Faculty of Engineering and Information Technology
University of Technology, Sydney

Ensemble Predictions: Empirical Studies on Learners' Performance and Sample Distributions

A thesis submitted in fulfillment of
the requirements for the degree of
Doctor of Philosophy

by

Guohua Liang

February 2014

CERTIFICATE OF AUTHORSHIP/ORIGINALITY

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signature of Candidate

Acknowledgments

I would like to express my deepest gratitude to my principal supervisor Professor Chengqi Zhang for providing me with such an excellent research environment; for the wonderful opportunities to explore my dreams to become a successful scientist; for his supervision; ongoing encouragement and support. His deep insights, profound knowledge and outstanding contributions to our society have been a great model for me. All of these will benefit my future research career. It has been a great honor for me to be his PhD student.

I would especially like to express my deepest appreciation to my co-supervisor Professor Xingquan Zhu. His guidance and rigorous approach in all academic areas has been of great value, allowing me to make significant academic achievements in a short period of time. I am very grateful for his support and advice.

My warm thanks to all members, visitors and students of the Centre for Quantum Computation & Intelligent Systems, University of Technology, Sydney for their helpful discussions and encouragement.

I would like to dedicate this thesis to my family, my husband Bin Kong and my two sons, Jeffrey Kong and Kevin Kong for their love, support, and understanding, especially, my little son, Kevin. He always asks me, "Mum why you always go to your office doing your work?" Sometimes he talks to himself, "My mum has a lot of work to do"; he also frequently says to me,

"When you finish your work come back home, I will give you some gold stars."

Finally, I would like to express my gratitude to both the Graduate School and the Faculty of Engineering and IT, University of Technology, Sydney for offering financial support with the final stages of my thesis writing.

Contents

List of Figures

List of Tables

Abstract

Imbalanced data problems are among the most challenging in Data Mining and Machine Learning research. This dissertation investigates the performance of ensemble learning systems on different types of data environments, and proposes novel ensemble learning approaches for solving imbalanced data problems. Bagging is one of the most effective ensemble methods for classification tasks. Despite the popularity of bagging in many real-world applications, there is a major drawback on extremely imbalanced data. Much research has addressed the problems of imbalanced data by using over-sampling and/or under-sampling methods to generate an equally balanced training set to improve the performance of the prediction models. However, it is unclear which is the best ratio for training, and under which conditions bagging is outperformed by other sampling schemes on extremely imbalanced data.

Previous research has mainly been concerned with studying unstable learners as the key to ensuring the performance gain of a bagging predictor, with many key factors remaining unclear. Some questions have not been well answered: (1) What are the key factors for bagging predictors to achieve the best predictive performance for applications? and (2) What is the impact of varying the levels of class distribution on bagging predictors on different data environments. There is a lack of empirical investigation of these issues in the literature.

The main contributions of this dissertation are as follows:

1. This dissertation proposes novel approaches, uneven balanced bagging to boost the performance of the prediction model for solving imbalanced problems, and hybrid-sampling to enhance bagging for solving highly imbalanced time series classification problems.

2. This dissertation asserts that robustness and stability are two key factors for building a high performance bagging predictor. This dissertation also derives a new method, utilizing two-dimensional robustness and stability decomposition to rank the base learners into different categories for the purpose of comparing the performance of bagging predictors with respect to different learning algorithms. The experimental results demonstrate that bagging is influenced by the combination of robustness and instability, and indicate that robustness is important for bagging to achieve a highly accurate prediction model.

3. This dissertation investigates the sensitivity of bagging predictors. We demonstrate that bagging MLP and NB are insensitive to different levels of imbalanced class distribution.

4. This dissertation investigates the impact of varying levels of class distribution on bagging predictors with different learning algorithms on a range of data environments, to allow data mining practitioners to choose the best learners and understand what to expect when using bagging predictors.

Chapter 1

Introduction

Finding effective methods and improving predictive performance are of primary concern in all learning applications (Quinlan 1996); this is especially true for developing ensemble learning systems. The aim of ensemble learning is to improve the performance of a prediction model by generating and combining a set of multiple individual models. It is well known that a good ensemble learning system can be constructed if the individual models are accurate and diverse (Dietterich 2000*a*, Hansen & Salamon 1990, Krogh & Vedelsby 1995, Opitz & Shavlik 1996*b*). There are two main approaches to constructing the ensemble learning systems: parallel and serial. This thesis is restricted to parallel ensemble learning method, bagging.

Bagging (Breiman 1996*a*) (bootstrap aggregating), represents a popular ensemble method for improving performance of a prediction model using bootstrap sampling and voting techniques. Despite its promising ability to improve the accuracy of classification tasks and the popularity of bagging in many real-world applications, however, there is a major drawback for predicting the predefined class label on extremely imbalanced data-sets. Learning from imbalanced class distribution is considered to be one of ten

challenging problems in data mining research (Yang & Wu 2006). Imbalanced class distribution (Liang 2012, Liang & Cohn 2013) often causes learning algorithms to perform poorly on the minority class; in addition, the overall accuracy is an ineffective evaluation measure for the imbalanced classification task (Liang & Zhang 2011*b*, Liang, Zhu & Zhang 2011*b*), because it cannot represent the accuracy of minority class (Weiss & Provost 2001); other evaluation metrics must therefore be considered. Much research has addressed the problem of imbalanced data by using over-sampling and/or under-sampling methods to generate an equally balanced training set to improve the performance of the prediction models, but it is unclear what ratio of class distribution is the best for training a prediction model, and under which conditions bagging is outperformed by other sampling schemes in terms of extremely imbalanced classification. There is a shortage of novel sampling schemes for bagging to solve highly imbalanced classification in the literature.

Previous empirical studies (Breiman 1996*a*, Quinlan 1996, Opitz & Maclin 1999, Bauer & Kohavi 1999, Dietterich 2000*b*) have demonstrated that bagging is often more accurate than individual classifiers in the ensemble if the base learners are unstable. However, these studies have mainly been concerned with studying unstable learners as the key to ensuring the performance gain of a bagging predictor, with many key factors remaining unclear; in addition, these studies have not given consideration to learning from imbalanced class distribution nor the altered class distribution, and have only used estimated error rate as an evaluation measure. It is important to understand the performance of bagging with respect to different learning algorithms by considering two factors, robustness and stability, and the effect of varying levels of class distribution on different types of data environments. However, there is also a lack of empirical investigation those of issues in the literature.

This dissertation investigates the performance of ensemble learning systems with respect to different learning algorithms and varying levels of class distribution on various types of data environments, such as medical data, and time series data. It also proposes four novel approaches as follows:

1. A novel unevenly balanced bagging (UBagging) approach for solving extremely imbalanced classification. This novel approach is not only for solving extremely imbalanced classification problems, but also for solving almost balanced classification problems.

2. A novel Hybrid-sampling (H-sampling) to enhance bagging (HBagging) approach for solving highly imbalanced time series classification (HITSC) problems.

3. A novel approach, asserting that both stability and robustness are key requirements for building a high performance bagging predictor, is proposed. Formal definitions of robustness and stability are determined to investigate the performance of bagging predictors with respect to different learning algorithms on 48 data-sets.

4. A novel approach, two-dimensional decomposition of robustness and stability, is proposed to rank base learners into different categories, strong, weak, stable and unstable learners, to investigate the performance of bagging predictors and established under what condition it could be improved.

1.1 Objectives

The aim of this dissertation is to contribute to the investigation of the performance of ensemble learning systems with respect to various learning

algorithms and varying levels of class distribution in different data environments. The goal of this dissertation is to contribute to the solution of extremely imbalanced classification problems by proposing novel ensemble learning approaches, for solving extremely imbalanced classification and HITSC problems. The objectives of this dissertation are therefore as follows:

1. to propose a novel approach, UBagging, to boost the performance of bagging predictors for solving extremely imbalanced classification problems.

2. to propose a novel HBagging approach for solving HITSC problems.

3. to assert that both robustness and stability are key requirements for building a high performance bagging predictor, and to propose two formal definitions of the two factors: instability and robustness.

4. to propose a two-dimensional robustness and stability decomposition to rank the base learners into different categories, namely stable, unstable, strong, and weak learners, to investigate the performance of bagging predictors with respect to different learning algorithms and in relation to which conditions can be improved in terms of learning from original class distribution.

5. to evaluate the effect of varying levels of class distribution on the sensitivity of bagging predictors.

6. to investigate the performance of bagging predictors with respect to different levels of imbalanced class distribution in various types of data environments.

