several significant advantages over all other approaches in this paper: (1) NEATER does not operate on prior assumptions. New synthetic examples are not assigned to a particular class until after equilibrium has been reached. Thus, the new minority samples are actually more representative of the minority class; (ii) With NEATER, the game outcome determines an instance's membership in a more consistent and robust manner. For example, while other over-sampling methods may aid the classifier in providing higher accuracy for the minority class, they are less concerned with accuracy of the majority class, which in turn, suffers. Conversely, NEATER is able to attain high accuracy for the minority class without jeopardizing majority class accuracy; and (iii) NEATER is able to successfully

obtain representative synthetic instances while generating a very small degree of noise. In contrast, all of the other algorithms are only able to accomplish one of these objectives well, but are unable to combat both the representative synthetic data issue as well as the noise problem. For example, SMOTE and ADASYN both work to create representative synthetic data, but they also both generate a high level of noise, and their decision boundary is greatly influenced. On the other hand, B-SMOTE, SMOTE-TK, SMOTE-ENN and SMOTE-SL are all able to keep noise levels down and only have minimal impact on the decision boundary, but they fail to create many representative synthetic instances and ignore the interior instances. Thus, these techniques perform well when using the SVM classifier, but yield weaker results when matched with any other classifier.

For complete analysis, we also ran some experiments with a cluster based model approach, Expectation-Maximization (EM) [32]. It is an iterative method for finding maximum likelihood estimates of parameters in statistical models, where the model depends on unobserved latent variables. EM does not perform well when the dataset is imbalanced. It has improved the classification from the original imbalanced datasets only slightly, and only surpasses one other sampling algorithm. All other algorithms still perform better. For example, EM implementations [109] to classify the imbalanced datasets, Glass-3 dataset, result in AUC measurements of 0.665, 0.577 and 0.500 for C4.5, Random Forest and SVM respectively; GM measurements of 0.665, 0.585 and 0.500 and AGF measurements of 0.611, 0.560 and 0.500, which is better than B-SMOTE but still does not perform the remaining sampling algorithms including NEATER.

The outcome of this algorithm - class membership probability - also has potential for further future use: (i) the probabilities can be passed to the classifier as additional input to enhance performance, and (ii) the designer has the ability to choose a different threshold parameter for each separate application domain. In other words, it allows for flexibility in choosing how conservative the classifier ought to be. Such an approach may be beneficial for problems, such as medical diagnoses, where predicting a false negative - a Type II error - could potentially be much more serious than diagnosing a false positive-Type I error.

Chapter 4

US-SSL

4.1 A Novel Framework for Handling Imbalanced Data in Supervised Learning: A Semi-Supervised Learning Approach

Traditionally, SSL is of interest when a dataset has both labeled and unlabeled data. It can potentially utilize both labeled and unlabeled data to achieve better performance than supervised learning. The key to SSL is that it allows for exploitation into the geometric structure of the data distribution. Close data points should belong to the same class and decision boundaries should lie in regions of low data density [142].

4.1.1 Data Pre-processing Procedure

The targeted datasets are all initially labeled, but they are yet imbalanced. With US, we create unlabeled data by stripping the labels from the majority class instances. As such, the problem is transformed from supervised to semi-supervised. It is then solved to identify and remove borderline instances, especially those which overlap largely with the minority instances (Fig. 4.1).

4.1.2 Empowering Imbalanced Data in Supervised Learning: A Semi-Supervised Learning Approach (US-SSL)

US and SSL can each be viewed as possessing the common goal of drastically compressing data without losing the underlying information. All learning strategies must therefore be based on a belief in the hidden inherent simplicity of relationships $P(A|B)$. Our method will take advantage of this concept to under-sample the data using SSL. We will refer to this technique as US-SSL. In this method, removing overlapping examples establishes well-defined class clusters in the training set, which leads to well-defined classification rules.

SSL: We will extend three algorithms from the existing paradigms to demonstrate the effectiveness of our approach: Local and Global Consistency (LGC) [142], Yet Another Two-Stage Idea (YATSI) [36], and Semi-Supervised Learning via Random Forests (SSLRF) [75].

(i) LGC: A graph-based approach where the graph G is fully connected, with

no self-loop. The edges of G are weighted with a positive and symmetric function w which represents a pairwise relationship between the vertices. The key point of the method is to let every point iteratively spread its label information to its neighbors until a global state is reached. The weights are scaled by a parameter σ for propagation. During each iteration, each point receives the information from its neighbor and also retains its initial information. A parameter α allows for adjustment of the relative amount of information provided by the neighbors and the initial point. When convergence is reached, each unlabeled point is assigned the label of the class it has received the most information for during the iteration process [142].

(ii) YATSI: An algorithm that uses one classifier for labeling the test data after training on the training set. In the initializing step, all instances from the test set have a weight of 0. In each subsequent step, they get a weight of *current step / number of steps*. This implies that all provided classifiers need to be able to handle weighted instances [36].

(iii) SSLRF: A collective classifier which uses Random Trees to build predictions on the test set. It divides the test set into folds and successively adds the test instances with the best predictions to the training set. The first iteration trains solely on the training set and determines the distributions for all the instances in the test set. From these predictions the best are chosen (this number is the same as the number of instances in a fold). From then on, the classifier is trained with the training file from the previous run plus the best instances determined during the previous iteration [75].

4.1.2.1 US-SSL

Consider the dataset D which contains all minority examples and all remaining majority examples after the redundant, borderline and noisy examples have been removed. All examples in D are labeled. Divide D into k equal sets of size $1/k$. Strip the labels from one of these sets and use the remaining $k-1$ sets to relabel it via an SSL algorithm. Specifically, we have k datasets of size X where $X = (X_l, X_u)$ of labeled examples $X_l = \{x_1, ..., x_l\}$ and unlabeled examples $X_u = \{x_{l+1}, ..., x_n\}$, along with the corresponding class labels $\{y_1, ..., y_l\}$, where y_i in our settings has two possible classes: either positive (minority) or negative (majority). After a semi-supervised method is applied in each dataset X_1 to X_k, all examples in D are then labeled. The safe negative examples are more likely to keep their labels. The examples in the overlapping regions between classes, and those located farther from the decision boundary, will be relabeled as positive examples if they are closer to the minority examples or lie in regions in decision boundaries of low minority density. These mislabeled majority examples are removed from D. The resulting set T is used as the training set for the corresponding imbalance problem (Fig. 4.1). If further under-sampling is desired to achieve a specific balance ratio, RUS is then used to remove additional majority examples.

4.1.3 US-SSL Implementation Details:

1. Let S be the original training set.

2. Remove redundant examples and all negative examples participating in Tomek

links (this removes those negative examples that are believed to be borderline and/or noisy). The resulting set D will have the remaining negative examples and all positive examples.

3. Divide D into k equal-sized sets.

4. Iteratively strip the labels of $1/k$ set and create a dataset X, where $k-1$ sets are labeled and one set is unlabeled.

5. Apply a semi-supervised algorithm to relabel the stripped label set.

6. Remove all majority examples that were relabeled as minority, and generate a new training set T which will be used to build the prediction model for the imbalance problem.

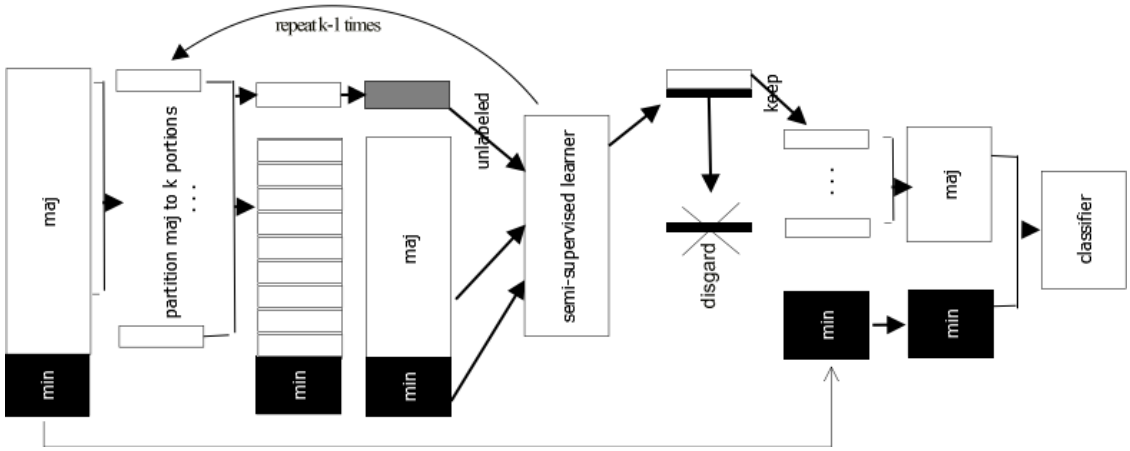


Figure 4.1: Illustration of the US-SSL algorithm.

4.1.4 Framework for US-SSL Method Selection

As discussed above, there are many US methods which have been presented in recent years. Most of these studies report the average results among a set of imbalanced datasets. However, looking at each set individually, it is clear that there is no method which performs best as compared to all other methods for all datasets. Therefore, given an imbalanced dataset, can we ask the following questions: Which under-sampling method should be used? Which method is the best?

There is no direct answer. As reported by many studies and also as apparent from the extensive analysis we presented in our proposed method [5], there is no method that outperforms all others in all given datasets. In this part of the paper, we will present how our framework uses SSL to help determine the most suitable way to select a US method for a given dataset.

In semi-supervised learning, one should use a method whose assumptions fit the problem structure. Determining such a method, though, is a very challenging task. However, as a general rule, manifold-based algorithms should be used for manifold-like datasets. Similarly, cluster-based algorithms should be used for cluster-like datasets. We will use this rule to choose the proper SSL method for our US-SSL framework. Thus, the datasets can be divided into two groups: Manifold-like and Cluster-like.

We extended three algorithms from the existing paradigms to demonstrate the effectiveness of our approach: Local and Global Consistency (LGC) [142], Graph

Transduction via Alternating Minimization (GTAM) [125], and Transductive Support Vector Machine (TSVM) [67]. For the purpose of our framewrok, there were many possible algorithms to choose from [19]. We chose these specific manifold and cluster-based algorithms since they are well-known, widely used and considered to be elite methods in their respective categories:

(i) LGC: A graph-based approach where the graph G is fully connected, with no self-loop. The edges of G are weighted with a positive and symmetric function w which represents a pairwise relationship between the vertices. The key point of the method is to let every point iteratively spread its label information to its neighbors until a global state is reached. The weights are scaled by a parameter σ for propagation. During each iteration, each point receives the information from its neighbor and also retains its initial information. A parameter α allows for adjustment of the relative amount of information provided by the neighbors and the initial point. When convergence is reached, each unlabeled point is assigned the label of the class it has received the most information for during the iteration process [142].

(ii) GTAM: This method aims to deal with some of the limitations of popular SSL methods like LGC, which are sensitive to the initial set of labels provided by the user. It is a propagation algorithm that more reliably minimizes a cost function over both a function on the graph and a binary label matrix. It is designed to propagate the initial labels while performing optimization over both label variables, and is resilient to label imbalances [125].

(iii) TSVM: TSVM is an extension of the SVM method (also known as semi-supervised SVM, S3VM). TSVM uses unlabeled data to find the decision boundary

with the largest margin between classes. Unlike SVM, TSVM tries to maximize the margin with a linear boundary by considering both labeled and unlabeled instances, which might deliver a lower number of generalization errors [123]. The unlabeled data drive the decision boundary away from dense regions [143]. However, if the dense regions are overlapping, TSVM might not find the correct decision boundary between such regions (clusters).

Manifold-based algorithms: 'manifold assumption' assumes that the true structure of the data lies in a low-dimensional manifold embedded in the high-dimensional data space. Such a manifold assumption would deliver better estimates and similarity measures about the data. LGC is an algorithm that belongs to this category and has demonstrated impressive performance on relatively complex manifold structures.

Cluster-based algorithms: most cluster-based approaches attempt to find a low-density region to separate classes, avoiding placing the decision boundary inside clusters (cutting through high-density regions). TSVM [25] is a typical example.

Given these SSL algorithms and the general rule for algorithm selection, we predict that the cluster-like datasets will have a well-defined training set, and therefore yield better classification performance if they are preprocessed using the cluster-based algorithm (TSVM) in the US-SSL framework. Similarly, this should hold true for the manifold-like datasets when we use either manifold method such as LGC or GTAM as the base SSL algorithm in the US-SSL framework. To better observe the effect of US-SSL, we provide an example of a training dataset (Noisy Two-Moon) with 231 majority examples and 100 minority examples (Fig. 4.2). The blue circles represent the majority group, and the red squares represent the minority group. (a)

is the original dataset, (b) - (f) are the top 5 algorithms in terms of performance (not ordered). The proposed US-SSL variations - (d), (e) and (f) - always perform in the top three among all classifiers tested. Figure 4.3 demonstrates the overall process of the US-SSL framework where we start by determining the structure of the imbalance dataset. Then we choose the SSL algorithm to use for the US step. After, we apply the US-SSL algorithm to balance and refine the dataset to establish a well-defined training set.

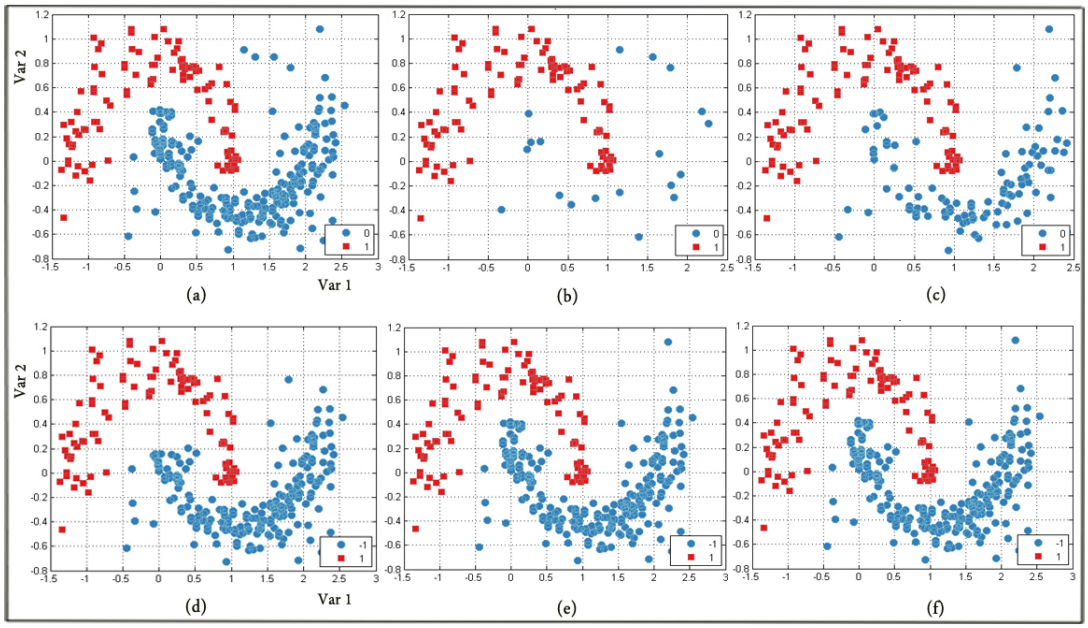


Figure 4.2: Comparison of different US mechanisms (the blue circles represent the majority group and the red squares represent the minority group): (a) original imbalanced data distribution; (b) data distribution after OSS method; (c) data distribution after RUS method; (d) data distribution after US-TSVM method; (e) data distribution after US-LGC method; and (f) data distribution after US-GTAM method.