1.2 Contributions

Corresponding to the objectives of my research described in Section 1.1, this dissertation presents empirical studies to address a number of issues:

- How to use an ensemble learning method to boost the performance of bagging predictors for solving extremely imbalanced classification and HITSC problems, and under what condition, bagging is outperformed by other sampling schemes in terms of extremely imbalanced classification and HITSC.
- How to address the two factors, robustness and stability, which are both key requirements for building high performance bagging predictors.
- How to rank base learners into different categories to investigate the performance of bagging predictors with respect to different learning algorithms on 48 data-sets.
- How to investigate the effect of varying levels of class distribution on the sensitivity of bagging predictors.
- How to evaluate the performance of bagging predictors with respect to different levels of imbalanced class distribution in various types of data environments.

The contributions of this study can be summarized as follows:

1. A novel ensemble learning UBagging approach is proposed to boost the performance of a prediction model for solving extremely imbalanced problems. The experimental results demonstrate that the novel UBagging approach is statistically significantly superior to single learner $J48$ (Single$J48$), standard bagging (SBagging), and equally balanced bagging (BBagging) (Liang & Cohn 2013).

2. A novel HBagging approach (Liang 2013) is proposed for solving HITSC problems. The experimental results demonstrate that this HBagging approach is dramatically superior to the exiting approaches (Liang & Zhang 2012*b*, Liang 2012, Cao, Li, Woon & Ng 2011).

3. A novel assertion that both robustness and stability are key factors for building a high performance bagging predictor is proposed. The two key factors, robustness and stability, are formally defined to investigate the performance of bagging predictors with respect to 12 learning algorithms on 48 data-sets. We demonstrate that bagging is influenced by the combination of instability and robustness, and point out that robustness is an important factor for achieving a highly accurate prediction model (Liang, Zhu & Zhang 2011*a*, Liang et al. 2011*b*).

4. A novel categorization of base learners is proposed using two-dimensional robustness and stability decomposition to rank the base learners into different categories. A clear picture of the categorization of 12 base learners is provided (Liang et al. 2011*a*).

5. The effect of varying the levels of class distributions on the sensitivity of bagging is investigated. We demonstrate that bagging MLP and NB are insensitive to different levels of imbalanced class distribution (Liang 2012).

6. The impact of varying the levels of class distributions on the performance of bagging predictors with respect to different learning algorithms on imbalanced data and medical data is investigated (Liang & Zhang 2011*b*, Liang et al. 2011*b*, Liang, Zhu & Zhang 2014).

7. The statistical analyses of the experimental results instil confidence in the validity of the conclusions of this research.

1.3 Organisation

This dissertation is organized as follows:

- Chapter 2 presents the literature review.
- Chapter 3 proposes a novel UBagging approach for solving the extremely imbalanced classification problems.
- Chapter 4 empirically evaluates the performance of bagging predictors with respect to different learning algorithms.
- Chapter 5 investigates the sensitivity of bagging predictors to imbalanced class distribution.
- Chapter 6 presents empirical evaluations of the impact of varying levels of class distribution on bagging performance.
- Chapter 7 investigates the performance of bagging predictors on specific domains.
- Chapter 8 proposes a novel H-sampling to enhance bagging approach for solving HITSC problems and compares the performance of under-sampling, over-sampling and H-sampling techniques and state of the art time series classification methods on HITSC.
- Chapter 9 draws conclusions and proposes future work.

1.4 Publications Related to the Thesis

During my PhD study, my published, accepted and submitted papers are as follows:

The published and accepted papers:

1. Liang, G. 2013, 'An effective method for imbalanced time series classification: Hybrid-sampling', Proceedings of the 26th Australasian Joint Conference on Artificial Intelligence, AI 2013, Dunedin, New Zealand, pp. 374-385. (part of Chapter 8).

2. Liang, G. & Cohn, A.G. 2013, 'An effective approach for imbalanced classification: Unevenly balanced bagging', Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013, Washington, USA, pp. 1633-1634. (part of Chapter 3).

3. Liang, G., Zhu, X. & Zhang, C. 2014, 'The effect of varying levels of class distribution on bagging with different algorithms: An empirical study', International Journal of Machine Learning and Cybernetics, IJMLC, vol 5, no. 1, pp. 63-71. (part of Chapter 6).

4. Liang, G. 2012, 'An investigation of sensitivity on bagging predictors: An empirical approach', Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI 2012, Toronto, Canada, pp. 2439-2440. (part of Chapter 5).

5. Liang, G. & Zhang, C. 2012, 'An efficient and simple under-sampling technique for imbalanced time series classification', Proceedings of the ACM International Conference on Information and Knowledge Management, CIKM 2012, pp. 2339-2342. (part of Chapter 8).

6. Liang, G. & Zhang, C. 2012, 'A comparative study of sampling methods and algorithms for imbalanced time series classification', Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence, AI 2012, Sydney, Australia, pp. 637-648. (part of Chapter 8).

7. Liang, G., Zhu, X. & Zhang, C. 2011, 'An empirical study of bagging predictors for different learning algorithms', Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, USA, pp. 1802-1803. (part of Chapter 4).

8. Liang, G., Zhu, X. & Zhang, C. 2011, 'An empirical study of bagging predictors for imbalanced data with different levels of class distribution', Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence, AI 2011, pp. 213-222.(part of Chapter 6).

9. Liang, G. & Zhang, C. 2011, 'An empirical evaluation of bagging with different learning algorithms on imbalanced data', Proceedings of the 7th International Conference on Advanced Data Mining and Applications, ADMA 2011, pp. 339-352. (part of Chapter 6).

10. Liang, G. & Zhang, C. 2011, 'Empirical study of bagging predictors on medical data', Proceedings of the 9th Australian Data Mining Conference, AusDM 2011, pp. 31-40. (Chapter 7).

    The submitted paper under-review:

11. Liang, G., Cohn, A.G., Wu, X. 2014, 'Comparison of variant bagging for imbalanced classification', Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI 2014,. (part of Chapter 3).

Chapter 2

Related Work

2.1 Ensemble Learning

Ensemble learning has been a popular method of boosting the performance of prediction models (Cieslak 2010) and has the capacity to improve the performance of base learners (Webb & Zheng 2004). It has been one of the most active areas of research in supervised learning to study methods for constructing good ensembles of classifiers (Dietterich 2000*a*). Quite a large volume of research has focused on ensemble learning, both theoretically (Hansen & Salamon 1990, Krogh & Vedelsby 1995) and empirically (Hashem 1997, Opitz & Shavlik 1996*a*, Opitz & Shavlik 1996*b*) demonstrating that a good ensemble requires the individual classifiers in the ensemble to be accurate and diverse (Opitz & Maclin 1999). Ensemble learning consists of learning methods that construct a set of trained classifiers to classify new instances in the test-set by taking a vote of their predictions from those individual trained classifiers. There are two main approaches to the construction of ensembles of classifiers: parallel and serial (Tuv 2006). Bagging (Breiman 1996*a*) and Boosting (Freund & Schapire 1996) are two representative techniques.

It is well known that the performance of an ensemble prediction model is better than the performance of any of its individual classifiers if the base learners in the ensemble are highly accurate and diverse (Dietterich 2000*a*, Hansen & Salamon 1990). Therefore, accuracy and diversity are two important factors for building a high performance ensemble prediction model.

2.1.1 General Ensemble Learning Approaches

Dietterich reviewed the ensemble learning algorithms and summarized the two primary approaches to designing ensemble learning algorithms (Dietterich 2000*a*, Dietterich 2003) as follows:

1. the first approach is to construct each classifier independently to make a set of classifiers accurate and diverse. Several ways to force diversity are as follows:

    - the first way is to generate different subsets of training data to construct multiple classifiers, e.g., Bagging, "Bootstrap Aggregating" (Breiman 1996*a*) uses bootstrap sampling techniques to randomly select samples with replacement to form different subsets of training data. The sample size of the subsets of training data m is the same as the sample size of the original training data. The final prediction is made by taking a majority vote of a set of the predictions of trained classifiers.
    - the second way is to generate different subsets of input features to force the diversity, e.g., Cherkauer selected different subsets of the input features to group together features that were based on different image processing operations, for instance, principal component analysis and the fast Fourier transform (Cherkauer 1996).

- the third way is to use the technique called error-correcting output coding (ECOC) (Dietterich & Bakiri 1995) to generate the output labels of the training data, e.g., the combined ECOC with ADABOOST called ADABOOST.OC (Schapire 1997), which is superior to the ECOC method and bagging.
- the fourth way is to inject randomness into the learning algorithm, e.g., randomized trees (Dietterich 2000*b*), random subspace method (Ho 1998), and random decision forests (Breiman 2001), which combine bagging with the random subspace method.

2. the second approach uses an additive model to construct a set of component models, and the prediction is made by taking the weighted sum of a set of component models, e.g., Adaboost (Freund & Schapire 1996) algorithm is an effective method for constructing an additive model (Dietterich 2003).

   - Let $d_l(x_i)$ be the weight on data point x_i, during iteration l of the algorithm.
   - Initially, a weight $d_1(x_i) = \frac{1}{m}$ (m is the number of data points) is assigned to all training data points i.
   - In each iteration, the weighted error is computed and applied to update the weights on the training examples.
   - The final classifier is constructed by a weighted vote of the individual classifiers. Each classifier is weighted according to its accuracy on the weighted training set that it was trained on.

2.1.2 Empirical Studies on Ensemble Learning

Numerous empirical studies have compared the performance of bagging, boosting and other ensemble methods with different classification methods (Quinlan 1996, Opitz & Maclin 1999, Bauer & Kohavi 1999, Dietterich 2000*b*, West, Dellana & Qian 2005, Banfield, Hall, Bowyer & Kegelmeyer 2007, Lopes, Scalabrin & Fernandes 2008, Kim & Kang 2010). For example, Banfield et al. conducted an experimental evaluation of bagging as a baseline against seven other randomization-based ensemble techniques (Banfield et al. 2007). These investigations mainly focused on comparing the performance of standard bagging and other ensemble methods, for instance, boosting, random subspaces, three variations of random forests and randomized C4.5. The bagging method has been used in many real world applications, such as decision support application in the health care field, diagnosis model for the medical field, face recognition, protein structural class prediction, and software engineering prediction problems. Tu, Shin & Shin applied a bagging algorithm for a real world application, the diagnosis of Heart disease to identify the warning signs of Heart disease in patients (Tu, Shin & Shin 2009*b*). This research also compared the effectiveness of the bagging predictor with single learner decision tree J4.8 on four data-sets of Heart disease database from the UCI KDD Archive. The research compared three measures, sensitivity, specificity and accuracy, to evaluate the accuracy of classification. The experimental results showed that the bagging method is more effective than the single learner decision tree. An experimental comparison of LibSVMs, C4.5, bagging C4.5, AdaBoosting C4.5 and Random Forest five classification methods on seven micro-array cancer data-sets has been conducted (Hu, Li, Plank, Wang & Daggard 2006). This research compared the average accuracies of ten-fold cross validation tests to confirm the findings that all ensemble methods out-perform C4.5.

2.2 Bagging

Section 2.2 includes four subsections as follows: (1) Subsection 2.2.1 presents the basic concept and framework of bagging; (2) Subsection 2.2.2 shows the bagging algorithm; (3) Subsection 2.2.3 indicates the advantages of the bagging predictor; and (4) Subsection 2.2.4 presents the background of, and approaches to, bagging.

2.2.1 Basic Concept and Framework of Bagging

Subsection 2.2.1 presents the basic concept and framework of bagging.

Bagging represents a set of classifiers C_k (integer k indicates the numbers of bootstrap samples) which are trained from a set of bootstrap samples D_k to form an ensemble method for prediction, and its function is to predict new samples by a set of classifiers; a final prediction is made by taking a majority vote of individual classifiers.

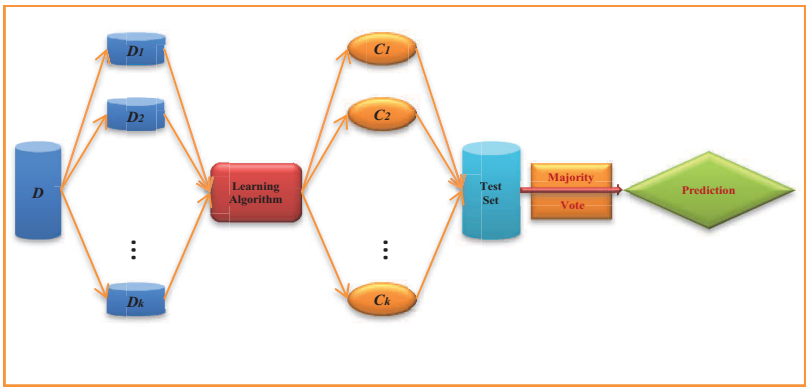

Figure 2.1: Framework of bagging

Figure 2.1 illustrates the basic framework of a bagging prediction model using bootstrap sampling and voting techniques to improve the performance of the bagging prediction model. Bagging is known as bootstrap aggregating.

Firstly, for each of the bootstrap samples $(D_1, D_2, ..., D_k)$, a new training set D_k is randomly drawn from the original training set D of m instances with replacement conducted by repeated drawing m times. Each bootstrap sample therefore contains the same number of m instances as the original training set D; some instances may appear many times, while some instances may not appear. Secondly, the k bootstrap samples of a training set with m instances will generate k classifiers $(C_1, C_2, ..., C_k)$. Finally, the unseen instance x of the test set will be predicted by applying each of the k classifiers C_k (integer $k = 1$ to k) and a final decision C^* is made by majority vote of the individual classifiers $(C_1, C_2, ..., C_k)$.

2.2.2 Bagging Algorithm

The bagging algorithm is indicated in Algorithm 1.

Algorithm 1: Bagging

Output: A composite model, C^*.

Method:

for $i = 1$ **to** k **do**

Create bootstrap sample of size n, D_i by sampling D with replacement;

Train a base classifier model C_i from D_i;

end

To use the composite model, C^* for Test set T on a instance, x and it's true class label is y:

$$C^*(x) = \arg\max_y \sum_i \delta\left(C_i(x) = y\right)$$

Delta function $\delta(\cdot) = 1$ if argument is true, else 0.

2.2.3 Advantages of Bagging

Subsection 2.2.3 presents the advantages of bagging predictors as follows:

1. Bagging is one of the most popular and effective parallel ensemble learning methods (Goebel 2004).

2. It requires less memory than boosting (Freund & Schapire 1996).

3. It improves the performance of the prediction models if the base learners are unstable;

4. It is simply and easy to implement by using random sampling technique to generate bootstrap samples with replacement to build a set of trained classifiers and by using the voting technique, majority vote to make the final prediction.

5. It improves predictive accuracy, and has been applied to wide range of real world applications.

6. It is more robust for classification noise than boosting; although boosting can reduce more classification error than bagging on average, degradation can occur on some data-sets.

2.2.4 Bagging Background and Approaches

Bagging (Breiman 1996*a*) uses bootstrap samples (Efron & Tibshirani 1993) to build a set of classifiers to form a prediction model; the final decision of the prediction model is made by taking a majority vote of the predictions of the individual classifiers in the ensemble. Breiman points out that instability is an important factor in improving the accuracy of a prediction model (Breiman 1996*a*). Bagging is widely accepted as a variance-reduction technique, and is therefore mostly applied to unstable,

high variance algorithms (Tuv 2006). Many theories (Friedman 1997, Domingos 2000, Kohavi & Wolpert 1996, Kong & Dietterich 1995, Breiman 1996*b*, Valentini & Dietterich 2002) have been proposed on the effectiveness of bagging for classifications based on bias and variance decomposition (Opitz & Maclin 1999).

Theoretical investigations of why bagging works have been presented by many researchers (Friedman & Hall 2007, Bühlmann & Yu 2002, Buja & Stuetzle 2000, Buja & Stuetzle 2006). Breiman reported empirical evidence that bagging is a variance reduction technique and that the effectiveness of bagging relies on the instability of the base learner (Breiman 1996*a*), while Bauer and Kohavi also indicated that bagging reduces the bias portion of the error (Bauer & Kohavi 1999). In addition, Buja and Stuetzle conducted a theoretical investigation to understand bagging by a simple real-valued U-statistic of i.i.d. data. In their simulations, they applied bagging to CART trees, and observed that bagging CART trees can reduce both bias and variance. Therefore, their claim that bagging always reduces variance is in fact not true (Buja & Stuetzle 2006).

Previous empirical studies (Breiman 1996*a*, Opitz & Maclin 1999, Quinlan 1996, Bauer & Kohavi 1999, Dietterich 2000*b*) have demonstrated that bagging is often more accurate than any of the individual base learners in the ensemble if the base learners are unstable. However, most previous experimental research only focuses on comparing the performance of bagging and other ensemble methods with one or two base learners, such as $J48$ and neural network, and only consider one factor, instability, as a key factor. None of them has compared the performance of various bagging predictors against one another, nor has most previous research given consideration to learning from imbalanced class distribution nor the altered class distribution, and has only used accuracy/error rate evaluation metric. Furthermore, there are inexplicit

measures to rank the base learners into different categories to analyze the effectiveness of bagging. To the best of our knowledge, no previous research has made such extensive comparisons that rank the bagging predictors underlying base learners over a large number of data-sets to provide a full comparison of the overall performance of bagging predictors. There is a distinct advantage to including 12 learning algorithms on various types data environments in this dissertation.