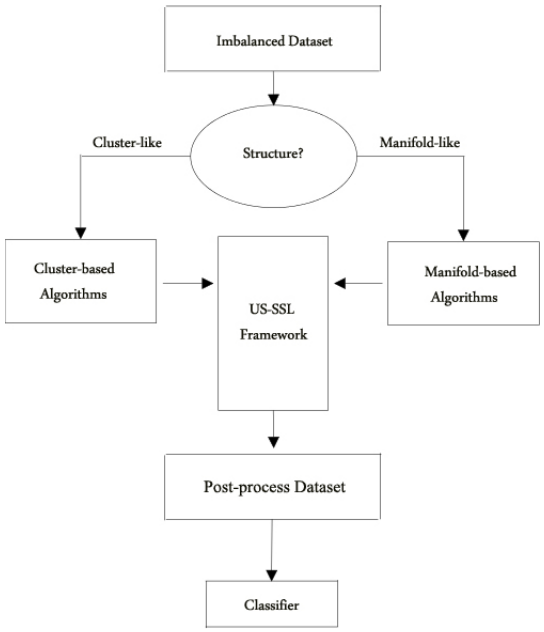


Figure 4.3: Illustration of the US-SSL framework.

4.2 Experimental Design and Results

4.2.1 Datasets

Dataset Group A:

We obtained the fourteen datasets used in our empirical study from the UCI repository [89]. All original multi-class datasets were first transformed into two-class problems. Table 4.1 summarizes the characteristics of the datasets, where (A) is the number of samples, (B) is the number of minority samples, (C) is the imbalance ratio and (D) is the number of attributes.

Dataset Group B: We obtained eight datasets. Six are the benchmark datasets used in "*Semi-Supervised Learning*" by Olivier Chapelle [19]. Three of these six were

Dataset	A	B	C	D
Glass-0	214	4	2.06	9
Ecoli-1	336	11	3.36	7
LetterA	20,000	789	3.95	17
Thyroid-2	215	10	4.92	5
PC1	1,109	77	6.94	21
Glass-3	214	17	7.94	10
Yeast-3	1,484	120	8.11	8
Ecoli-3	336	27	8.19	7
Page-blocks	5,472	479	8.77	10
Satimage-4	6,435	626	9.73	37
CM1	498	49	9.83	22
Vehicle1	846	181	21.3	18
ILPD	583	167	28.6	10
Liver Disorder	345	144	41.70	7

Table 4.1: Imbalanced datasets.

artificially created in order to create situations that correspond to certain assumptions. This was done to allow relation of the performance of the algorithms to those assumptions. The five other datasets were derived from real data. Thus, presumably, the performance on these is indicative of the performance in real applications. All original datasets contained two classes and they were all originally balanced, except for SecStr and Breast Cancer, which were slightly imbalanced: 42.8% and 37.2%, respectively. Each of these datasets was created or altered to make the learner's task more difficult (details below).

For example, to prevent the experimenters from using domain knowledge, the author tried to obscure structure in the data by shuffling the pixels in the images, and the same number of dimensions (241) and points (1500) for most datasets were used in an attempt to obscure the origin of the data and to increase the comparability of the results. This was done in an attempt to evaluate the power of the presented algorithms themselves in the most neutral manner possible. In particular, in some cases taking advantage of domain knowledge should be avoided, which is not the case for others.

We have added to these given difficulties by making the dataset highly imbalanced ($< 20\%$ IR) and severely imbalanced ($< 10\%$ IR). A brief description of each dataset is provided below. Table 4.2 summarizes the main characteristics of the datasets, including the imbalance ratio, and the number of attributes. It is sorted by the number of attributes from high to low. The datasets are as follows:

g241c: This dataset was generated such that the cluster assumption holds, (i.e., the classes correspond to clusters), but the manifold assumption does not and all

Table 4.2: Basic properties of imbalanced benchmark datasets.

Dataset	#Features	Classes	#Examples	
Digist1	241	2	1,020	
USPS	241	2	1,020	
G241c	241	2	1,020	
G241d	241	2	1,020	IR = 18%
BCI	117	2	272	
Breast	30	2	386	
SecStr	15	2	56,901	
Two-moon	2	2	476	
Digist1	241	2	885	
USPS	241	2	885	
G241c	241	2	885	
G241d	241	2	885	IR = 09%
BCI	117	2	236	
Breast	30	2	335	
SecStr	15	2	49,370	
Two-moon	2	2	413	

dimensions are standardized, which means that they are shifted and re-scaled to zero-mean and unit variance. A two-dimensional projection of the data is shown on the left side of Figure 4.4.

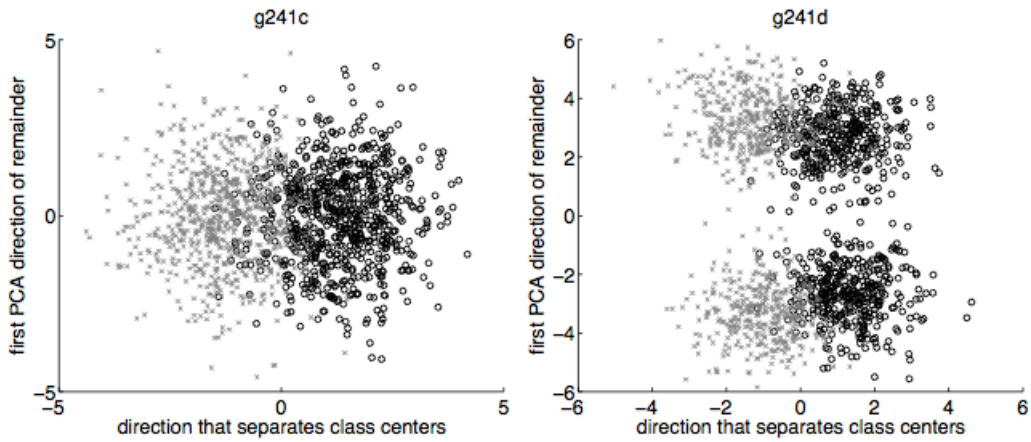


Figure 4.4: Two-dimensional projections of g241c (left) and g241d (right). Black circles, class +1; gray crosses, class -1.

g241d: This dataset was constructed to have a potentially misleading cluster structure, and no manifold structure. First, 375 points were drawn from each of two unit-variance isotropic Gaussians, the centers of which have a distance of 6 in a random direction. These points form the class +1. Next, the centers of two further Gaussians for class -1 were fixed by moving a distance of 2.5 in a random direction from each of the former centers. A two-dimensional projection of the resulting data is shown on the right side of Figure 4.4.

Digit1: This dataset was designed to consist of points which are close to a low-dimensional manifold embedded into a high-dimensional space, but not to show a pronounced cluster structure. To make the task a bit more difficult, a sequence of transformations is applied to the data [19]. As an example, the first data point is

shown in Figure 4.5 (left) and the result of this transformation is shown on the right side of Figure 4.5.

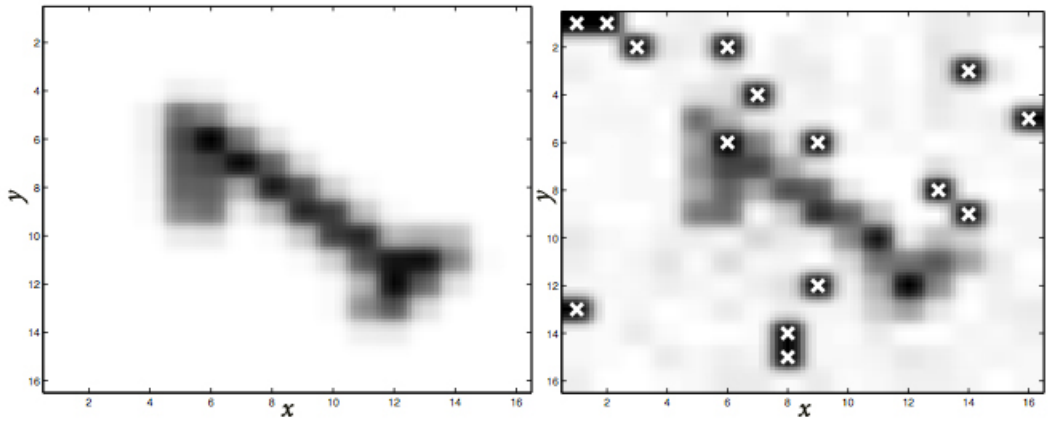


Figure 4.5: First data point from Digit1 dataset. Original image (left), and after rescaling, adding noise, and masking dimensions (x) (right).

USPS: We derived a benchmark dataset from the famous USPS set of handwritten digits as follows: We randomly drew 150 images of each of the ten digits. The digits '2' and '5' were assigned to the class +1, and all the others formed class -1. The classes were thus imbalanced with relative sizes of 1:4. We also expected both the cluster assumption and the manifold assumption to hold. To make the task more challenging, a sequence of transformations is applied to the data [19]. Figure 4.6 illustrates the impact.

BCI: This dataset originates from research on the development of a brain computer interface (BCI) [74]. A single person (subject C) performed 400 trials during each of which he imagined movements with either the left hand (class -1) or the right hand (class +1). In each trial, electroencephalography (EEG) was recorded from 39 electrodes. An autoregressive model of order 3 was fitted to each of the resulting 39

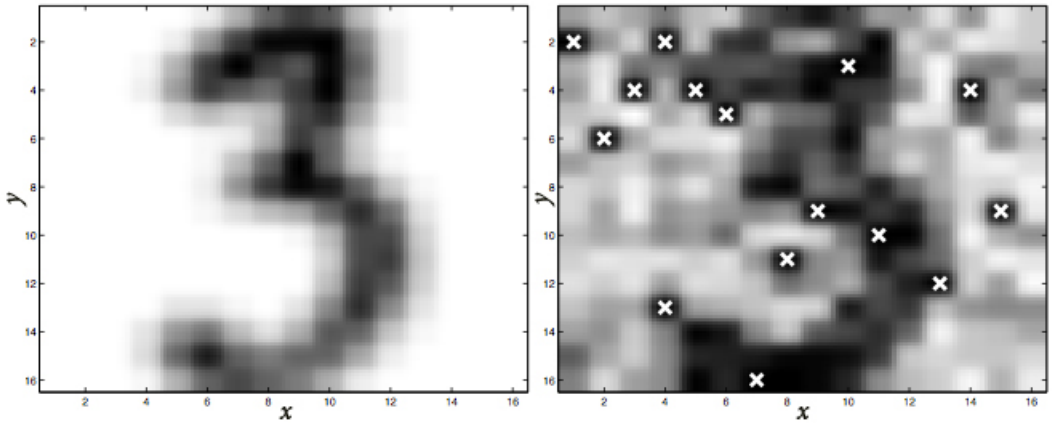


Figure 4.6: Fourth data point from the USPS dataset. Original image (left), and after rescaling, adding noise, and masking dimensions (x) (right).

time series. The trial was represented by the total of 117 fitted parameters.

SecStr: The main purpose of this benchmark dataset is to investigate the extent to which current methods can cope with large-scale application. The task is to predict the secondary structure of a given amino acid in a protein based on a sequence window centered around that amino acid. The dataset is based on the CB513 set, which was created by Cuff and Barton and consists of 513 proteins [30]. The 513 proteins consist of a total of 84,119 amino acids, 440 of which (X, Z, and B) were not considered.

Two-moon: 3D noisy two-moon data which contain 300 positive and 300 negative sample points, as well as an additional 200 noisy background points.

Breast Cancer: Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The numerical attributes of the datasets are all normalized to

span the range [0,1]. Complete details about each of these datasets are available [19, 126, 89].

4.2.2 Experimental Setup

Three different well-known classifiers were used: C4.5, Random Forest (RF) and SVM [53]. We used the Waikato Environment for Knowledge Analysis (WEKA) implementation for the three algorithms [53]. These different classification algorithms allowed us to compare our approach to other methods which are able to handle misclassification costs directly. We have adopted a five-fold cross validation technique to estimate the AUC measure. Each fold is divided into $k = 20$ equal sets of size $1/20$ for the relabeling step via the SSL algorithm. Each classifier has been applied to the original (imbalanced) training datasets and also to the preprocessed datasets using state-of-the-art US techniques. We integrated the three SSL algorithms described earlier into our framework to demonstrate the effectiveness of our approach.

We applied Friedman's ranking of the algorithms for each dataset independently, according to the AUC results, to evaluate how well our algorithms ranked compared to the other algorithms. Next, to support our findings, we used the non-parametric statistical test - Wilcoxon's paired signed rank test with a 95% confidence level - to asses the statistically significant differences between each pair of the US algorithms.

Prediction

In under-sampling, one should use a method whose assumptions best fit the problem's structure. Determining such a method, though, is a very challenging

task. In the US-SSL framework (Fig. 4.3), given SSL algorithms and some rules for algorithm selection, we can choose an appropriate way to under-sample a given dataset. We predict that the cluster-like datasets will have a well-defined training set, and therefore yield better classification performance if they are preprocessed using the cluster-based algorithm (TSVM) in the US-SSL framework. Similarly, for manifold-like datasets, we should obtain a favorable result when we use either LGC or GTAM as the base SSL algorithm in the US-SSL framework.

Therefore, as a general rule, manifold-based algorithms should be used for manifold-like datasets, and cluster-based algorithms should be used for cluster-like datasets. We will use this rule to choose the proper SSL method for our US-SSL framework. Hence, the datasets can be divided into two groups: manifold-like and cluster-like.