There is a debate in the scientific community as to whether bagging can improve the performance of SVM. Some researchers have reported that SVM are stable classifiers and bagging is not expected to improve the performance of SVM and may cause a slight deterioration (Buciu, Kotropoulos & Pitas 2006). However, different results on the performance of bagged SVM have been observed, and other researchers have demonstrated an improvement in bagged SVM performance (Kim, Pang, Je, Kim & Bang 2002, Valentini & Dietterich 2003).

Alternative bagging methods to improve the performance of the prediction model have been proposed (Valentini & Dietterich 2003, Zaman & Hirose 2008, Zhu, Bao & Qiu 2008, Zhu & Yang 2008, Zhu 2007, Su, Khoshgoftarr & Zhu 2008, Hothorn & Lausen 2003, Leung & Parker 2003, Frank & Pfahringer 2006, Zaman & Hirose 2009, Collobert, Bengio & Bengio 2002), and some bagging has solved imbalanced data problems (Liu, Wu & Zhou 2009, Zhu & Yang 2008, Zhu 2007, Kang & Cho 2006, Li 2007, Molinara, Ricamato & Tortorella 2007).

Despite its promising capabilities in improving the accuracy of classification tasks and the popularity of bagging in many real-world applications, there is a major drawback for solving extremely imbalanced classification problems. Much research has addressed the problem of imbalanced data by using over-sampling or under-sampling methods to generate a more equally balanced training set to improve the performance

of the prediction models. However, it is unclear what ratio is the best for training, and under which conditions bagging is outperformed by other sampling schemes in extremely imbalanced data. In addition, some questions have not been clearly answered. The key area of concern is the performance of bagging predictors with respect to different learning algorithms on various types of data environments, and with respect to various levels of class distribution in the existing research.

2.3 Statistical Test

To conduct a rigorous and fair analysis, non-parametric tests were performed for the statistical comparison of learners: the Wilcoxon signed-rank test for the comparison of two learners in Subsection 2.3.1, and the Friedman test with the corresponding post-hoc Nemenyi test for the comparison of multiple learners in Subsection 2.3.2 (Demšar 2006).

2.3.1 Wilcoxon Signed-rank Test

The Wilcoxon signed-rank test is a non-parametric statistical hypothesis test which is considered to be an alternative to the paired t-test. Its main difference from a t-test is that it does not require assumptions to be made about the populations of a normal distribution. This test is the most accurate non-parametric test for paired data to determine whether there is a difference between paired samples. The Wilcoxon signed-rank test is considered to be safe from a statistical point of view and is more powerful than the t-test when test conditions cannot meet the assumption requirements of a parametric test (Demšar 2006). We therefore performed this test to determine whether there really is an improvement of performance between the two learners, bagging and single learner.

2.3.2 Friedman Test and Post-hoc Nemenyi Test

Both Friedman and post-hoc Nemenyi tests are non-parametric for comparing multiple algorithms over multiple data-sets.

Firstly, all the algorithms are ranked on each data-set, giving the best performing algorithm the rank of 1, the second best rank 2, and so on. If there are ties, average values are assigned.

Secondly, the average rank of the algorithm is obtained by equation 2.1 , where r_i^j is the rank of the $j-th$ of d algorithms on the $i-th$ of N data-sets.

$$R_j = \frac{1}{N} \sum_i r_i^j \tag{2.1}$$

Finally, the Friedman test compares the average ranks of algorithms and checks whether there is a significant difference between the mean ranks. The Friedman statistic is calculated by equation 2.2.

$$\chi_F^2 = \frac{12N}{d(d+1)} [\sum_j R_j^2 - \frac{d+(d+1)^2}{4}] \tag{2.2}$$

where N is the number of data-sets, d is the number of algorithms compared, and R_j is the average rank of algorithms. This statistic is χ_F^2 distributed with $k-1$ degrees of freedom.

The Null Hypothesis of this test states that the performances of all algorithms are equivalent. If the Null Hypothesis is rejected, it does not determine which particular algorithms differ from one another. A post-hoc Nemenyi test is needed for additional exploration of the differences between mean ranks to provide specific information on which mean ranks are significantly different from others to identify them. The critical difference is calculated by equation 2.3.

$$CD = q_\alpha \sqrt{\frac{d(d+1)}{6N}} \tag{2.3}$$

The critical values q_α are based on the studentized range statistic divided by $\sqrt{2}$ (Demšar 2006). If the mean ranks are different by at least the critical difference, the performance of two learners is significantly different.

2.4 Sampling Techniques

Sampling techniques are widely used to treat imbalanced class distribution problems (Weiss & Provost 2003, Chawla, Bowyer, Hall & Kegelmeyer 2002, Chawla, Lazarevic, Hall & Bowyer 2003, Han, Wang & Mao 2005, Bunkhumpornpat, Sinapiromsaran & Lursinsap 2009). There are two main categories of sampling methods: under-sampling the majority class (Liu et al. 2009, Kubat & Matwin 1997) and over-sampling the monitory class (Chawla et al. 2002, Chawla et al. 2003, Han et al. 2005, Bunkhumpornpat et al. 2009, Bunkhumpornpat, Sinapiromsaran & Lursinsap 2011) to modify the degree of class distribution to any desired level (Batista, Prati & Monard 2004).

There are advantages and disadvantages to using under-sampling and over-sampling methods as follows:

- The advantages of under-sampling: only uses a subset of the majority class for training, so the training process becomes faster, and thus is very efficient.

- The main disadvantages of under-sampling may lose important and useful information for training and may degrade the performance of the prediction models, even though it significantly reduces the computational cost of training, because only a proportion of the majority class examples are selected to train prediction models.

- The main disadvantages of over-sampling are that over-sampling dramatically increases the computational cost of training and training

time, and may cause over-fitting, even though it maintains the important information for training, because additional large number of new positive examples with high dimensional features are generated to balance the training set for HITSC (Liang & Zhang 2012*b*).

Random under-sampling (RUS) and random over-sampling (ROS) are two of the simplest sampling methods in statistics (Cochran 1977) that are used as baseline methods to deal with imbalanced class distribution problems (Laurikkala 2001). For example, RUS selects all positive class samples from the minority class, sample size P into a sub-set S, then randomly selects negative class samples from majority class samples with the same sample size P into the sub-set S to form a balanced sub-set S. The simple random under-sampling method reduces the size of the majority class to compensate for the imbalance (Yang, Zhang, Zhou & Zomaya 2011). Even though the training process becomes faster, and computationally efficient, it may lose potentially useful information. On the other hand, ROS randomly duplicates instances of the minority class to balance the data-set.

A number of advanced intelligent approaches have been introduced as follows:

1. The one-sided selection procedures (Kubat & Matwin 1997) remove the noise or duplicated instances from the majority class, by keeping all the positive samples and randomly selecting a representative subset of the negative sample in a subset C, then removing the number of redundant negative samples by using 1-NN rule with the subset C to re-classify the training samples and adding the mis-classified training samples to subset C, next removing the noisy and borderline majority samples to form a new training set T.

2. NCL (Laurikkala 2001) introduced Neighborhood Cleaning Rule as an under-sampling technique by balancing imbalanced class distribution

with data reduction and comparing three under-sampling methods. The experimental results determined that NCL outperformed simple random under-sampling and one-side selection methods.

3. SMOTE (Chawla et al. 2002) is considered to be the state of the art over-sampling technique (Bunkhumpornpat et al. 2011). This technique created synthetic minority class instances by selecting some of the nearest minority neighbors of a minority instance (MI), and generating new minority class instances along the lines between MI and each nearest minority neighbor. SMOTE reduced the imbalanced class distribution without causing over-fitting (Yen, Lee, Lin & Ying 2006), but may result in an overgeneralization problem, as the drawback of SMOTE that it blindly generates synthetic minority class instances without considering the size of the majority class (Yen et al. 2006, Bunkhumpornpat et al. 2011).

4. Borderline-SMOTE (Han et al. 2005) divides positive instances into three regions: noise, borderline, and safe. It uses the same over-sampling technique as SMOTE, but it only generates the synthetic instances from the borderline instances of the positive instances, which is different from the SMOTE technique which over-samples all instances of the positive class.

5. The Safe-Level-SMOTE (Bunkhumpornpat et al. 2009) technique defines a safe level and a safe level ratio. The safe level is used to assign a safe level to each positive instance before the synthetic instances are generated, and the safe level ratio is used to select the safe positions for generating synthetic instances. This technique is designed to improve over-sampling technique.