Table 4.3: First row: number of components kept in the dimensionality reduction; second row: estimate of the manifold dimension according to Hein and Audbert algorithm

Digit1	USPS	BCI	g241c	g241d
4	9	8	38	33
15	4	9	66	63

The way that was used to identify these two main categories is by performing dimensionality reduction to calculate the number of component kept and an intrinsic dimensionality estimation as described in [59]. The datasets that lie near a low-dimensional manifold are considered manifold-like (Table 2). This seems to be the case for datasets such as: Two-moon, Breast Cancer, Digit1, USPS, and BCI. For the first three, this can be easily explained by the fact that the data represent images.

A cluster assumption which states that classes are often separated by a low-density region, that is, if two data points are in the same cluster they are likely to share the same label. In other words, the data tend to cluster in such a way that two classes will not share the same cluster (also known as low-density separation assumption [19]). By construction, this is the case for datasets g241c and g241d which are considered the cluster-like datasets.

4.3 Discussion

Analysis for Group A:

Table 4.3 presents the average AUC values across all datasets obtained with the three classifiers using different sampling approaches. US-SSLRF significantly outperformed other US algorithms for C4.5 and Random Forest classifier, followed by RUS and US-YATSI. In the case of SVM, the US-SSLRF and US-YATSI algorithms ranked second and third after RUS, respectively LGC did not perform as well as the SSLRF and YATSI semi-supervised algorithms, due to its sensitivity to the initial labels and label class imbalance [142].

The Friedman average ranks for the three classifiers are depicted in Fig. 4.7. This serves as further confirmation of the findings with regard to the AUC. Among the US algorithms, US-SSLRF has the best overall ranking, followed by RUS and US-YATSI. Imbalanced datasets produced the highest average ranks (worst performance) with all classifiers.

Table 4.4: Average AUC, GM and AGF values for three different classifiers

Dataset	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.746	0.719	0.701	0.786	0.750	0.719	0.622	0.573	0.573
CPM	0.705	0.607	0.639	0.754	0.749	0.726	0.595	0.489	0.566
NearMiss-2	0.670	0.622	0.510	0.685	0.635	0.498	0.702	0.685	0.554
OSS	0.791	0.786	0.702	0.810	0.817	0.735	0.740	0.732	0.674
RUS	0.806	0.799	0.689	0.825	0.831	0.726	**0.763**	**0.753**	0.660
SBC	0.586	0.575	0.307	0.594	0.590	0.317	0.564	0.523	0.260
US-LGC	0.768	0.765	0.718	0.804	0.812	**0.764**	0.719	0.712	**0.679**
US-SSLRF	**0.812**	**0.803**	**0.732**	**0.834**	**0.836**	**0.764**	0.747	0.730	0.665
US-YATSI	0.797	0.794	0.721	0.822	0.826	0.761	0.754	0.747	0.669

For the non-parametric Wilcoxon signed-rank test, Table 4.4 reports the z-values and the p-values obtained, where the symbol "**" indicates that the null-hypothesis of equivalence with the control algorithm is not rejected at a significance level of $\alpha = 0.05$ (no significant difference between the two methods). The Wilcoxon test results reveal the higher performance of US-SSLRF and US-YATSI over all other US approaches with Random Forest. With C4.5, US-SSLRF performs significantly better than all other algorithms, except RUS, US-YATSI and OSS, which behave equally well. For SVM, the presented three SSL algorithms and NearMiss-2 outperform all other techniques.

As a general framework, US-SSL improves the classification performance for imbalanced datasets. Two of the three semi-supervised learners proposed outperformed all other algorithms (or performed equally well). No one algorithm performed the best in all given datasets. Generally, given an imbalanced dataset, it is difficult to determine which sampling technique should to be used. This proposed framework

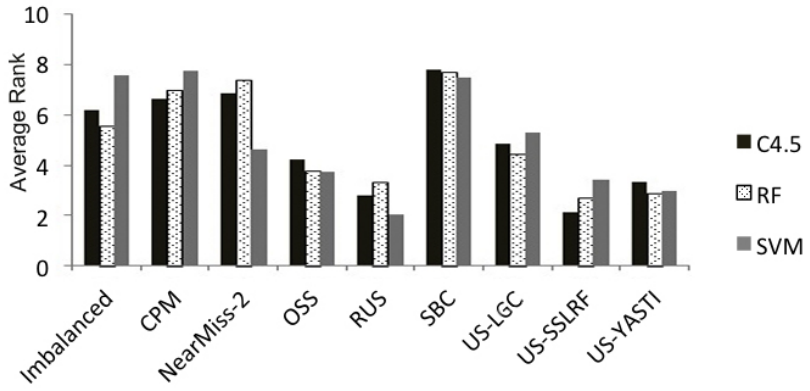


Figure 4.7: Friedman's average ranks for the three classifiers.

Algorithm	z	p value	Algorithm	z	p value	Algorithm	z	p value
C4.5 (US-SSLRF is the control method)			RF (US-SSLRF is the control method)			SVM (US-SSLRF is the control method)		
Imbalanced	-3.8490		Imbalanced	-6.3800		Imbalanced	-5.9580	
CPM	-6.7210		CPM	-6.1720		CPM	-5.4800	
NearMiss-2	-5.7610		NearMiss-2	-6.7020		RUS	-4.8200	
SBC	-6.9020		OSS	-5.1890		SBC	-5.1700	
US-LGC	-4.7780		SBC	-6.0060		OSS	-2.4510	.014
OSS**	-1.5530	.121	US-LGC	-6.7300		US-LGC**	-1.6990	.089
US-YATSI**	-1.5200	.129	RUS	-3.7350	.002	US-YATSI**	-1.6150	.105
RUS**	-0.1540	.881	US-YATSI**	-1.4270	.153	NearMiss-2**	-0.8590	.390

Table 4.5: US-SSL results obtained from the Wilcoxon signed-rank test.

has great potential to solve this problem. By knowing the underlying structure of the dataset, we can choose the most suitable semi-supervised learner based on which category of learner has already been established as performing well for the given structure.

Analysis for Group B: In our analysis, we will aim to answer three questions: (i) whether the US-SSL method will still outperform other US methods given three different SSL algorithm and more difficult US datasets; (ii) whether the predicted SSL algorithm produces the best results as compared to other SSL algorithms; and (iii)

whether the predicted SSL will produce the best/equivalent results in comparison to the state-of-the-art methods, and whether the imbalance ratio (imbalanced vs. highly imbalanced) will have any effect.

Tables 4.5 and 4.6 summarize the average AUC, GM and AGF values across all datasets obtained with the three classifiers using the three US-SSL versions and the most successful state-of-the-art US approaches. As expected, on average, classification with US-SSL methods yielded the best or comparable to the best performing algorithms. This, in turn, contributes to stronger classification results, according to the AUC metric. For the imbalanced datasets (IR = 18%), the performance of C4.5 and SVM are ranked highest when used with US-LGC as the sampling method. Random Forests has the better performance results with US-TSVM. The average of GM metrics across all datasets comparably followed the AUC metric behavior with Random Forest. US-TSVM exhibits the better results with SVM, and US-SSL methods closely follow RUS with C4.5. However, as per AGF, US-SSL methods exhibit the better results across all classifiers. For highly imbalanced datasets (IR = 09%), US-SSL methods perform significantly better than other algorithms, specifically US-TSVM where it ranked the highest in terms of GM and AGF values. However, as per AUC, US-SSL methods perform equally well to RUS for C4.5 and Random Forest classifiers.

The Friedman average ranks for the three classifiers are depicted in Figures 4.8 and 4.9. This serves as further confirmation of the findings with regard to the AUC. Figure 4.8 presents the average rank for the three classifiers for the imbalanced datasets (IR = 18%). For the C4.5 (IR = 18%), US-LGC clearly arises as the US

algorithm with the lowest ranking, that is, the highest performance on average. This is followed by US-TSVM and RUS. For the Random Forest classifier, US-TSVM is the technique with the best ranking, followed by RUS and US-LGC. Figure 4.9 presents the average rank for the three classifiers for the imbalanced datasets (IR = 09%). For C4.5 and SVM, US-TSVM and RUS ranked the best among all other algorithms. When using Random Forest, US-LGC ranks the best followed by RUS and US-GTAM.

For the non-parametric Wilcoxon signed-rank test, Tables 4.7 and 4.8 report the z-values and the p-values obtained, where the symbol "**" indicates that the null-hypothesis of equivalence with the control algorithm is not rejected at a significance level of $\alpha = 0.05$ (no significant difference between the two methods). The Wilcoxon test results for the first group of datasets (IR = 18%) reveal the higher performance of US-US-TSVM over all other US approaches with Random Forest. With C4.5, US-LGC performs significantly better than all other algorithms, except RUS, and US-TSVM, which behave equally well. For SVM, the US-LGC and US-TSVM algorithms outperform all other techniques.

For the highly imbalanced datasets (IR = 09%), US-SSL methods perform significantly better than CPM and OSS, but behave equally as well as RUS with C4.5 and Random Forest. However, there is no significant improvement for the SVM classifier.

The findings above concur with our earlier hypothesis that, on average, US-SSL algorithms produce significantly better or equivalent results as compared to the state-of-the-art US algorithms, even with more difficult datasets and different SSL algorithms [5]. However, for this part of the analysis, we will consider each dataset

individually to evaluate our prediction framework regarding which SSL method will result in a well-defined training set when used in a US-SSL under-sampling context.

In general, it appears that using a US-SSL method produces balanced datasets which accurately represent the underlying class distribution. This results in improvement in the classification performance for imbalanced datasets. However, no one algorithm performed the best in all given datasets. Generally, given an imbalanced dataset, it is difficult to determine which sampling technique should be used. This proposed framework has great potential to solve this problem. By knowing the underlying structure of the dataset, we can choose the most suitable semi-supervised learner based on which category of learners has already been established as performing well for the given structure. We will discuss the results in two parts: IR $< 20\%$ (imbalanced) and IR $< 10\%$ (highly imbalanced), respectively.

Table 4.6: Average AUC, GM and AGF values for the three different classifiers (IR = 18%).

Dataset	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.718	0.704	0.721	0.737	0.700	0.748	0.777	0.774	0.780
CPM	0.654	0.635	0.620	0.715	0.699	0.688	0.748	0.735	0.702
OSS	0.661	0.655	0.659	0.718	0.697	0.646	0.750	0.736	0.643
RUS	0.728	**0.728**	0.700	0.753	0.749	0.709	0.789	0.789	0.762
US-TSVM	0.728	0.726	**0.724**	**0.756**	**0.752**	**0.749**	0.797	**0.794**	**0.797**
US-LGC	**0.731**	0.723	0.723	0.743	0.739	0.742	**0.800**	0.785	0.795
US-GTAM	0.720	0.708	0.681	0.741	0.725	0.690	0.772	0.765	0.741

Imbalanced datasets (IR = 18%): Table 4.9 presents the detailed matrices of each dataset in the benchmark for three different classifiers. The summary below describes which algorithm performed best, given various classifiers, for each of the

Table 4.7: Average AUC, GM and AGF values for the three different classifiers (IR = 09%).

Dataset	C4.5			Random Forest			SVM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
Imbalanced	0.702	0.656	0.692	0.676	0.520	0.682	0.749	0.673	0.758
CPM	0.662	0.617	0.636	0.677	0.608	0.666	0.730	0.668	0.703
OSS	0.690	0.683	0.617	0.716	0.697	0.645	**0.768**	0.727	0.713
RUS	**0.722**	0.722	0.639	**0.739**	0.735	0.636	**0.768**	**0.767**	0.695
US-TSVM	**0.722**	**0.723**	**0.715**	0.719	**0.743**	**0.707**	0.767	**0.767**	**0.764**
US-LGC	0.715	0.704	0.690	**0.739**	0.729	0.686	0.757	0.757	0.736
US-GTAM	0.694	0.679	0.613	0.710	0.689	0.621	0.738	0.727	0.666

Table 4.8: US-SSL results obtained from the Wilcoxon signed-rank test (IR = 18%).

Algorithm	z	$p$value (10^3)	Algorithm	z	$p$value (10^3)	Algorithm	z	$p$value (10^3)
C4.5 (US-LGC is the control method)			RF (US-TSVM is the control method)			SVM (US-LGC is the control method)		
Imbalanced	-4.738	0	Imbalanced	-6.170	0	Imbalanced	-6.006	0
CPM	-3.486	0	CPM	-3.943	0	CPM	-3.257	1
OSS	-2.729	6	OSS	-3.000	2	OSS	-2.630	8
RUS**	-1.687	91	RUS	-2.071	38	RUS	-1.992	46
US-GTAM	-2.228	9	US-GTAM	-2.686	7	US-GTAM	-2.342	19
US-TSVM**	-1.338	180	US-LGC	-2.581	9	US-TSVM**	-1.500	133

Table 4.9: US-SSL results obtained from the Wilcoxon signed-rank test (IR = 09%).

Algorithm	z	$p$value (10^3)	Algorithm	z	$p$value (10^3)	Algorithm	z	$p$value (10^3)
C4.5 (US-TSVM is the control method)			RF (US-LGC is the control method)			SVM (US-TSVM is the control method)		
Imbalanced	-5.951	0	Imbalanced	-4.719	0	Imbalanced	-5.877	0
CPM	-4.285	0	CPM	-2.428	15	CPM	-2.554	10
OSS	-3.284	1	OSS	-2.113	34	OSS**	-0.844	400
RUS**	-1.000	317	RUS**	-1.857	62	RUS**	-0.900	368
US-GTAM	-2.671	7	US-GTAM**	-1.400	161	US-GTAM**	-1.314	190
US-LGC**	-1.137	254	US-TSVM**	-1.485	136	US-LGC**	-0.925	352

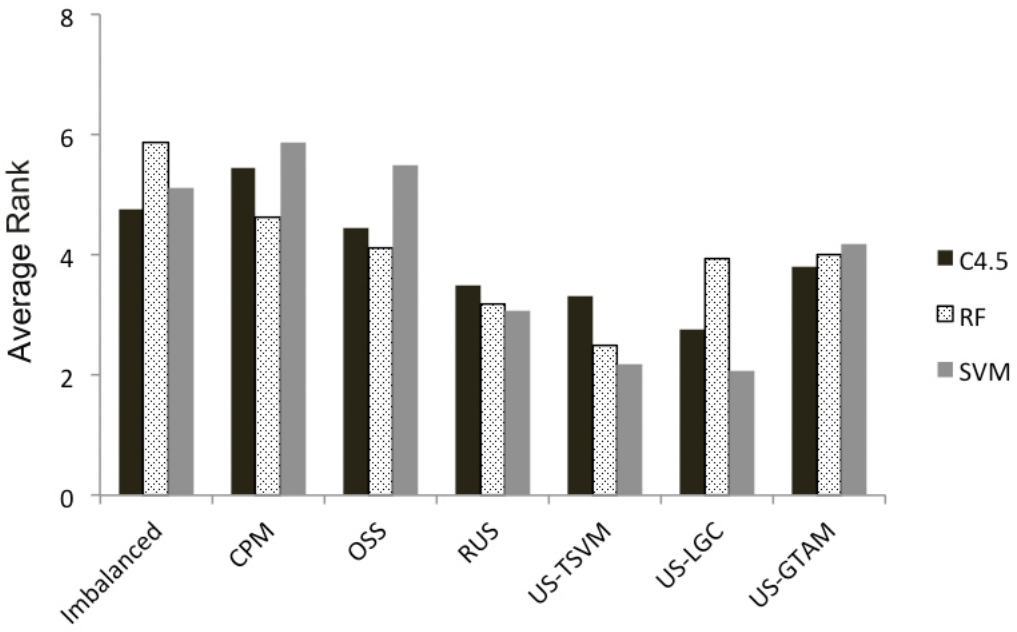


Figure 4.8: Friedman's average ranks for the three classifiers (IR = 18%).

seven datasets: BCI, Breast Cancer, Digits, Two-moon, SecStr, USPS, and G241c and G241d (see below).

BCI: With C4.5 and SVM classifiers, the US-LGC algorithm performed significantly better than the remaining algorithms. With Random Forest, the CPM and US-TSVM algorithms performed better than US-LGC. The highest-recorded performance was obtained using the SVM classifier and US-LGC as the US algorithm.

Breast Cancer: With C4.5 and Random Forest, US-LGC produced better results than all algorithms besides OSS, in terms of AUC values. However, as per AGF, US-LGC exhibited significantly better results. For SVM, US-LGC yielded the best performance in all metrics when compared with state-of-the-art algorithms.

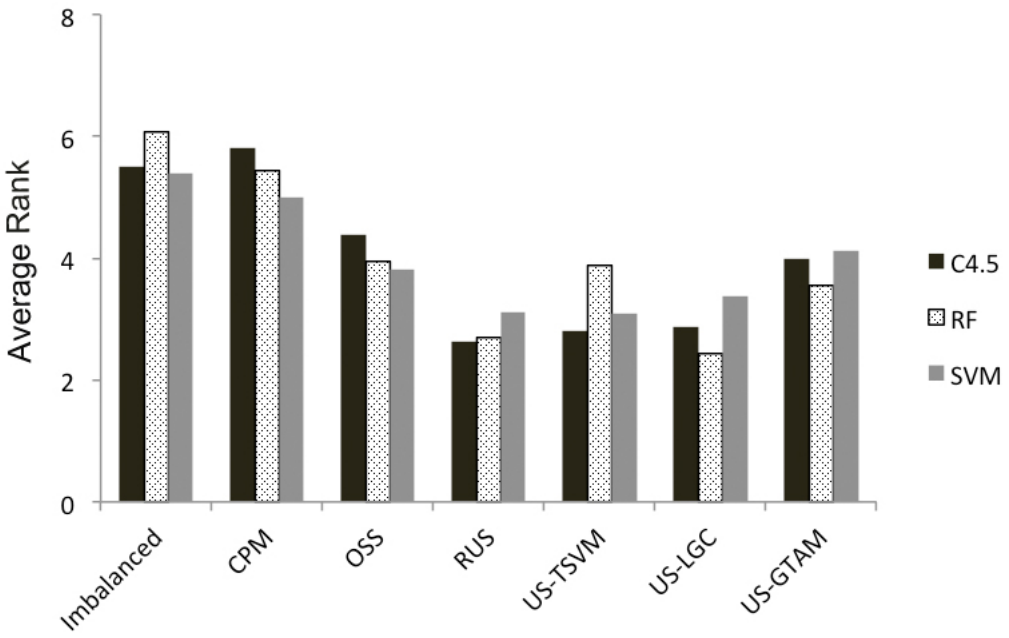


Figure 4.9: Friedman's average ranks for the three classifiers (IR = 09%).

Digits: US-LGC recorded the highest values among all algorithms for both C4.5 and Random Forest classifiers; yet, in terms of AGF, it came in second to other US-SSL algorithms (US-TSVM). With SVM, US-LGC produced better results than all other algorithms aside from RUS, in terms of AUC values.

Two-moon: The performance of C4.5 and Random Forest was ranked highest when used with US-GTAM. SVM performed better when combined with US-LGC. However, it did significantly worse than C4.5 and Random Forest.

SecStr: Across all classifiers, there was no difference in performance in terms of AUC, GM or AGT between US-TSVM, US-LGC, US-GTAM and RUS algorithms.

USPS: Regardless of classifier, the US-TSVM algorithm yielded better performance as compared to all other sampling algorithms.

G241c and G241d: Both of these are cluster-like datasets and thus, as expected, the very best results were obtained when using cluster-based algorithms such as TSVM in US-TSVM for US, with the SVM classifier. G241c ranked first with C4.5 and third with Random Forest. G241d ranked first with C4.5 and Random Forest in terms of GM values and first and second in terms of AUC values, respectively.

Highly imbalanced datasets (IR = 09%): Table 4.10 presents the detailed matrices of each dataset in the benchmark for the three different classifiers. The summary below describes which algorithm performed the best, given various classifiers, for each of the seven datasets: BCI, Breast Cancer, Digits, Two-moon, SecStr, USPS, and G241c and G241d.

BCI: With the C4.5 classifier, US-LGC ranked first in terms of AGF values, along with US-TSVM and CPM. It ranked second in terms of AUC and GM values, behind the OSS algorithm. In the case of Random Forest, US-LGC performed better than all other algorithms in terms of GM and AGF metrics, and it ranked second after CPM in terms of AUC values. With the SVM classifier, US-GTAM ranks second after OSS algorithm.

Breast Cancer: With C4.5, US-LGC and US-TSVM performed as well as the RUS algorithm in terms of AUC and GM metrics. However, they significantly outperformed RUS in terms of AGF values. In the case of SVM, US-GTAM outperformed all other algorithms.

Table 4.10: AUC, GM, and AGF results for 3 different classifiers (IR = 18%).

Results for the C4.5 Classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.493	0.461	0.490	0.549	0.478	0.535	0.563	0.549	0.515	0.565	0.565	0.553	0.553	0.553	0.530	**0.570**	**0.570**	**0.555**	0.535	0.494	0.478
Breast	0.900	0.908	0.895	0.891	0.890	0.875	**0.915**	**0.914**	0.886	0.893	0.893	0.873	0.894	0.894	**0.915**	**0.914**	**0.914**	0.911	0.905	0.904	0.899
Digits	0.897	**0.906**	0.893	0.689	0.689	0.644	0.689	0.689	0.799	0.866	0.865	0.834	0.851	0.851	**0.903**	**0.901**	0.864	0.897	0.836	0.835	0.792
G241c	0.557	0.529	0.555	0.569	0.559	0.556	0.569	0.559	0.568	0.584	0.584	0.548	**0.593**	**0.592**	**0.576**	0.577	0.576	0.555	0.585	0.585	0.548
G241d	0.571	0.538	0.582	**0.589**	**0.576**	0.577	**0.589**	**0.576**	0.516	0.564	0.563	0.528	0.573	**0.576**	0.555	0.583	0.562	**0.584**	0.585	0.538	0.558
Two-moon	0.961	0.971	0.971	0.679	0.678	0.604	0.679	0.678	0.850	0.977	0.977	0.974	0.952	0.951	0.931	0.923	0.917	0.898	**0.985**	**0.985**	**0.981**
SecStr	0.583	0.540	0.588	0.593	0.540	0.556	0.605	0.602	0.556	**0.613**	**0.613**	0.579	0.611	0.610	**0.598**	0.611	0.611	**0.598**	0.593	0.610	**0.598**
USPS	0.781	0.780	0.791	0.672	0.671	0.613	0.683	0.675	0.581	0.766	0.766	0.709	**0.798**	**0.782**	**0.786**	0.770	0.770	0.784	0.734	0.712	0.594

Results for the Random Forest Classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.570	0.541	**0.583**	**0.573**	**0.568**	0.567	0.565	0.493	0.464	0.550	0.536	0.506	**0.573**	0.567	0.563	0.547	0.532	0.535	0.532	0.481	0.453
Breast	0.903	0.921	0.936	0.934	0.934	0.925	**0.942**	0.942	0.929	0.927	0.927	0.911	0.935	**0.943**	0.941	0.939	0.938	**0.947**	0.927	0.926	0.910
Digits	0.908	0.905	0.931	0.806	0.805	0.773	0.806	0.805	0.633	0.905	0.905	0.876	0.918	0.911	0.931	**0.929**	**0.914**	**0.943**	0.898	0.871	0.858
G241c	0.580	0.482	0.569	0.622	0.609	**0.613**	0.622	0.609	0.553	**0.636**	**0.624**	0.569	0.598	0.588	0.565	0.575	0.560	0.576	0.614	0.601	0.545
G241d	0.552	0.433	0.541	0.587	0.545	**0.588**	0.587	0.545	0.510	0.575	0.565	0.509	**0.609**	**0.598**	0.581	0.568	**0.603**	0.567	0.587	0.578	0.521
Two-moon	0.977	0.977	0.978	0.842	0.841	0.818	0.842	0.841	0.912	0.981	0.980	0.974	0.961	0.952	0.947	0.934	0.923	0.909	**0.985**	**0.985**	**0.981**
SecStr	0.589	0.535	0.595	0.599	0.535	0.531	0.610	0.588	0.531	**0.617**	**0.613**	0.565	0.616	0.611	0.605	0.610	0.606	**0.610**	0.616	0.611	**0.610**
USPS	0.813	0.808	0.852	0.756	0.756	0.691	0.771	0.751	0.636	0.838	0.837	0.765	**0.844**	**0.843**	**0.862**	0.841	0.838	0.851	0.768	0.747	0.643

Results for the SVM classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.642	0.593	0.662	0.643	0.620	0.646	0.620	0.583	0.549	0.607	0.607	0.588	0.650	**0.642**	0.648	**0.669**	0.637	**0.668**	0.623	0.611	0.577
Breast	0.915	0.922	0.949	0.918	0.915	0.942	0.950	0.949	0.943	0.939	0.939	0.936	0.930	0.939	**0.963**	**0.965**	**0.965**	0.954	0.962	0.950	0.961
Digits	0.922	0.931	0.927	0.903	0.902	0.895	0.903	0.902	0.922	**0.949**	**0.949**	0.928	0.937	0.937	**0.947**	0.940	0.940	0.944	0.889	0.876	0.835
G241c	0.810	0.807	0.808	0.791	0.790	0.773	0.791	0.790	0.772	0.806	0.806	0.774	**0.823**	**0.816**	**0.816**	0.818	0.773	0.814	0.799	0.770	0.791
G241d	0.771	0.775	0.779	0.762	0.755	0.756	0.762	0.755	0.742	0.783	0.783	0.745	**0.801**	0.800	**0.801**	0.783	0.760	0.781	0.775	**0.804**	0.763
Two-moon	0.818	0.823	0.834	0.645	0.575	0.608	0.645	0.575	0.245	0.825	**0.825**	0.804	0.837	0.820	0.834	**0.838**	0.818	**0.842**	0.825	**0.825**	0.830
SecStr	0.500	0.500	0.421	0.500	0.500	0.215	0.500	0.500	0.215	**0.530**	0.530	**0.506**	**0.530**	**0.530**	**0.506**	0.530	0.530	**0.506**	**0.530**	**0.530**	**0.506**
USPS	0.836	0.838	0.858	0.821	0.821	0.778	0.834	0.833	0.759	**0.869**	**0.869**	0.820	**0.869**	**0.869**	**0.860**	0.855	0.855	0.851	0.770	0.756	0.662

Digits: With the Random Forest and SVM classifiers, US-LGC performed better than all other algorithms. In the case of C4.5, US-LGC recorded the highest AGF values, but fell behind OSS in terms of AUC and GM values.

Two-moon: US-LGC fell behind both RUS and OSS in terms of AUC and GM values with both the C4.5 and Random Forest classifiers. However, US-LGC recorded higher AGF values than both RUS and OSS. In terms of the SVM classifier, US-LGC outperformed all other algorithms in all metrics.

SecStr: With C4.5 and Random Forest, there were no significant differences between the US-TSVM, US-LGC, US-GTAM, CPM and RUS algorithms.

USPS: The US-TSVM algorithm resulted in the best performance among all other sampling algorithms across all classifiers.

G241c and G241d: For G241c, US-TSVM outperformed all other algorithms with the C4.5 and Random Forest classifiers. With SVM, US-TSVM ranked second after RUS and OSS in terms of AUC, yet it still maintained higher AGF than both RUS and OSS. G241d outperformed all other algorithms with SVM and Random Forest classifiers. However, with C4.5, it did fall behind RUS in terms of AUC and GM values, despite recording higher AGF values.