In the literature, previous studies comparing under-sampling and

over-sampling with decision trees $C4.5$ indicate that under-sampling is more effective than over-sampling methods (Drummond, Holte et al. 2003, Domingos 1999, Ling & Li 1998, Liu et al. 2009). For example, compared under-sampling and over-sampling techniques, and respectively reported that under-sampling produced better classifiers for $C4.5 - Rules$ and better lift index for boosted $C4.5$. Drummond et al. used cost curve to analyze their experimental results and showed that under-sampling produce a reasonable sensitivity to change in mis-classification cost and class distribution, while over-sampling is ineffective, often producing little or no change in performance (Drummond et al. 2003). In addition, under-sampling is an efficient and popular method for learning from imbalanced class distribution (Liu et al. 2009).

Japkowicz compared several sampling methods and determined that both over-sampling and under-sampling methods are very effective in dealing with imbalanced class distribution problems, and there is no significant advantage in using sophisticated over-sampling and under-sampling methods over simple random over-sampling and under-sampling methods (Japkowicz 2000). Comparative studies of various re-sampling techniques show that simple RUS and ROS perform better than the intelligent techniques mentioned above (Batista et al. 2004, Van Hulse, Khoshgoftaar & Napolitano 2007).

2.5 Evaluation Metrics

Accuracy is a commonly used measure for evaluating the performance of a classifier in general terms. However, it is an ineffective metric for measuring the performance of a prediction model on extremely imbalanced data-sets. As in real world applications, the proportion of the minority class is much smaller than the whole population. The minority class is the class in which

users are interested and normally a high prediction accuracy is required in a minority class; however, accuracy or error rate has limitations in evaluating the performance of a classifier on a minority class (Fawcett 2006). We therefore select True Positive Rate (TPR), Geometric mean (G_{mean}), Receiver Operating Characteristic (ROC) curve, and Area Under the ROC curve (AUC) as evaluation metrics for this empirical study.

Table 2.1: Confusion matrix for a binary classification problem

	Predicted Positives	Predicted Negatives
Actual Positives (P)	True Positives (TP)	False Negatives (FN)
Actual Negatives (N)	False Positives (FP)	True Negatives (TN)

Table 2.1 presents the confusion matrix for a binary classification problem. The confusion matrix indicates the differences between the true and predicted class samples. The columns represent the Predicted Positives and Negatives in each class; the rows represent the Actual Positives (P) and Negatives (N) in each class.

In this thesis, we consider the minority class as the positive class and the majority class as the negative class. Following this convention, in Table 2.1, True Positives (TP) refers to the number of positive instances correctly classified as the positive class; True Negatives (TN) refers to the number of negative instances correctly classified as the negative class; False Positives (FP) refers to the number of negative instances incorrectly classified as the positive class; and False Negatives (FN) refers to the number of positive instances incorrectly classified as the negative class (Chawla 2010). TPR and TNR evaluate the performance of a binary classification algorithm directly

on the minority class and the majority class respectively. TPR refers to the proportion of the minority class that has been correctly classified as a positive class, while TNR refers to the proportion of the majority class that has been correctly classified as a negative class. The G_{mean} of the accuracy rate of the majority class and minority class is recommended as a performance measure to compare different algorithms by monitoring the accuracy rates of both the majority and the minority classes (Ng & Dash 2006, Provost & Fawcett 2001), and is suggested as a performance measure to assess the performance of learning methods for imbalanced learning (Ng & Dash 2006, Provost & Fawcett 2001, He & Garcia 2009). The formulas of the evaluation measures are defined as follows:

$$TPR = Recall = \frac{TP}{TP + FN} \tag{2.4}$$

$$FPR = \frac{FP}{FP + TN} \tag{2.5}$$

$$TNR = \frac{TN}{TN + FP} \tag{2.6}$$

$$FNR = \frac{FN}{FN + TP} \tag{2.7}$$

$$G_{mean} = \sqrt{TPR * TNR} \tag{2.8}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.9}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.10}$$

$$F_{\beta} = (1 + \beta)\frac{Recall * Precision}{\beta^2 * Precision + Recall} \tag{2.11}$$

Equations 2.4 to 2.11 present the formulas of True Positive Rate (TPR), *Recall*, False Positive Rate (FPR), True Negative Rate (TNR), False Negative Rate (FNR), G_{mean}, *Accuracy*, *Recall*, *Precision*, and F_{value}, respectively.

2.5.1 *ROC*

A *ROC* curve, a two-dimensional plot, is used to plot the False Positive Rate (FPR), and True Positive Rate (TPR) on the x-axis and y-axis, respectively. In the ROC plot, the upper left point (0,1) stands for "perfect point", presenting 100% true positives and zero false positives, while the point (1,0) is the least desired point, called "*ROC* Hell" (Qin 2005), presenting zero true positives and 100% false positives.

In the *ROC* space, one point is better than another if it is close to the "perfect point", (Provost & Fawcett 1997). In this study, a *ROC* curve is used to represent the performance of each bagging predictor at nine different levels of class distribution in Table 6.1.

The *ROC* is a well known performance metric for evaluating and comparing algorithms. In the literature, previous work anticipates that sampling will produce the same effect as moving the decision threshold or adjusting the cost matrix, and experimental results demonstrate that the over- and under- sampling procedures produce *ROC* curves almost identical to those produced by varying the decision threshold of Naïve Bayes (Maloof 2003). Moreover, *ROC* has been considered as an alternative measure for comparing the performance of classifiers across the entire range of class distributions and error costs (Provost, Fawcett & Kohavi 1998, Ling, Huang & Zhang 2003).

In addition, *ROC* is a good way of visualizing the performance of a prediction model, and *AUC* is a desirable way to obtain a single figure as a

measure of comparing a number of different prediction models (Bradley 1997). *AUC* (Han et al. 2005, Bradley 1997) is not biased against the minority class and it has an important statistical property (Fawcett 2004), so it is commonly used as an evaluation criterion to assess the average performance of classifiers on data with imbalanced class distribution (Fawcett 2006, Kotsiantis, Kanellopoulos & Pintelas 2006, He & Garcia 2009). The calculated *AUC* of *ROC* curves is considered as an evaluation metric to compare bagging and single learners over multiple imbalanced data-sets, and also to examine the group of comparisons of *ROC* curves between 12 bagging predictors and single learners on the *Diabetes* data-set.

Equations 2.4 and 2.5 present the formulas TPR and FPR, respectively. In our study, when the class distribution is varied, nine pairs of (FPR, TPR) are used to plot a *ROC* curve, so each *ROC* curve represents the performance of prediction models at different levels of class distribution.

2.5.2 How to Calculate *AUC* of *ROC*

The simple trapezoidal method is used to calculate *AUC* of *ROC* as the sum of the areas of trapezoids in the *ROC* space in this study. Each *ROC* curve is formed by i pairs of (FPR_i, TPR_i). The formula of the area of each trapezoid is indicated as follows (Slaby 2007):

$$S = \frac{(y_i + y_{i-1})(x_i - x_{i-1})}{2} \tag{2.12}$$

$$S = \frac{(TPR_i + TPR_{i-1})(FPR_i - FPR_{i-1})}{2} \tag{2.13}$$

The FPR is sorted from small to large. In the literature (Slaby 2007), the Matlab notation of the AUC of ROC is given as follows:

$$AUC = sum(\frac{(tpr(2:n) + tpr(1:n-1)) * (fpr(2:n) - fpr(1:n-1))}{2}) \tag{2.14}$$

2.6 Basic Learning Algorithms

The 12 most common learning algorithms have been selected for this study from the WEKA implementation, and default parameter settings are used .

- We first select four of the top ten algorithms (Wu, Kumar et al. 2008):
    1. $C4.5$ (Quinlan 1986) decision trees ($J48$) is proposed, which is based on gain ratio to select the splitting attribute. When $J48$ is used in WEKA, it is a unstable learner;
    2. Support Vector Machines (SVM), a complex model for classification which uses mapping to transform the original training data into a higher dimension and the decision boundary, is determined by finding the optimal separating hyper-planes, and default setting is selected from the WEKA implementation for this study; the experimental result indicates that it is a stable learner, however in the literature, there is debate as to whether it is a stable learner or an unstable learner.
    3. Naïve Bayes (NB) learner based on Bayes theorem, a simple, yet effective learner for large data-sets. It is a stable learner.
    4. K-nearest-neighbors (KNN), a lazy learner in the WEKA implementation, is used for this study with the default setting; it is a stable learner.