Summary and findings:

To compare the different methods and evaluate our prediction framework, the results are summarized in Tables 4.11 and 4.12. Some of the main findings of the US-SSl framework include:

Table 4.11: AUC, GM, and AGF results for 3 different classifiers (IR = 09%)

Results for the C4.5 Classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.540	0.490	0.533	0.555	0.511	**0.549**	**0.575**	**0.575**	0.522	0.518	0.517	0.458	0.572	0.571	**0.549**	0.572	0.571	**0.549**	0.541	0.525	0.491
Breast	0.912	0.920	0.885	0.913	0.912	0.868	0.855	0.854	0.740	**0.940**	**0.939**	0.839	0.922	0.923	**0.925**	0.922	0.917	**0.925**	0.921	0.917	0.849
Digits	0.801	0.796	0.749	0.772	0.762	0.739	**0.835**	**0.835**	0.744	0.807	0.807	0.719	0.830	0.829	**0.829**	0.833	0.832	0.798	0.792	0.759	0.714
G241c	0.598	0.547	0.592	0.564	0.535	0.533	0.521	0.500	0.486	0.576	0.576	0.490	**0.608**	**0.602**	**0.603**	0.572	0.548	0.566	0.544	0.543	0.519
G241d	0.549	0.466	0.547	0.529	0.488	0.506	0.540	0.526	0.496	**0.606**	**0.606**	0.508	0.564	0.563	0.547	0.560	0.538	**0.558**	0.575	0.558	0.487
Two-moon	0.937	0.946	0.923	0.729	0.686	0.721	0.956	0.955	0.894	**0.971**	**0.971**	0.935	0.946	0.898	**0.953**	0.949	0.949	0.944	0.895	0.893	0.815
SecStr	0.540	0.367	**0.558**	0.550	0.367	0.569	0.610	0.592	0.569	**0.611**	0.610	0.516	0.550	**0.611**	**0.558**	**0.611**	**0.611**	**0.558**	**0.610**	**0.592**	**0.558**
USPS	0.736	0.715	0.748	0.684	0.676	0.604	0.628	0.625	0.490	0.749	0.747	0.648	**0.782**	**0.782**	**0.759**	0.704	0.664	0.627	0.669	0.630	0.475

Results for the Random Forest Classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.497	0.268	0.473	**0.586**	0.542	0.579	0.504	0.497	0.474	0.518	0.518	0.458	0.497	0.545	0.473	0.581	**0.552**	**0.590**	0.550	0.506	0.481
Breast	0.936	0.945	0.947	0.894	0.891	0.882	0.940	0.940	0.878	**0.957**	**0.956**	0.884	0.936	0.934	**0.954**	0.941	0.940	0.933	0.949	0.937	0.885
Digits	0.772	0.752	0.839	0.703	0.687	0.671	0.783	0.770	0.613	0.838	0.837	0.720	0.782	**0.880**	0.839	**0.881**	0.878	**0.856**	0.850	0.829	0.798
G241c	0.503	0.128	0.476	0.577	0.487	**0.579**	0.557	0.498	0.542	0.573	0.565	0.447	**0.606**	**0.608**	0.578	0.579	0.573	0.485	0.584	0.578	0.540
G241d	0.500	0.091	0.468	0.533	0.379	0.536	0.574	0.528	**0.553**	0.585	0.572	0.445	**0.607**	**0.600**	0.543	0.595	0.576	0.482	0.565	0.555	0.437
Two-moon	0.944	0.953	**0.948**	0.859	0.859	0.811	0.961	0.961	0.910	**0.965**	**0.964**	0.934	0.941	0.930	0.939	0.959	0.959	**0.948**	0.895	0.893	0.850
SecStr	0.530	0.326	0.544	0.540	0.326	**0.571**	0.609	0.587	**0.571**	0.609	0.604	0.489	0.540	**0.606**	0.544	**0.610**	**0.606**	0.542	**0.609**	**0.587**	0.542
USPS	0.728	0.694	0.761	0.722	0.697	0.698	0.799	0.792	0.619	**0.866**	**0.864**	0.710	0.843	0.841	**0.791**	0.770	0.750	0.648	0.677	0.606	0.434

Results for the SVM classifier

Dataset	Imbalanced			CPM			OSS			RUS			US-TSVM			US-LGC			US-GTAM		
	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF	AUC	GM	AGF
BCI	0.580	0.399	0.588	0.649	0.616	0.646	**0.663**	**0.643**	**0.649**	0.584	0.583	0.532	0.594	0.593	0.588	0.587	0.586	0.585	0.662	0.627	0.627
Breast	0.941	0.939	0.964	0.951	0.949	**0.970**	0.966	0.966	0.966	0.942	0.941	0.938	0.942	0.942	0.964	0.952	0.952	0.958	**0.967**	**0.967**	0.965
Digits	0.911	0.905	0.921	0.873	0.871	0.845	0.899	0.899	0.844	0.904	0.904	0.841	0.911	0.902	**0.928**	**0.914**	**0.908**	0.910	0.852	0.848	0.765
G241c	0.707	0.668	0.708	0.696	0.670	0.677	**0.739**	0.726	0.705	**0.739**	**0.738**	0.652	0.734	0.733	**0.715**	0.705	0.703	0.714	0.724	0.751	0.697
G241d	0.725	0.689	0.730	0.734	0.716	0.712	0.713	0.701	0.669	0.733	0.732	0.639	**0.738**	0.738	**0.743**	0.727	**0.743**	0.740	0.663	0.662	0.580
Two-moon	0.796	0.771	0.825	0.592	0.483	0.538	0.815	0.812	0.687	0.826	0.826	0.732	0.830	0.830	**0.845**	**0.831**	**0.831**	**0.845**	0.806	0.804	0.775
SecStr	0.517	0.215	**0.495**	0.517	0.215	0.462	0.500	0.215	0.462	**0.530**	0.529	0.457	0.517	**0.530**	**0.495**	**0.530**	**0.530**	0.462	**0.530**	**0.530**	0.462
USPS	0.816	0.801	**0.833**	0.826	0.822	0.772	0.853	0.852	0.723	**0.886**	**0.885**	0.770	**0.866**	0.866	**0.833**	0.811	0.802	0.672	0.701	0.630	0.454

Table 4.12: Summary of how well the selected US-SSL method performs in comparison to the actual reported results for the imbalanced data (IR $< 20\%$). This result is based on the highest reported performance across all three classifiers: C4.5, Random Forests and SVM.

Dataset	Pre-experiment selection	Ranking for selected algorithm AUC/GM/AGF	Actual best algorithm AUC/GM/AGF
BCI	US-LGC/US-GTAM	1/2/1	US-LGC/US-TSVM/US-LGC
Breast	US-LGC/US-GTAM	1/1/2	US-LGC/US-LGC/US-TSVM
Digits	US-LGC/US-GTAM	2/2/2	RUS/RUS/US-TSVM
G241c	US-TSVM	1/1/1	US-TSVM/US-TSVM/US-TSVM
G241n	US-TSVM	1/2/1	US-TSVM/US-GTAM/US-TSVM
Two-moon	US-LGC/US-GTAM	1/1/1	US-GTAM/US-GTAM/US-GTAM
SecStr	US-LGC/US-GTAM	1/1/1	Tied with all US-SSL alorithms and RUS
USPS	US-LGC/US-GTAM	2/2/2	US-TSVM, RUS/US-TSVM, RUS/US-TSVM

Table 4.13: Summary of how well the selected US-SSL method performs in comparison to the actual reported results for highly imbalanced data (IR $< 10\%$). This result is based on the highest reported performance across all three classifiers: C4.5, Random Forests and SVM.

Dataset	Pre-experiment selection	Ranking for selected algorithm AUC/GM/AGF	Actual best algorithm AUC/GM/AGF
BCI	US-LGC/US-GTAM	2/2/2	OSS/OSS/OSS
Breast	US-LGC/US-GTAM	1/1/2	US-GTAM/US-GTAM/CPM
Digits	US-LGC/US-GTAM	1/1/2	US-LGC/US-LGC/US-TSVM
G241c	US-TSVM	2/2/1	RUS, OSS/RUS/US-TSVM
G241n	US-TSVM	1/2/1	US-TSVM/US-LGC/US-TSVM
Two-moon	US-LGC/US-GTAM	3/3/1	RUS/RUS/US-LGC
SecStr	US-LGC/US-GTAM	1/1/1	Tied with all US-SSL alorithms and RUS
USPS	US-LGC/US-GTAM	4/4/2	US-TSVM/US-TSVM/US-TSVM

- With imbalanced cluster-like datasets, the US-SSL framework - which uses a cluster-based algorithm - outperforms all other state-of-the-art US algorithms. This means that US-SSL results in a better and more well-defined and better training dataset than any other algorithm for both imbalanced and highly imbalanced datasets.

- For highly imbalanced datasets, US via the US-SSL algorithms ranks second in AUC and GM, but has higher AGF in comparison to any of the rest of the US algorithms, which tend to drop dramatically as IR increases. What this means is that if the predicted US-SSL algorithm ranked second behind any of the state-of-the-art algorithms, it still maintained the highest AGF values.

- In the literature [125], as a semi-supervised problem, the GTAM algorithm has reportedly performed better than LGC when applied to the balanced two-moon dataset. As expected, the US-GTAM recorded better performance than US-LGC with two of the three classifiers: C4.5 and Random Forest. However, with the highly imbalanced Two-Moon dataset, US-LGC took back the lead. In both cases, a manifold-based algorithm outperformed all other algorithms.

- With USPS datasets, US-TSVM outperforms all other sampling algorithms, regardless of the classifier used. This is true despite expecting a manifold-based algorithm to outperform a cluster-based algorithm for this dataset structure.

- It is worth pointing out that one should not necessarily expect a significant improvement with some datasets. The dataset SecStr, for example, demonstrates a situation in which it is difficult to perform better than other US algorithms.

At least for SecStr, this might result from the amounts of unlabeled data that are utilized.

- Finally, one should note that, aside from being imbalanced, each dataset in the second round of datasets was altered to make the learner's task even more difficult and to obscure the origin of the data. This was done to illustrate this algorithm's ability to perform well on complex and unfamiliar datasets. This was also done to increase the comparability of the results.

In all cases, we believe that there is no 'black box' solution. To successfully perform US, a good understanding of the nature of the data is crucial. Indeed, in supervised learning, it seems that a good generic learning algorithm can perform well on many real-world datasets without specific domain knowledge when the dataset is balanced. In contrast, the task is harder when dealing with imbalanced datasets and knowledge about data distribution - that correlates the label of a data point with its situation within the distribution - is essential to balance the dataset; therefore, it seems much more difficult to design a general US method. Instead, with the US-SSL framework and the powerful semi-supervised learning algorithms which distinguish themselves through the ability to make use of available prior knowledge about the domain and data distribution, a better, well-defined dataset can be generated and will yeild classification improvements.

There are many semi-supervised algorithms and more continue to be developed each year. For current as well as novel algorithms, using this model will result in considerable improvement in classifier performance through attaining better matching

between problem structure and model assumptions.

Chapter 5

CardioRS

5.1 Framework for Predicting Cardiovascular Events

5.1.1 Overview

The MESA data, as shown in Table 2.5, are imbalanced when the number of patients with events is compared to the number without events. If used directly to train a classifier, the resulting decision boundary will be severely biased, which could result in poor performance. To address this challenge, we developed a framework whereby the imbalanced data is processed via a sampling algorithm and a Cost-Sensitive Learning approach-CSL [40] is used. Mis-classification costs are used to select the best training distribution. We chose these two algorithms based to their ability to effectively increase the sensitivity of a classifier to the minority class as well as their unique ability to avoid over-fitting, which can be inevitable when using other

oversampling techniques.

We built 14 learning-based models, one per gender for each of the three event types. These models are based on an approach called cost-sensitive learning, whose foundation was originally presented by Charles Elkan [40]. The learner, or classifier, that we are using is a decision tree based algorithm (DT-based algorithm). Decision tree based algorithms are one of the most powerful learning algorithms and have been developed for many types of research over the past two decades [70]. Due to the nature of the application and the inherent characteristics of MESA data, we have chosen to use a cost sensitive decision tree learner [121]. We have elected to use a decision- tree-based learner because, while it uses computational and mathematical techniques, it is still descriptive and easy to convert to rules which physicians and non-computer science researchers can work with and verify.

In many real-world applications, especially in the medical community, the costs of misclassifications are often disparate. For example, the cost of incorrectly diagnosing a patient to be healthy when in fact he or she is unhealthy may be potentially much more devastating than diagnosing a healthy person as being ill. The MESA data are severely imbalanced in terms of outcomes and, as a result, the decision boundary will be severely biased. Hence, to overcome this obstacle, we approach the problem using a two stage method. First, we over-sample the data prior to the training stage using NEATER. Next, we apply a CSL approach to our DL classifier. The models are then tested on the remainder of the MESA data, which have not been seen by the classifiers.

To insure the accuracy and capability of the algorithm, the data are divided into

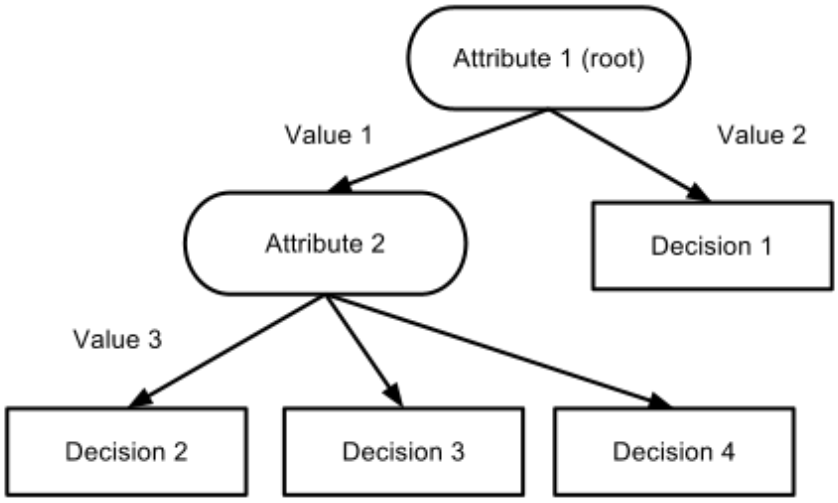
two groups: training data and testing data. The developed models are built using the training data, while their performance and effectiveness are evaluated, not by their performance on the training data itself, but rather by their ability to perform well on the testing data which the classifier has not seen before. Furthermore, to insure and increase the learning models' robustness and ability to be generalized, we use a 3-fold cross validation approach: two-thirds of the data are used for training, and the evaluation of the model is measured by how well it does on the unseen testing dataset.

5.1.2 Decision Tree Classifier (J48)

As noted above, decision trees (DT) are one of the most powerful learning algorithms and have been used in many applications for more than two decades. They are commonly applied for solving classification problems. The classification process consists of assigning a class to a sample using a model created based on several attributes of the sample. The data is represented in the form of a DT to precisely predict the values chosen by a decision from a set of predictive attributes. Thus, each dataset consists of a list of predictive attributes and decisions to be predicted. A DT is composed of leaves, nodes, and branches. Each node of the tree corresponds to a classified object property, called an attribute. Each branch of the tree corresponds to a possible value of the father attribute and each leaf of the tree corresponds to a class (Fig 5.1).

There are several DT algorithms, including the J48 algorithm. The J48 algorithm

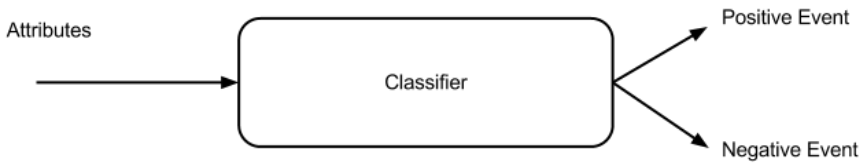
Figure 5.1: Depiction of a decision tree classifier.



is based on implementation of the C4.5 decision tree learning algorithm proposed by Quinlan [102]. In a given dataset, the algorithm identifies the attributes that discriminate between various instances most clearly (highest normalized information gain). At each step, the algorithm uses the most predictive attribute and splits a node based on this attribute. J48 attempts to account for noise and missing data and handles numeric attributes by determining where to place thresholds for decision splits. The main parameters that can be set for this algorithm are the confidence threshold, the minimum number of instances per leaf, and the number of folds for reduced error pruning. J48 is a popular DT learning algorithm used in a multitude of domains. In our experiments, we used the Waikato Environment for Knowledge Analysis (WEKA) [132] Open Source Software implementation of J48. Decision trees classify instances by starting at the root node, testing the attribute specified at this node, and then moving down the tree branch according to the attribute value given. Every path of the resulting tree, which spans from the root to a leaf, can be translated

into a rule according to the following statement: If Conditions Then Conclusion or If Attribute 1 = Value 1 And Attribute 2 = Value 2 Then Class 1 (Fig 5.2).