- Next the most commonly used neural network algorithm, Multi-layer Perceptron (MLP) with default setting is selected from the WEKA implementation. In the literature, it is mostly considered to be a unstable learner, however, the experimental results show that it is a strong and stable learner, and has a similar value of stability as KNN.
- Then we select four family tree learners: Random Tree (RandTree), RepTree, Naïve Bayes Tree (NBTree) and Decision Stump (DStump). They are unstable learners.
- Finally, we select three rule learners: PART, Decision Table (DTable), and OneR. They are unstable learners.

2.7 Benchmark Data-sets

Table 2.2 gives a summary of the characteristics of the 48 data-sets used in this experiment. The first and fourth columns indicate the ID number and the name of the data-set. The second and fifth columns present the information about the data itself which includes the number of attributes and instances, the number of attributes excluding the class and the number of instances in each data-set. The third and sixth columns present the information about the class of each data-set, and the number of classes on each data-set. The selection of the 48 data-sets covers the number of instances, which vary from small to large (up to 20,000), the number of attributes, which vary from five to 70, and the number of classes, which vary from binary classes to multiple classes (up to 29).

2.7.1 Imbalanced Data-sets

Table 2.3 shows a summary of the characteristics of the 14 imbalanced data-sets. The data-sets were employed using different criteria, such as the number

of instances from 57 up to 3772, the number of attributes from seven up to 61, and the frequency of each class from almost balanced to extremely imbalanced.

2.7.2 Selection of Medical Data-sets

A summary of the characteristics of the eight imbalanced medical data-sets is displayed in Table 2.4. The selected medical data-sets are binary classes. The selection of the eight data-sets covers the number of instances, which varies from small to large up to 3772, the number of attributes, which varies from nine to 31, and the natural class distribution (P%), which indicates the percentage of the positive instances from the total instances of each data-set. The results vary from 6.1% for the extremely imbalanced data-set *Sick* to 45% for the almost balanced data-sets *Heart-c* and $StatlogHeart$.

For the data-set selection, we first select the *Breastc* data-set which has ten attributes and 286 instances, in which the proportion of the minority class is 29%; next, we select four moderately imbalanced data-sets, $WDBC$, *Heart-h*, *Diabetes* and $WBreastc$, in which the proportions of the minority class are 37%, 36%, 34% and 34%, respectively; then, we select an extremely imbalanced *Sick* data-set, which has 30 attributes and 3772 instances, in which the proportion of the minority class is 6%; finally, we select two almost balanced data-sets, *Heart-c* and $StatlogHeart$ data-sets, in which the proportions of the minority class are about 45%.

Table 2.2: Experimental data-set characteristics

Data-Sets		Information Data		Class Data	Data-Sets		Information Data		Class Data
ID	Name	Attributes	Instances	Classes	ID	Name	Attributes	Instances	Classes.
1	Abalone	9	4177	29	25	Lymph	19	148	4
2	Anneal	39	989	6	26	Monk1	7	432	2
3	Audiology	70	226	24	27	Monk2	7	432	2
4	Auto-mpg	8	398	3	28	Monk3	7	432	2
5	Balance	5	625	3	29	Mushroom	23	8124	2
6	Breast	10	286	2	30	Pima	9	768	2
7	Bupa	7	345	2	31	Segment	20	2310	7
8	Car	7	1728	4	32	Sick	30	3772	2
9	Cmc	10	1473	3	33	Sonar	61	208	2
10	Colic	23	368	2	34	Soybeen	36	683	19
11	Crx	16	690	2	35	Spambase	58	4601	2
12	Crx-	21	1000	2	36	Splice	61	3190	3
13	Diabetes	9	768	2	37	StatlogHeart	14	270	2
14	Ecoli	8	336	8	38	Ta	6	151	3
15	Glass	10	214	7	39	Tic-tac-toe	10	958	2
16	Hayes	5	132	3	40	Tumor	18	339	22
17	Heart-c	14	303	5	41	Vehicle	19	846	4
18	Heart-h	14	294	5	42	Vowel	14	990	11
19	Ionosphere	35	351	2	43	Waveform	41	5000	3
20	Iris	5	150	3	44	WBreastc	10	699	2
21	Kr-vs-kp	37	3196	2	45	Wdbc	31	569	2
22	Labor	17	57	2	46	Wine	14	178	3
23	Led	25	1000	10	47	Yeast	9	1484	10
24	Letter	17	20000	26	48	Zoo	17	101	7

Table 2.3: Imbalanced data-sets

Data-sets		Information Data		class data	
Index	Name	Attributes	Instances	Frequency	Classes
1	Breastc	10	286	201, 85	2
2	Bupa	7	345	145, 200	2
3	Crx	16	690	307,383	2
4	Crx-g	21	1000	700,300	2
5	Diabetes	9	768	500, 268	2
6	Ionosphere	35	351	126,225	2
7	Kr-vs-kp	37	3196	1669,1527	2
8	Labour	17	57	20,37	2
9	StatlogHeart	14	270	120, 150	2
10	Sick	30	3772	3541, 231	2
11	Sonar	61	208	97,111	2
12	Tic-tac-toe	10	958	626,332	2
13	WBreastc	10	699	458,241	2
14	WDBC	31	569	212,357	2

Table 2.4: Medical data-sets

data-sets		information data		class data		
index	name	attributes	instances	frequency	$P\%$	classes
1	Breastc	10	286	201, 85	29%	2
2	Diabetes	9	768	500, 268	34%	2
3	Heart-c	14	303	165,138	45%	2
4	Heart-h	14	294	188,106	36%	2
5	StatlogHeart	14	270	120,150	44%	2
6	Sick	30	3772	3541,231	6%	2
7	WBreastc	10	699	458,241	34%	2
8	WDBC	31	569	212,357	37%	2

Chapter 3

An Effective Approach for Imbalanced Classification: UBagging

This chapter proposes a novel supervised ensemble learning approach, unevenly balanced bagging (UBagging) for solving extremely imbalanced problems. Learning from imbalanced data is an important problem in data mining research, and one which occurs widely in many real world applications, e.g., cancer detection on an imbalanced medical data-set. Much research has addressed the problem of imbalanced data by using over-sampling and/or under-sampling methods to generate equally balanced training sets to improve the performance of the prediction models, but it is unclear what ratio of class distribution is best for training a prediction model. Bagging is one of the most popular and effective ensemble learning methods for improving the performance of prediction models; however, there is a major drawback on extremely imbalanced data-sets. It is unclear under which conditions standard bagging is outperformed by other sampling schemes in terms of imbalanced classification. These issues

motivate us to propose a novel approach, called unevenly balanced bagging (UBagging), which generates a set of moderately unevenly balanced bootstrap samples of training sub-sets to increase the diversity of the individual classifiers in the ensemble to boost the performance of the final prediction model for imbalanced binary classification. In comparing the performance of four prediction models based on decision trees $C4.5(J48)$ as base learning algorithms, our experimental results demonstrate that UBagging is effective and statistically significantly superior to single learner $J48$ (Single$J48$), standard bagging (SBagging), and equally balanced bagging (BBagging) on 32 imbalanced data-sets.

The chapter is organized as follows. Section 3.1 gives an introduction. Section 3.2 outlines the designed UBagging algorithm. Section 3.3 presents related work. Section 3.4 provides the experimental setting. Section 3.5 discusses the experimental results to compare the performance of the four prediction models, Single$J48$, SBagging, BBagging, and UBagging based on two evaluation metrics, F_{value} and G_{mean}. Section 3.6 concludes this chapter.

3.1 Introduction

Imbalanced class distribution (Weiss & Provost 2001) refers to the situation in which the numbers of training samples are unevenly distributed among different classes and the costs of misclassifying an instance in different classes are different. The imbalanced class distribution problem is one of the top ten challenging problems in data mining research (Yang & Wu 2006), and one which appears in a large number of real world applications in various domains, such as fraud detection (Phua, Alahakoon & Lee 2004), RNA gene prediction (Meyer 2007), and biomedical data prediction (Mazurowski, Habas, Zurada, Lo, Baker & Tourassi 2008). In medical cancer detection, for example, which is a typical imbalanced binary classification task, patients with cancer are

considered to be a minority class, generally forming a small proportion of the whole population in an imbalanced medical data-set; however the cost of misclassifying such samples in the interesting minority class as a normal patient is much higher than the cost of misclassifying normal samples as a patient with cancer in the other class. The minority class is the more interesting class for data miners, and it is essential that the performance of the prediction model is highly accurate in predicting the minority class.