Figure 5.2: Depiction of attributes and decisions.



5.1.3 CardioRS Framework

In decision trees, cost-sensitive fitting can be accomplished in three ways: (i) cost-sensitive adjustments can be applied to the decision threshold; (ii) cost-sensitive considerations can be given to the split criteria at each node; or (iii) cost-sensitive pruning schemes can be applied to the tree. We used the first approach to determine the threshold. However, instead of relying on the training distribution or exact mis-classification costs, the technique we employed uses the receiver operating characteristic (ROC) evaluation procedure [41, 42]. ROC plots the range of performance values as the decision threshold is moved from the point where the total mis-classifications on the positive class are maximally costly to the point where total mis-classifications on the negative class are maximally costly. The decision threshold that yields the most dominant point on the ROC curve is then used as the final decision threshold [111].

Figure 5.3 depicts the overall process of CardioRS, beginning from the initial

dataset, to pre-processing the data using NEATER, then building our cost sensitive classifier based on J48, and finally, extracting the decision rules. For example, the following decision rules were extracted from the CardioRS-M model for male subjects:

Rule 1: If CACS >145.6 AND HT =Yes AND SBP <113 THEN Negative

Rule 2: If CACS >145.6 AND HT =Yes AND SBP >113 AND HDL >58.1 AND AGE <81 AND CACS >175.4 THEN Positive

Rule 3: If CACS = 0 AND CHOL <240 AND SBP <120.8 THEN Negative

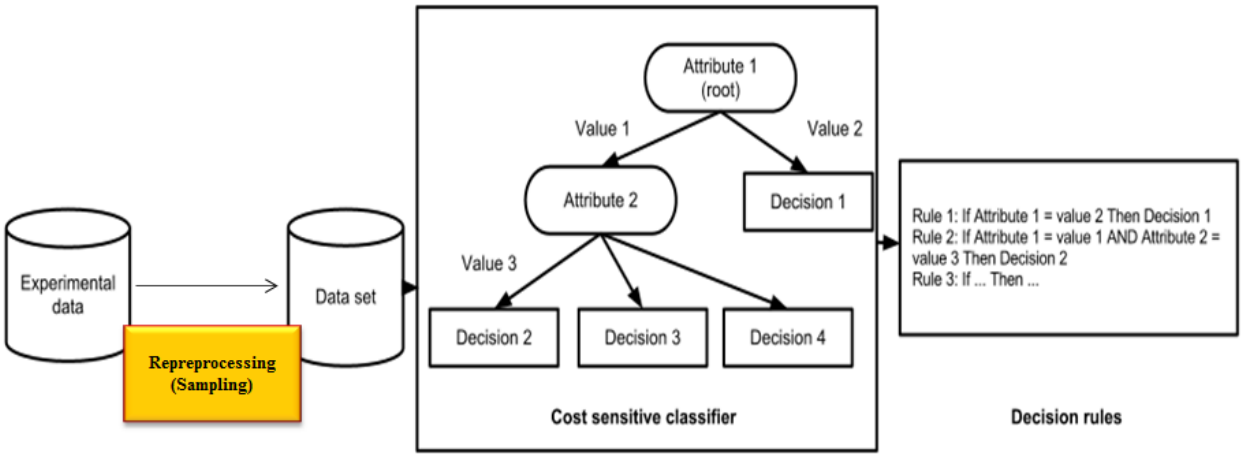
Some of the rules extracted from CardioX-F for female subjects are:

Rule 1: If AGE <57 AND SBP <107.5 THEN Negative

Rule 2: If AGE >57 AND LDL <222.7 AND SBP <125 AND LDL <157 AND HT = NO THEN Negative

Rule 3: If AGE >57 AND CHOL >222.7 AND SBP <169 AND HT = NO THEN Positive

Figure 5.3: Depiction of the CardioRS framework.



5.2 Experimental Design and Results

5.2.1 Experiments Setup

In the first set of experiments, we performed an analysis to determine the sensitivity and specificity of the FRS methodology, and whether adding CACS to the FRS risk factors improved the classification rate for any of the events and for each gender separately. We used this analysis to compare the performance of our learning-based approach with the statistical approach, and determine whether it confirmed other previous findings as presented in the scientific literature.

5.2.2 Statistical Analysis

Ten-year-estimated risk was computed for each participant using a Cox proportional-hazards model. All risk-prediction equations derived from the Framingham Heart

Study. $B^2_{m,w}$ is the base model where calcium is added as an additional risk factor. We used the standard Framingham risk factors as presented in Table 2.6 and then grouped the scores into the following two preselected 10-year risk prediction categories: low (<10%) and high (10% ≥). Table 5.1 presents the event rates for CVDA, CHDA and CHDH, based on FRS for all participants and grouped based on gender.

We then stratified the rates by the two levels of FRS and the following preselected four categories of CACS: 0, 1 to 100, 101 to 300, and 301 and more. Table 5.2 depicts the likelihood of cardiac events according to increasing levels of CACS and FRS. The tables illustrate the distribution of CVDA, CHDA and CHDH events across the eight categories defined by CACS X FRS.

Lastly, we recomputed the ten-year-estimated risk for each participant after adding calcium to the existing risk factors. However, to obtain a fair comparison among all approaches, we also adjusted the risk ranges into two categories, as we had done previously. Furthermore, we reported the performances of all the approaches on the same testing sets, since the performance of our proposed model is also being evaluated based on these same unseen testing sets.

Table 5.3 illustrates the sensitivity (Sn) and specificity (Sp) of the traditional FRS where only the traditional risk scores have been considered, and illustrates the evaluation when adding CACS to these factors (refer to Table 2.6 for each baseline risk factor). It is essential to remember that we are applying a 3-fold cross validation approach. By partitioning the available data into three sets, we drastically reduce the number of samples which can be used for training the model. The results are obtained from evaluating these models on the testing set, which is one-third of the

Table 5.1: CHDH, CHDA and CVDA events classified as "low risk" (<10%) and "high risk" (10% ≥) and the standard FRS of patients

		(FRS <10%)		(FRS ≥ 10%)	
Model		No. of Events	Size of Cohort	No. of Events	Size of Cohort
$B^1_{m,w}$	MEN	104	2,832	17	159
	WOMEN	58	3,289	11	144
$B^2_{m,w}$	MEN	82	2,623	39	368
	WOMEN	49	3,243	20	190
$B^3_{m,w}$	MEN	120	2,976	1	15
	WOMEN	67	3,429	2	4
$B^4_{m,w}$	MEN	185	2,836	24	155
	WOMEN	102	3,418	4	15
$B^5_{m,w}$	MEN	3	236	268	2,755
	WOMEN	15	1,216	154	2,217
$B^6_{m,w}$	MEN	4	358	267	2,633
	WOMEN	23	1,527	146	1,906
$B^7_{m,w}$	MEN	191	2,592	80	399
	WOMEN	138	3,243	31	190

Table 5.2: CHDH-MEN, CHDH-WOMEN events by Coronary Artery Calcium (CAC) score in men and women in the two risk categories: "Low Risk" (<10%) and "High Risk" (10% $\geq$) and the Standard FRS of Patients

		(FRS <10%)		(FRS $\geq$ 10%)	
Events	CAC Score	#Events	Size of Cohort	#Events	Size of Cohort
CHDH-MEN	CACS = 0	6	648	9	579
	0 CACS 100	5	298	29	546
	100 CACS 300	3	88	24	320
	300 CACS	2	75	43	437
CHDH-WOMEN	CACS = 0	12	1780	4	346
	0 CACS 100	9	593	9	198
	100 CACS 300	10	197	4	84
	300 CACS	10	155	11	80

entire MESA dataset—unlike the COX models, which are built using the whole dataset.

Table 5.3: Sensitivity and specificity under FRS for each event type: CVDA, CHDA and CHDH

Events	Model	Sn(%)	Sp(%)	Model	Sn(%)	Sp(%)
CHDH	B_m^1	14.05	85.05	B_w^1	15.94	96.04
	B_m^2	28.23	88.53	B_w^2	28.98	94.94
	B_m^3	1.26	99.51	B_w^3	2.89	99.94
CHDA	B_m^4	11.48	90.29	B_w^4	3.77	99.66
CVDA	B_m^5	98.80	8.56	B_w^5	87.09	36.70
	B_m^6	98.52	13.01	B_w^6	76.36	46.07
	B_m^7	29.52	88.72	B_w^7	18.34	95.12

5.2.3 CardioRS Analysis

As with the statistical approach presented above, in the second set of experiments using our learning-based approach, we also built the models with and without CACS. Initially, as expected, most of the baseline characteristics, including traditional cardiovascular risk factors, were significant. CAC was added to the risk factors and CardioRS was rebuilt after introducing the new factor. We rebuilt the models for each gender separately. Table 5.4 presents the sensitivity and specificity under the CardioRS learning-based approach for each event type: CVDA, CHDA, and CHDH.

Table 5.4: Sensitivity and specificity under CRS for men for each event type: CVDA, CHDA and CHDH. NEATER as the base algorithm

Events	Model	Sn(%)	Sp(%)	Model	Sn(%)	Sp(%)
CHDH	CRS_m^1	47.83	63.77	CRS_w^1	59.43	69.37
	CRS_m^2	47.87	75.43	CRS_w^2	68.13	69.60
	CRS_m^3	48.93	59.57	CRS_w^3	73.93	48.50
CHDA	CRS_m^4	44.53	65.60	CRS_w^4	59.27	59.00
CVDA	CRS_m^5	41.60	76.70	CRS_w^5	59.77	72.60
	CRS_m^6	43.10	72.27	CRS_w^6	52.07	72.93
	CRS_m^7	53.03	63.07	CRS_w^7	50.87	70.07

Note that due to the skewed class distribution, the NEATER algorithm was applied to the training data before building the models. Utilization of this data-boosting algorithm increased the availability of cases with class labels of interest, including patients with CVD and CHD events. The investigation of their performance was evaluated in real data with which the models are unfamiliar. We built CardioRS models based on DT algorithms, which were applied to the dataset modified by NEATER. In this step, DTs (J48) were constructed based on the new parameters in terms of total cost (TC). Finally, we used 3-fold cross-validation to empirically validate the results. The three results from the folds are averaged to produce a single estimation.

5.2.4 Comparison and Evaluation

Table 5.5 presents the sensitivity and specificity differences between the two approaches, as well as the NRI improvement for each gender separately. All the models use only the standard factors in their prediction. Conversely, $B^2_{m,w}$ and $CRS^2_{m,w}$ are the models that consider calcium as an additional risk factor for CHDH event prediction.

Table 5.5: Difference of sensitivity, specificity between base models and CRS for all events - NEATER is the base algorithm

Events	ΔSn(%)	ΔSp(%)	NRI	Events	ΔSn(%)	ΔSp(%)	NRI
Men				Women			
CHDH	33.78	-21.28	12.50	CHDH	43.49	-26.67	16.82
	19.64	-13.10	6.54		39.15	-25.34	13.81
	47.67	-39.94	7.73		71.04	-51.44	19.60
CHDA	33.05	-24.69	8.36	CHDA	55.50	-40.66	14.84
CVDA	-57.20	68.14	10.94	CVDA	-27.32	35.90	8.58
	-55.42	59.26	3.84		-24.29	26.86	2.57
	23.51	-25.65	-2.14		32.53	-25.05	7.48

Table 5.3 illustrates that adding CACS to the traditional risk factors for CHDH events significantly improves the sensitivity, and therefore the overall prediction model, in both genders for both risk categories: an NRI improvement of 6.54%

for men and 13.81% for women for both risk categories. These findings clearly indicate the significant improvement in sensitivity and overall performance gleaned by adding calcium to the traditional risk factors. This demonstrates that adding CACS improves the traditional statistical predictive approach and raises the possibility of further extension by incorporating more factors into the existing model. The imbalance ratio between positive and negative events in the dataset has a severe (strong) impact on the overall performance of the prediction model; the recorded specificity is much higher than sensitivity in all models. This indicates that the low sensitivity recorded is directly proportional to the positive events available for building the models. Since the costs of mis-classifications in our models are often disparate, it is essential to improve sensitivity in future models.

There are three main questions which must be addressed with regard to the introduction of our CRS algorithm (which used only 2/3 of the data to build its various models, because 1/3 is reserved for testing). First, how well CRS will perform when given only the traditional risk factor as its input. Second, whether the addition of CAC increases performance, as it did in the traditional approach. Finally, how well CRS will perform in comparison to the traditional statistical approach. To answer these questions, we will refer to Table 5.5, which shows the NRI between approaches.

CRS is superior to the statistical approach in its ability to identify positive instances (subjects with an event): recorded sensitivity of 47.83%, 44.53% and 41.60% for CHDH, CHDA and CVDA, respectively, for male subjects and 59.43%, 59.27%, and 59.77% for CHDH, CHDA and CVDA, respectively, for female subjects, which is higher than the sensitivity recorded from the statistical models. This is essential,

especially because the costs of mis-classifications are often disparate in such cases. Another interesting observation is that adding the calcium score did not dramatically increase our model's sensitivity from 47.83% to 47.87% for males. However, it improved the specificity of the models by 11.66%.

Tables 5.5 illustrates the differences in sensitivity, specificity and NRI improvement between each base model and its equivalent CRS model using the same feature set. For men, $CRS^5_{m,w}$ - $CRS^6_{m,w}$ models have lower sensitivity than the base models, however, the base models records higher sensitivity in the expense of a very low specificity. Model $CRS^7_{m,w}$ has a higher sensitivity for both male and female subjects. For CHDA events, $CRS^4_{m,w}$ has significantly higher NRI than $B^4_{m,w}$. Nonetheless, $CRS^4_{m,w}$ records better sensitivity in all cases. Similarly, when adding the calcium score to the traditional risk factors, $CRS^2_{m,w}$ records an improvement of 6.54% and 13.81%, compared to the base model.

Finally, the specificity and overall performance of CRS models are much higher than the statistical models, given the risk factors used to build $CRS^5_{m,w}$ and $CRS^6_{m,w}$ for CVDA events. The CRS model does not perform as well when built using the same risk factors used to build $CRS^7_{m,w}$ for male subjects.

This clearly illustrates that ML algorithms are capable of producing stronger prediction models which yield better (or at least equivalent) results, as compared to the currently employed statistical methodologies. Moreover, these ML models are built using only two-thirds of the data. Building Cox models requires the use of all data.

Chapter 6

Conclusion

6.1 Summary of Contributions

In this dissertation we have presented new methods to address the imbalanced data problem. We presented a novel game theory formulation for filtering over-sampled data for the case of imbalanced datasets. Specifically, we formulated the problem as a polymatrix game where the solution is reached once Nash equilibrium is found. We evaluated our algorithm on a wide variety of imbalanced datasets using different performance measures and compared it to established state-of-the-art over-sampling methods. The results support our analysis and indicate that the proposed method, NEATER, provides statistically significant better results. In our current implementation, we used replicator dynamics to reach the equilibrium since its evolutionary nature is most suitable for our approach. Other approaches could also be used to

reach equilibrium, and may even do so more efficiently [101, 108]. Improving efficiency will be part of the focus of future work.

We also presented a novel framework which integrates semi-supervised learning and US techniques to improve classification performance for imbalanced datasets. Specifically, semi-supervised learning is used to identify the most relevant instances in the majority. By removing overlapping examples, we establish a well-defined training set. The extensive experimental results described here support our analysis and indicate that the frameworks we have proposed provide statistically significant improvements.

6.2 Future Work

Further enhancements and expansions to the current NEATER algorithm include: (i) using the probabilities generated to further enhance the classifier; (ii) evaluating the algorithm to multi-class imbalanced classification problems where two players i and j with finite possible strategies will have a payoff function which can be represented as a $k_i \times k_j$ matrix. A strategy tuple is a unique choice of actions by each player. For example, a 3-class problem will have $3X3$ partial payoff matrix and so on [128]; and (iii) determining which imbalanced datasets are most likely to take advantage of this approach and estimating the sample size increase needed for optimal classification performance.

For the US-SSL method, aside from extending the algorithm to multi-class imbalanced classification problems, further significant enhancements and expansions to

the current algorithm could include determining the estimate sample size decrease needed for the classifier to provide optimal prediction. Additionally, by knowing the underlying structure of the dataset, we can further categorize the datasets into more specific categories such as: generative, graph-based, heuristic approaches, etc [19]. By doing so, it will help selecting the most suitable semi-supervised learner that has already been established as performing well for the given structure.

Bibliography

[1] A. Agatston, W. Janowitz, F. Hildner, N. Zusmer, M. Viamonte Jr, and R. Detrano. Quantification of coronary artery calcium using ultrafast Computed Tomography. *Journal of the American College of Cardiology*, 15(4):827–832, 1990.

[2] H. Ahn, H. Moon, M. J. Fazzari, N. Lim, J. J. Chen, and R. L. Kodell. Classification by ensembles from random partitions of high-dimensional data. *Computational Statistics & Data Analysis*, 51(12):6166–6179, 2007.

[3] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In Springer, editor, *Machine Learning: ECML 2004*, pages 39–50. Springer, 2004.

[4] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. KEEL data-mining software: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, pages 255–287, April 2010.

[5] B. A. Almogahed and I. A. Kakadiaris. Empowering imbalanced data in supervised learning: A semi-supervised learning approach. In *Proc. International Conference on Artificial Neural Networks*, Hamburg, Germany, Sep 15-19 2014.

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

[7] K. Anderson, P. Odell, P. Wilson, and W. Kannel. Cardiovascular disease risk profiles. *American Heart Journal*, 121(1):293–298, 1991.

[8] G. Assmann, P. Cullen, and H. Schulte. Simple scoring scheme for calculating the risk of acute coronary events based on the 10-year follow-up of the prospective cardiovascular mnster (procam) study. *Circulation*, 105(3):310–315, 2002.

[9] R. Barandela, J. S. Sánchez, V. García, and E. Rangel. Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851, 2003.

[10] G. Batista, R. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[11] D. Bild, D. Bluemke, G. Burke, R. Detrano, A. Diez Roux, A. Folsom, P. Greenland, D. JacobsJr, R. Kronmal, K. Liu, et al. Multi-ethnic study of atherosclerosis: objectives and design. *American journal of epidemiology*, 156(9):871, 2002.

[12] R. Blagus and L. Lusa. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics*, 14(1):106, 2013.

[13] A. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7):1145–1159, 1997.

[14] C. Bunkhumpornpat, K. Sinapiromsaran, and C. Lursinsap. Safe-level-smote: safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In Springer, editor, *Advances in Knowledge Discovery and Data Mining*, pages 475–482. Springer, 2009.

[15] E. Byon, A. Shrivastava, and Y. Ding. A classification procedure for highly imbalanced class sizes. *IIE Transactions*, 42(4):288–303, 2010.

[16] J. Carr, J. Nelson, N. Wong, M. McNitt-Gray, Y. Arad, J. Jacobs, D.R., S. Sidney, D. Bild, O. Williams, and R. Detrano. Calcified coronary artery plaque measurement with cardiac CT in population-based studies: Standardized protocol of Multi-Ethnic Study of Atherosclerosis (MESA) and Coronary Artery Risk Development in Young Adults (CARDIA) study. *Radiology*, 234(1):35–43, 2005.

[17] J. Catto, D. Linkens, M. Abbod, M. Chen, J. Burton, K. Feeley, and F. Hamdy. Artificial intelligence in predicting bladder cancer outcome. *Clinical Cancer Research*, 9(11):4172–4177, 2003.

[18] B. Chaitman, K. Davis, L. Fisher, M. Bourassa, M. Mock, J. Lesperance, W. Rogers, D. Fray, D. Tyras, and M. Judkins. A life table and cox regression analysis of patients with combined proximal left anterior descending and proximal left circumflex coronary artery disease: non-left main equivalent lesions (cass). *Circulation*, 68(6):1163–1170, 1983.

[19] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.

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

[21] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, pages 321–357, 2011.

[22] J. J. Chen, C. A. Tsai, J. F. Young, and R. L. Kodell. Classification ensembles for unbalanced class sizes in predictive toxicology. *SAR and QSAR in Environmental Research*, 16(6):517–529, 2005.

[23] S. Chen, W. Wang, S. Lee, K. Nafa, J. Lee, K. Romans, P. Watson, S. Gruber, D. Euhus, K. Kinzler, J. Jass, S. Gallinger, N. Lindor, G. Casey, N. Ellis, F. Giardiello, K. Offit, and G. Parmigiani. Prediction of germline mutations and cancer risk in the lynch syndrome. *Journal of The American Medical Association*, 296(12):1479–1487, 2006.

[24] X. Chen, E. Song, and G. Ma. An adaptive cost-sensitive classifier. In *Proc.* 2^{nd} *International Conference on Computer Automation Engineering*, pages 699–701, Singapore, February 26-28 2010.

[25] B. C. Christensen, A. E. Houseman, C. J. Marsit, S. Zheng, M. R. Wrensch, J. L. Wiemels, H. H. Nelson, M. R. Karagas, J. F. Padbury, R. Bueno, D. J. Sugarbaker, R. Yeh, J. K. Wiencke, and K. T. Kelsey. Aging and environmental exposures alter tissue-specific dna methylation dependent upon cpg island context. *PLOS Genetics*, 5(8):e1000602, aug 2009.

[26] G. Cohen, M. Hilario, H. Sax, S. Hugonnet, and A. Geissbuhler. Learning from imbalanced data in surveillance of nosocomial infection. *Artificial Intelligence in Medicine*, 37(1):7–18, 2006.

[27] G. Colditz, K. Atwood, K. Emmons, R. Monson, W. Willett, D. Trichopoulos, and D. Hunter. Harvard report on cancer prevention volume 4: harvard cancer risk index. *Cancer Causes and Control*, 11(6):477–488, 2000.

[28] R. Conroy, K. Pyörälä, A. Fitzgerald, S. Sans, A. Menotti, G. De Backer, D. De Bacquer, P. Ducimetiere, P. Jousilahti, U. Keil, I. Njlstad, R. Oganov, T. Thomsen, H. Tunstall-Pedoe, A. Tverdal, H. Wedel, P. Whincup, L. Wilhelmsen, and I. Graham. Estimation of ten-year risk of fatal cardiovascular

disease in Europe: the score project. *European Heart Journal*, 24(11):987–1003, 2003.

[29] R. Cressman. *The stability concept of evolutionary game theory: A dynamic approach*. Springer-Verlag, 1992.

[30] J. A. Cuff and G. J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519, 1999.

[31] D. Delen, G. Walker, and A. Kadam. Predicting breast cancer survivability: a comparison of three data mining methods. *Artificial Intelligence in Medicine*, 34(2):113–127, 2005.

[32] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[33] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[34] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, 2011.

[35] P. Domingos. MetaCost: A general method for making classifiers cost-sensitive. In *Proc. 5th International Conference on Knowledge Discovery and Data Mining*, pages 155–164, San Diego, CA, August 15-18 1999.

[36] K. Driessens, P. Reutemann, B. Pfahringer, and C. Leschi. Using weighted nearest neighbor to benefit from unlabeled data. In *Proc. Advances in Knowledge Discovery and Data Mining*, pages 60–69, Singapore, 2006.

[37] J. Driver, J. Gaziano, R. Gelber, I. Lee, J. Buring, T. Kurth, et al. Development of a risk score for colorectal cancer in men. *The American journal of medicine*, 120(3):257–263, 2007.

[38] C. Drummond, R. C. Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Proc. Workshop on Learning from Imbalanced Datasets II*, volume 11, Washington, DC, August 2003. Citeseer.

[39] R. DAgostino Sr, R. Vasan, M. Pencina, P. Wolf, M. Cobain, J. Massaro, and W. Kannel. General cardiovascular risk profile for use in primary care. *Circulation*, 117(6):743–753, 2008.

[40] C. Elkan. The foundations of cost-sensitive learning. In *Proc. International Joint Conference on Artificial Intelligence*, volume 17, pages 973–978, Seattle, WA, August 2001.

[41] T. Fawcett. Roc graphs: notes and practical considerations for data mining researchers. *Hewlett-Packard Labs Technical Report HPL-2003-4*, 2003.

[42] T. Fawcett. An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[43] Framingham. Framingham heart study, march 2014.

[44] A. Freedman, M. Slattery, R. Ballard-Barbash, G. Willis, B. Cann, D. Pee, M. Gail, and R. Pfeiffer. Colorectal cancer risk prediction tool for white men and women without known susceptibility. *Journal of Clinical Oncology*, 27(5):686–693, 2009.

[45] S. García, A. Fernández, J. Luengo, and F. Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064, 2010.

[46] S. García and F. Herrera. Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy. *Evolutionary Computation*, 17(3):275–306, 2009.

[47] V. García, J. S. Sánchez, and R. A. Mollineda. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, 25(1):13–21, 2012.

[48] G. J. Gordon, R. V. Jensen, 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, Sep 2002.

[49] P. Greenland, J. S. Alpert, G. A. Beller, E. J. Benjamin, M. J. Budoff, Z. A. Fayad, E. Foster, M. A. Hlatky, J. M. Hodgson, F. G. Kushner, et al. 2010 accf/aha guideline for assessment of cardiovascular risk in asymptomatic

adultsa report of the american college of cardiology foundation/american heart association task force on practice guidelines developed in collaboration with the american society. *Journal of the American College of Cardiology*, 56(25):e50–e103, 2010.

[50] P. Greenland, R. Bonow, B. Brundage, M. Budoff, M. Eisenberg, S. Grundy, M. Lauer, W. Post, P. Raggi, R. Redberg, et al. Accf/aha 2007 clinical expert consensus document on coronary artery calcium scoring by Computed Tomography in global cardiovascular risk assessment and in evaluation of patients with chest pain: a report of the american college of cardiology foundation cl. *Journal of the American College of Cardiology*, 49(3):378–402, 2007.

[51] P. Greenland, R. O. Bonow, B. H. Brundage, M. J. Budoff, M. J. Eisenberg, S. M. Grundy, M. S. Lauer, W. S. Post, P. Raggi, R. F. Redberg, G. P. Rodgers, L. J. Shaw, A. J. Taylor, W. S. Weintraub, R. A. Harrington, J. Abrams, J. L. Anderson, E. R. Bates, C. L. Grines, M. A. Hlatky, R. C. Lichtenberg, J. R. Lindner, G. M. Pohost, R. S. Schofield, J. Shubrooks, S. J., J. H. Stein, C. M. Tracy, R. A. Vogel, and D. J. Wesley. ACCF/AHA 2007 Clinical expert consensus document on coronary artery calcium scoring by Computed Tomography in global cardiovascular risk assessment and in evaluation of patients with chest pain: A report of the American College of Cardiolo. *Circulation*, 115(3):402–26, 2007.

[52] I. Guyon, M. N. S. Gunn, and L. Zadeh. *Feature extraction*. Springer, 2006.

[53] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and H. Witten. WEKA data mining software. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.

[54] H. Han, W. Wang, and B. Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing*, volume 3644, pages 878–887. Springer, 2005.

[55] Y. Hayashida, K. Honda, Y. Osaka, T. Hara, T. Umaki, A. Tsuchida, T. Aoki, S. Hirohashi, and T. Yamada. Possible prediction of chemoradiosensitivity of esophageal cancer by serum protein profiling. *Clinical Cancer Research*, 11(22):8042–8047, 2005.

[56] H. He, Y. Bai, E. A. Garcia, and S. Li. ADASYN: adaptive synthetic sampling approach for imbalanced learning. In *Proc. IEEE International Joint Conference on Neural Networks*, pages 1322–1328, Hong Kong, June 2008.

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

[58] H. He and Y. Ma. *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons, 2013.

[59] M. Hein and J. Audibert. Intrinsic dimensionality estimation of submanifolds in r d. In *Proc. Proceedings of the 22nd international conference on Machine learning*, pages 289–296, Germany, August 2005. ACM.

[60] J. Hippisley-Cox, C. Coupland, Y. Vinogradova, J. Robson, M. May, and P. Brindle. Derivation and validation of qrisk, a new cardiovascular disease risk score for the united kingdom: prospective open cohort study. *Bmj*, 335(7611):136, 2007.

[61] J. Hippisley-Cox, C. Coupland, Y. Vinogradova, J. Robson, R. Minhas, A. Sheikh, and P. Brindle. Predicting cardiovascular risk in england and wales: prospective derivation and validation of qrisk2. *British Medical Journal*, 336(7659):1475–1482, 2008.

[62] J. Hofbauer and K. Sigmund. Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 40(4):479, 2003.

[63] R. C. Holte, L. E. Acker, and B. W. Porter. Concept learning and the problem of small disjuncts. In *Proc.* 11^{th} *International Joint Conference on Artificial Intelligence*, volume 1, Detroit, August 1989.