Bagging (Breiman 1996*a*), known as standard bagging (SBagging) was proposed in 1996. Because it is one of the most popular and effective ensemble learning methods for improving the performance of prediction models, it has been popularly used in a wide range of real world applications, such as micro-array expression (Hothorn, Lausen, Benner & Radespiel-Tröger 2004), natural language processing (Wang, Zhou, Qiu, Zhang & Huang 2010), and inductive logic programming (De Castro Dutra, Page, Santos Costa & Shavlik 2003). Many empirical studies (Breiman 1996*a*, Quinlan 1996, Opitz & Maclin 1999, Bauer & Kohavi 1999, Dietterich 2000*b*) have demonstrated that it achieves better performance than a single learner if the base learners are unstable (Breiman 1996*a*). However, in an extremely imbalanced situation, SBagging performs poorly, especially in rendering poor predictions of the minority class. This is because SBagging is similar to other traditional algorithms in attempting to achieve high classification accuracy and it tends to classify all instances as negative. This is the major drawback of using SBagging to deal with an imbalanced data-set.

Sampling techniques are considered to be an effective way to tackle the imbalanced class distribution problem. Goebel stated in his thesis there must be situations in which standard bagging is outperformed by other sampling schemes in terms of predictive performance, even though bagging is the most popular sampling scheme of an ensemble method using equally

weighted classifier votes (Goebel 2004). Indeed, in an imbalanced situation, it is unclear whether SBagging can be outperformed by other sampling schemes. These issues motivate us to propose a new sampling scheme, unevenly balanced bagging (UBagging) for outperforming the SBagging prediction models on imbalanced data-sets.

It was empirically demonstrated that bagging can be outperformed frequently by simple variations in sampling, such as half-sized bags and double-sized bags (Goebel 2004), but this work did not look at imbalanced data-sets, and used the same bag size throughout the ensemble. At present, in existing bagging-based sampling schemes for imbalanced data, most research has focused on using sampling methods to provide a set of equally balanced or roughly balanced training sub-sets for training classifiers to improve the performance of the prediction models for the imbalanced classification task (Li 2007, Hido, Kashima & Takahashi 2009). To our knowledge, nobody has used a set of unevenly balanced training subsets with different bag sizes and varying ratios of class distribution in the ensemble as a sampling scheme to try to outperform SBagging for imbalanced data.

This chapter proposes a new sampling scheme to generate a set of unevenly balanced bootstrap samples to form subsets for training the prediction model. In each bag, the positive examples P_i are randomly drawn with replacement from the entire set of positive examples P using a uniform distribution, whereas the negative examples N_i are randomly selected with replacement from the original training set again with a uniform distribution, but a different size for each bag, with $|N_i|$ varying from $\frac{1}{2} * |P|$ to $2 * |P|$ by $0.05 * |P|$ increments. Thus each ensemble contains 31 bags, varying in size from $1.5 * |P|$, $1.55 * |P|$, ..., $3 * |P|$.

In comparing the performance of four prediction models all based on the unstable learner, induction of decision trees C4:5 ($J48$) (Quinlan 1986) as a

base learner, which is one of the top ten learning algorithms in data mining (Wu, Kumar, Ross Quinlan, Ghosh, Yang, Motoda, McLachlan, Ng, Liu, Yu et al. 2008), our experimental results demonstrate that our proposed algorithm, UBagging, is effective and statistically significantly superior to single learner $J48$ (Single$J48$), SBagging, and equally balanced bagging (BBagging) at a 95% confidence interval on 32 imbalanced data-sets.

The key contributions of this chapter are as follows. (1) a new sampling scheme, the UBagging algorithm, is presented in Section 2. (2) Empirical investigation and statistical analysis of the performance of the four prediction models, Single$J48$, SBagging, BBagging and UBagging are comprehensively performed. (3) Our UBagging approach is demonstrated to be statistically significantly superior to the other three prediction models, Single$J48$, SBagging and BBagging, and is applicable to both extremely imbalanced and almost balanced data-sets.

3.2 The UBagging Algorithm

Algorithm 2 outlines the algorithm of our new approach, UBagging. Our designed algorithm maintains the philosophy of bagging, but is very different from previous approaches for imbalanced classification. In each sub-set of the training set, the positive instances are randomly selected with replacement from the entire positive class and evenly distributed, where the number of positive instances $|P_i|$ have the same size as the entire positive class, $|P|$; the negative instances are randomly selected from the negative class of the original training data with replacement, where the number of negative instances $|N_i|$ is incrementally increased by 5% of $|P|$ from $\frac{1}{2} * |P|$ to $2 * |P|$, which results in $k = 31$ (the number of training subsets in the ensemble). As a result, the size and class distribution of sub-sets are different in each bag in the ensemble.

For the proposed prediction model, a set of classifiers is built based on the unevenly balanced set of bootstrap samples; each new instance is classified by a set of classifiers, and the final prediction is made by using majority voting to aggregate the decisions of the set of classifiers.

Algorithm 2: Unevenly Balanced Bagging

Input:

D, original training set, containing $|P|$ positive
and $|N|$ negative instances;
a learning scheme (e.g. $J48$, decision trees algorithm);

Output: A composite model, C^*.

Method:

while $|N_i| < 2 * |P|$ **do**

Create unevenly balanced bootstrap samples of
size $|D_i|$ sub-sets, $D_i = P_i + N_i$ where
P_i and N_i are randomly drawn with replacement
from P and N, respectively:

$|P_i| = |P|$ and;
$|N_i| = (0.5 + 0.05 * i) * |P|$;

Train each base classifier model C_i from D_i;

end

To use the composite model, C^* for a test set T on an instance x where its true class label is y:

$$C^*(x) = \arg\max_y \sum_i \delta\left(C_i(x) = y\right)$$

Delta function $\delta(\cdot) = 1$ if argument is true, else 0.

3.3 Related Work

Ensemble learning methods are considered to be an active research area for solving important imbalanced class distribution problems. More and more research (Chawla et al. 2003, Guo & Viktor 2004, Hido et al. 2009, Li 2007) has focused on ensemble learning methods to solve the imbalance problem. Ensemble learning is considered to be a powerful technique for boosting the performance of base learners by combining a set of classifiers in the ensemble. Bagging (Breiman 1996*a*) and boosting (Freund & Schapire 1996) are the most popular representatives of the ensemble learning methods.

Bagging (Breiman 1996*a*), called SBagging, also known as "bootstrap aggregating", uses sampling and voting techniques to boost the performance of prediction models. A set of bootstrap samples (D_1, $D_2...D_k$) randomly selected from the original training set D with replacement, forms different sub-sets of training data, where the sample size of the sub-sets of the training data is the same as the sample size of the original training data; and a set of classifiers (C_1, $C_2...C_k$) trained from the set of bootstrap samples (D_1, $D_2...D_k$) forms an ensemble prediction model. Each new instance is predicted by the set of classifiers (C_1, $C_2...C_k$) and a final prediction is made by taking a majority vote of the individual classifiers.

Li has proposed bagging ensemble variation (BEV) (Li 2007) to solve imbalanced problems, where by the number of negative instances is divided into n disjoint sets, where the size of the disjoint sets is the same as the size of the positive instances in the original training set, and the N training sub-sets are formed by uniting one of the disjoint sets and all the positive instances of the original training set. Thus the bag size and the class distribution of the training sub-sets are the same in each bag, which is called Balanced Bagging (BBagging) in this study.

Goebel's thesis (Goebel 2004) investigated different sized bags for bagging, such as bags of half, unit and double size. He found that some data-sets did better with larger bags and some data-sets did better with smaller bags, but these approaches do not specifically look at imbalanced classifiers and use the same subset size in each ensemble - two significant differences from our approach.

Our previous study (Liang et al. 2011*b*) also considered varying the number of positive examples from $0.1 * (|P| + |N|)$ to $0.9 * (|P| + |N|)$, but again in each ensemble all bags were of the same size, since the number of negative examples was chosen to keep the bag size constant.

Our approach is very different from previous approaches because it generates unevenly balanced bootstrap samples with varying class distributions and unequal subset sizes in the ensemble.

3.4 Experimental Setup

A Java platform is used to investigate the performance of the prediction models. A 10-trial 10-fold cross-validation evaluation is employed for this study (i.e. each 10-fold cross-validation was repeated 10 times and the results averaged). The implementation of the decision trees, $C4.5$ (Quinlan 1986), $J48$ with default parameters from WEKA (Witten & Frank 2005), is used as the base learner. In each ensemble, 31 individual classifiers are used. For example, there are 1000 instances in the *German* data-set, which include 300 positive instances and 700 negative instances; for each subset of the training set, 300 positive instances are randomly selected with replacement from the entire positive class; negative instances are randomly drawn with replacement from the original training set with sizes of 150, 165,180, 195, ..., 600, respectively; therefore, the size of each subset of the training set is 450, 465, 480, 495, ..., 900 in the ensemble.