[64] B. Hu and W. Dong. A study on cost behaviors of binary classification measures in class-imbalanced problems. *arXiv preprint arXiv:1403.7100*, page 1, 2014.

[65] T. F. Imperiale, D. R. Wagner, C. Y. Lin, G. N. Larkin, J. D. Rogge, and D. F. Ransohoff. Using risk for advanced proximal colonic neoplasia to tailor endoscopic screening for colorectal cancer. *Annals of internal medicine*, 139(12):959–965, 2003.

[66] T. J. J. Howson. Equilibria of polymatrix games. *Management Science*, pages 312–318, 1972.

[67] T. Joachims. *Learning to classify text using support vector machines: methods, theory and algorithms*. Kluwer Academic Publishers, 2002.

[68] M. Joseph Yeboah, MD, P. Robyn L. McClelland, M. Tamar S. Polonsky, MD, M. Gregory L. Burke, MD, M. Christopher T. Sibley, M. Daniel OLeary, M. Jeffery J. Carr, MD, P. David C. Goff, MD, M. Philip Greenland, and M. David

M. Herrington, MD. Comparison of novel risk markers for improvement in cardiovascular risk assessment in intermediate-risk individuals. *Journal of The American Medical Association*, pages 788–795, 2012.

[69] I. Kaiserman, M. Rosner, and J. Peer. Forecasting the prognosis of choroidal melanoma with an artificial neural network. *Ophthalmology*, 112(9):1608–e1, 2005.

[70] S. Kotsiantis. Decision trees: a recent overview. *Artificial Intelligence Review*, pages 1–23, June 2011.

[71] D. M. Kreps. *Game theory and economic modelling. Clarendon.* Oxford, New York, 1990.

[72] M. Kubat, R. C. Holte, and S. Matwin. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2-3):195–215, 1998.

[73] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets: One-sided selection. In *Proc.* 14^{th} *International Conference on Machine Learning*, pages 179–186, TN, USA, July 8-12 1997.

[74] T. N. Lal, T. Hinterberger, G. Widman, M. Schr&quot and oder, J. Hill, W. Rosenstiel, C. Elger, B. Sch&quot and olkopf, and N. Birbaumer. Methods towards invasive human brain computer interfaces. *Advances in Neural Information Processing System*, pages 737–744, 2005.

[75] C. Leistner, A. Saffari, and H. Bischof. Semi-supervised random forests. In *Proc.* 12^{th} *International Conference on Computer Vision*, pages 506–513, Kyoto, Japan, Oct 2009.

[76] C. E. Lemke and J. T. Howson Jr. Equilibrium points of bimatrix games. *Journal of the Society for Industrial & Applied Mathematics*, 12(2):413–423, 1964.

[77] C. Lemnaru and P. Rodica. Imbalanced classification problems: systematic study, issues and best practices. In Springer, editor, *Enterprise Information Systems*, pages 35–50. Springer, 2012.

[78] J. Li, H. Liu, S. Ng, and L. Wong. Discovery of significant rules for classifying cancer diagnosis data. *Bioinformatics*, 19(suppl 2):ii93–ii102, 2003.

[79] J. Listgarten, S. Damaraju, B. Poulin, L. Cook, J. Dufour, A. Driga, J. Mackey, D. Wishart, R. Greiner, and B. Zanke. Predictive models for breast cancer

susceptibility from multiple single nucleotide polymorphisms. *Clinical Cancer Research*, 10(8):2725–2737, 2004.

[80] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory under-sampling for class-imbalance learning. In *Proc. Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 965–969, Hong Kong, Dec 2006. IEEE.

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

[82] E. Ma, S. Sasazuki, M. Iwasaki, N. Sawada, and M. Inoue. 10-year risk of colorectal cancer: development and validation of a prediction model in middle-aged japanese men. *Cancer Epidemiology*, 34(5):534–541, 2010.

[83] I. Mani and I. Zhang. Knn approach to unbalanced data distributions: a case study involving information extraction. In *Proc. Proceedings of Workshop on Learning from Imbalanced Datasets*, Washington DC, Jan 2003.

[84] A. Maratea, A. Petrosino, and M. Manzo. Adjusted f-measure and kernel scaling for imbalanced data learning. *Information Sciences*, 257:331–341, 2014.

[85] K. McCarthy, B. Zabar, and G. Weiss. Does cost-sensitive learning beat sampling for classifying rare classes? In *Proc. Proceedings of the 1st international workshop on Utility-based data mining*, pages 69–77, NY, August 2005. ACM.

[86] H. H. Meng, G. Z. Li, R. Wang, X. Zhao, and L. Chen. The imbalanced problem in mass-spectrometry data analysis. In *Proc. LNOR 9: The Second International Symposium on Optimization and Systems Biology (OSB108)*, pages 136–143, Lijiang, China, October 2008.

[87] C. Merz, P. Murphy, and D. Aha. UCI repository of machine learning databases. department of information and computer science, University of California, 2012.

[88] MESA. the multi-ethnic study of atherosclerosis, April 2013.

[89] P. M. Murphy and D. W. Aha. UCI repository of machine learning databases. *Machine-readable repository*, University of California, Department of Information and Computer Science, Irvine, CA, 1992.

[90] J. Nash. Non-cooperative games. *Annals of mathematics*, 54(2):286–295, 1951.

[91] J. Nelson, R. Kronmal, J. Carr, M. McNitt-Gray, N. Wong, C. Loria, J. Goldin, O. Williams, and R. Detrano. Measuring coronary calcium on ct images adjusted for attenuation differences1. *Radiology*, 255(2):403–414, 2005.

[92] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[93] U. of Edinburgh. University of edinburgh, march 2014.

[94] N. I. of Health. Third report of the national cholesterol education program (ncep) expert panel on detection, evaluation, and treatment of high blood cholesterol in adults (adult treatment panel iii). final report. Technical Report NIH Publication No. 02-5215., US Department of Health and Human Services, 2002.

[95] S. Oh. Error back-propagation algorithm for classification of imbalanced data. *Neurocomputing*, 74(6):1058–1061, 2011.

[96] P. C. Ordeshook. *Game theory and political theory: an introduction*. Cambridge University Press, 1986.

[97] A. Orriols-Puig and E. Bernadó-Mansilla. Evolutionary rule-based systems for imbalanced data sets. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 13(3):213–225, 2009.

[98] S. Palaniappan and R. Awang. Intelligent heart disease prediction system using data mining techniques. In *Proc. IEEE/ACS International Conference on Computer Systems and Applications*, pages 108–115, Petaling Jaya, Malaysia, March 2008. Ieee.

[99] L. Parthiban and R. Subramanian. Intelligent heart disease prediction system using canfis and genetic algorithm. *International Journal of Biological and Life Sciences*, 3(3):157–160, 2007.

[100] S. Patil and Y. Kumaraswamy. Intelligent and effective heart attack prediction system using data mining and artificial neural network. *European Journal of Scientific Research*, 31(4):642–656, 2009.

[101] R. Porter, E. Nudelman, and Y. Shoham. Simple search methods for finding a Nash equilibrium. *Games and Economic Behavior*, 63(2):642–662, 2008.

[102] J. Quinlan. *C4. 5: programs for machine learning*. Morgan kaufmann, 1993.

[103] S. Ramanna, L. C. Jain, and R. J. Howlett. *Emerging paradigms in machine learning*. Springer Publishing Company, Incorporated, 2012.

[104] E. Ramentol, Y. Caballero, R. Bello, and F. Herrera. Smote-rsb*: a hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. *Knowledge and information systems*, 33(2):245–265, 2012.

[105] B. Raskutti and A. Kowalczyk. Extreme re-balancing for SVMs: a case study. *ACM SIGKDD Explorations Newsletter*, 6(1):60–69, 2004.

[106] P. Ridker, J. Buring, N. Rifai, and N. Cook. Development and validation of improved algorithms for the assessment of global cardiovascular risk in women. *JAMA: the journal of the American Medical Association*, 297(6):611–619, 2007.

[107] P. Ridker, C. Cannon, D. Morrow, N. Rifai, L. Rose, C. McCabe, M. Pfeffer, and E. Braunwald. C-reactive protein levels and outcomes after statin therapy. *New England Journal of Medicine*, 352(1):20–28, 2005.

[108] S. Rota Bulò and I. M. Bomze. Infection and immunization: a new class of evolutionary game dynamics. *Games and Economic Behavior*, 71(1):193–211, 2011.

[109] T. Schneider. Analysis of incomplete climate data: estimation of mean values and covariance matrices and imputation of missing values. *Journal of Climate*, 14(5):853–871, 2001.

[110] S. Seiwerth, N. Stambuk, P. Konjevoda, N. Masic, A. Vasilj, M. Bura, I. Klapan, S. Manojlovic, and D. anic. Immunohistochemical analysis and prognostic value of cathepsin d determination in laryngeal squamous cell carcinoma. *Journal of Chemical Information and Computer Sciences*, 40(3):545–549, 2000.

[111] V. Sheng and C. Ling. Thresholding for making classifiers cost-sensitive. In *Proc. Proceedings of the 21st national conference on Artificial intelligence - Volume 1*, volume 21, pages 476–481, 2006.

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

[113] J. Smith. *Evolution and the theory of games*. Cambridge University Press, 1982.

[114] P. Snow, D. Kerr, J. Brandt, and D. Rodvold. Neural network and regression predictions of 5-year survival after colon carcinoma treatment. *Cancer*, 91(S8):1673–1678, 2001.

[115] T. Sorlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. Eisen, B. Michael, M. V. Rijn, S. Jeffrey, T. Thorsen, H. Quist, J. C. Matese, P. O. Brown, D. Botstein, E. Lønning, and A. Børresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc of the National Academy of Sciences*, 98:10869–10874, Sep 2001.

[116] G. Subbalakshmi, K. Ramesh, and M. Rao. Decision support in heart disease prediction system using naive bayes. *Indian Journal of Computer Science and Engineering (IJCSE)*, 2(2):170–176, 2011.

[117] D. Tao, X. Tang, X. Li, and X. Wu. Asymmetric bagging and random subspace for Support Vector Machines-based relevance feedback in image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1088–1099, 2006.

[118] A. Tewari, M. Issa, R. El-Galley, H. Stricker, J. Peabody, J. Pow-Sang, A. Shukla, Z. Wajsman, M. Rubin, J. Wei, J. Montie, R. Demers, C. Johnson, L. Lamerato, G. Divine, E. Crawford, E. Gamito, R. Farah, P. Narayan, G. Carlson, and M. Menon. Genetic adaptive neural network to predict biochemical failure after radical prostatectomy: a multi-institutional study. *Molecular Urology*, 5(4):163–169, 2001.

[119] K. M. Ting. An instance-weighting method to induce cost-sensitive trees. *Knowledge and Data Engineering, IEEE Transactions on*, 14(3):659–665, 2002.

[120] I. Tomek. Two modifications of CNN. *IEEE Trans. Syst. Man Cybern.*, 6:769–772, 1976.

[121] S. Vadera. Csnl: A cost-sensitive non-linear decision tree algorithm. *ACM Transactions on Knowledge Discovery from Data*, 4(2):6, 2010.

[122] J. Van Hulse, T. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proc.* 24^{th} *International Conference on Machine Learning*, pages 935–942. ACM, 2007.

[123] V. N. Vapnik. An overview of statistical learning theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[124] B. X. Wang and N. Japkowicz. Imbalanced data set learning with synthetic samples. In *Proc. IRIS Machine Learning Workshop*, Canada, June 2004.

[125] J. Wang, T. Jebara, and S. Chang. Graph transduction via alternating minimization. In *Proc. Proceedings of the 25th international conference on Machine learning*, pages 1144–1151, Helsinki, Finland, July 2008. ACM.

[126] J. Wang, T. Jebara, and S. Chang. Semi-supervised learning using greedy max-cut. *The Journal of Machine Learning Research*, 14(1):771–800, 2013.

[127] E. Wei, G. Colditz, E. Giovannucci, C. Fuchs, and B. Rosner. Cumulative risk of colon cancer up to age 70 years by risk factor status using data from the nurses health study. *American Journal of Epidemiology*, 170(7):863–872, 2009.

[128] J. W. Weibull. *Evolutionary game theory*. MIT press, 1997.

[129] G. M. Weiss. Mining with rarity: A unifying framework. *ACM SIGKDD Explorations Newsletter*, 6(1):7–19, 2004.

[130] Wikipedia. Wikipedia - the free encyclopedia, March 2012.

[131] P. Wilson, R. D'Agostino, D. Levy, A. Belanger, H. Silbershatz, and W. Kannel. Prediction of coronary heart disease using risk factor categories. *Circulation*, 97:1837–1847, 1998.

[132] I. Witten and E. Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[133] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1):67–82, 1997.

[134] M. Woodward, P. Brindle, and H. Tunstall-Pedoe. Adding social deprivation and family history to cardiovascular risk assessment: the assign score from the scottish heart health extended cohort (shhec). *Heart*, 93(2):172–176, 2007.

[135] S. Yaghoubi, W. Tang, S. Wang, J. Reed, J. Hsiai, R. Detrano, B. Brundage, et al. Offline assessment of atherosclerotic coronary calcium from electron beam tomograms. *American Journal of Cardiac Imaging*, 9(4):231–236, 1995.

[136] K. Yang, Z. Cai, J. Li, and G. Lin. A stable gene selection in microarray data analysis. *BMC bioinformatics*, 7(1):228, 2006.

[137] S. Yen and Y. Lee. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In S.-V. B. Heidelberg, editor, *International Conference on Intelligent Computing*, volume 344, pages 731–740, Kunming, China, 2006.

[138] S. Yen, Y. Lee, C. Lin, and J. Ying. Investigating the effect of sampling methods for imbalanced data distributions. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4163–4168, Taipei, Oct 2006.

[139] K. Yoon and S. Kwek. An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics. In *Proc. Hybrid Intelligent Systems*, pages 6–pp, Rio de Janeiro, Brazil, Nov 2005.

[140] D. Zhang, W. Liu, X. Gong, and H. Jin. A novel improved smote resampling algorithm based on fractal. *Journal of Computational Information Systems*, 7(6):2204–2211, 2011.

[141] G. Zhang. A modified svm classifier based on rs in medical disease prediction. In *Proc. Computational Intelligence and Design, 2009. ISCID'09. Second International Symposium on*, volume 1, pages 144–147, Changsha, China, Dec 2009. IEEE.

[142] D. Zhou, O. Bousquet, T. N. Lal, , and B. Scholkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16(16):321–328, 2004.

[143] X. Zhu. Semi-supervised learning literature survey. *Computer Science, University of Wisconsin-Madison*, 2:3, 2006.