Table 3.1: Imbalanced data-sets (ordered by $\%P$)

Data-sets		Data Information		Class Information		
Index	Name	Instances	Attributes	%P	Positive Class	Class
1	Abalone19	4177	8	0.80%	19	2
2	LetterA	20000	16	3.90%	A	2
3	Car3	1728	6	4.00%	good	2
4	Sick	3772	29	6.10%	Sick	2
5	FlagWhite	194	29	8.80%	white	2
6	Ecoli-4	336	7	10.40%	iMU	2
7	Yeast-ME3	1484	8	11.00%	ME3	2
8	Glass7	214	9	13.60%	headlamps	2
9	Segment1	2310	19	14.30%	brickface	2
10	Hepatitis	155	19	20.60%	DIE	2
11	VehicleVan	846	18	23.50%	van	2
12	Splice1	3190	61	24.00%	EI	2
13	Haberman	306	3	26.50%	2	2
14	Lungcancer	32	57	28.10%	1	2
15	Breastc	286	9	29.70%	recurrence	2
16	Credit-g	1000	20	30.00%	bad	2
17	German	1000	20	30.00%	2	2
18	Breastw	699	9	34.50%	malignant	2
19	Tic-tac-toe	958	9	34.70%	negative	2
20	Diabetes	768	8	34.90%	test_positive	2
21	Labor	57	16	35.10%	bad	2
22	Ionosphere	351	34	35.90%	b	2
23	Heart-h	294	13	36.10%	>50_1	2
24	Colic	368	22	37.00%	no	2
25	WDBC	569	30	37.30%	1	2
26	Spambase	4601	57	39.40%	1	2
27	Liver-dis	345	6	42.00%	1	2
28	Heart-stat	270	13	44.40%	present	2
29	Credit-a	690	15	44.50%	0	2
30	Heart-c	303	13	45.50%	>50_1	2
31	Sonar	208	60	46.60%	Rock	2
32	Kr-vs-kp	3196	36	47.80%	nwon	2

3.4.1 Data-sets

Table 3.1 shows the 32 imbalanced data-sets selected for this study from UCI (Merz & Murphy 2006). The first two columns present the index and the name of each data-set, the third and fourth columns indicate the number of instances and attributes of the data information, and the final three columns show the class information: the proportion of the positive examples to the whole data-set, the name of the positive class, and the number of classes. The data-sets have different criteria, e.g., the number of instances from 57 to 20000, the number of attributes from 6 to 61, and the proportion of the positive examples varying from 0.8% to 47.8%.

3.5 Experimental Results and Analysis

This section presents the comparison of the performance of four prediction models based on two evaluation metrics, F_{value} and G_{mean}.

Table 3.2 presents a comparison of the performance of the four prediction models using the average value and average rank based on two evaluation metrics, F_{value} and G_{mean}. The results of the average ranks of F_{value} and G_{mean} are the output of the Friedman test, and the Null Hypothesis of this test is rejected, so a post-hoc Nemenyi test is required to calculate the "critical difference" to determine where one prediction model is significantly different from another (Demšar 2006). In Table 3.2, the first two columns indicate the index and name of the data-sets. The final four columns present the performance of the four prediction models. The last two rows present the summary of the experimental results, which respectively indicate the average of evaluation metrics with standard deviation (STD) and the average rank of evaluation metrics with "critical difference" of the Nemenyi test over the 32 data-sets.

Table 3.2: Comparison of the performance of four prediction models based on F_{value} and G_{mean}

Data-sets		F_{value}				G_{mean}			
Index	Name	SingleJ48	Sbagging	Bbagging	Ubagging	SingleJ48	Sbagging	Bbagging	Ubagging
1	Abalone19	NA	NA	0.051	**0.058**	0	0	0.836	**0.854**
2	LetterA	0.954	**0.964**	0.879	0.902	0.971	0.972	0.994	**0.996**
3	Car3a	NA	0.235	0.378	**0.391**	0	0.378	0.929	**0.933**
4	Sick	0.89	**0.897**	0.798	0.822	0.931	0.926	0.976	**0.979**
5	FlagWhite	NA	NA	0.272	**0.348**	0	0	0.697	**0.798**
6	Ecoli4	0.579	0.634	0.624	**0.64**	0.707	0.734	0.917	**0.923**
7	Yeast-ME3	0.763	0.768	0.781	**0.792**	0.867	0.866	0.952	**0.955**
8	Glass7	0.837	0.844	0.859	**0.872**	0.897	0.905	0.959	**0.96**
9	Segment1	0.976	0.976	0.974	**0.977**	0.983	0.983	**0.995**	**0.995**
10	Hepatitis	0.366	0.489	0.705	**0.722**	0.53	0.619	0.88	**0.887**
11	VehicleVan	0.874	0.909	0.903	**0.913**	0.919	0.944	0.966	**0.97**
12	Splice1	**0.942**	0.937	0.928	0.932	0.969	0.961	0.972	**0.974**
13	Haerman	0.261	0.326	0.612	**0.62**	0.418	0.472	0.743	**0.752**
14	Lungcancer	0.549	0.294	0.727	**0.757**	0.655	0.439	0.836	**0.861**
15	Breastc	0.364	0.376	0.599	**0.613**	0.488	0.5	0.667	**0.693**
16	Credit-g	0.461	0.505	0.678	**0.702**	0.587	0.618	0.773	**0.799**
17	German	0.454	0.502	0.684	**0.709**	0.581	0.615	0.779	**0.807**
18	Breast-w	0.924	0.944	0.958	**0.97**	0.944	0.958	0.976	**0.983**
19	Tic-tac-toe	0.769	**0.906**	0.848	0.88	0.816	0.915	0.899	**0.925**
20	Diabetes	0.621	0.637	0.792	**0.811**	0.702	0.714	0.848	**0.866**
21	Labor	0.671	0.797	0.85	**0.865**	0.738	0.834	0.898	**0.91**
22	Ionosphere	0.858	0.893	0.942	**0.955**	0.883	0.904	0.965	**0.972**
23	Heart-h	0.693	0.71	0.804	**0.817**	0.753	0.769	0.853	**0.864**
24	colic	0.783	0.79	0.818	**0.837**	0.818	0.822	0.86	**0.877**
25	WDBC	0.912	0.943	0.968	**0.972**	0.928	0.953	0.979	**0.982**
26	Spambase	0.908	0.93	0.95	**0.956**	0.924	0.94	0.964	**0.968**
27	Liver-dis	0.548	0.628	0.798	**0.817**	0.619	0.684	0.797	**0.823**
28	Heart-sta	0.755	0.786	0.884	**0.895**	0.781	0.808	0.889	**0.902**
29	Credit-a	0.833	0.849	0.866	**0.876**	0.849	0.863	0.873	**0.884**
30	Heart-c	0.733	0.764	0.869	**0.881**	0.756	0.785	0.866	**0.881**
31	Sonar	0.716	0.764	0.899	**0.901**	0.733	0.785	0.896	**0.898**
32	Kr-vs-kp	0.994	0.994	0.994	0.995	0.994	0.994	0.994	**0.995**
Average		0.656	0.687	0.772	**0.787**	0.711	0.739	0.888	**0.902**
STD		0.284	0.276	0.207	**0.202**	0.274	0.254	0.087	**0.076**
Mean Rank		3.64	2.77	2.37	**1.22**	3.8	3.14	2.05	**1.02**
"critical difference"		0.829				0.829			

The experimental results indicate that our proposed new sampling scheme, UBagging, performs the best on average with the smallest STD and average rank based on both evaluation metrics, F_{value} and G_{mean}, across all data-sets (results in bold indicate the best overall performance out of the four classifiers). Table 3.2 shows that UBagging achieves the highest average of (0.787) with the lowest average rank (1.22) for F_{value}, and the highest average of (0.902) with the lowest average rank (1.02) for G_{mean}. Therefore, the UBagging prediction model is better than the other three prediction models: Single$J48$, SBagging, and BBagging.

In addition, Table 3.2 indicates that UBagging achieves better G_{mean}s than the other three prediction models on all the data-sets, except for BBagging on the $Segment1$ data-set (where they perform equally well); in Table 3.2 on the other hand, SBagging achieves the best F_{value} compared to the other three prediction models on only 3 out of 32 data-sets, $LetterA$, $Sick$ and $Tic-tac-toe$, while Single$J48$ achieves the best F_{value} on 1 out of 32 data-sets, $Splice1$. We note that $lettersA$, $Sick$, and $Segment1$ are three large, highly imbalanced data-sets. We further investigate the performance of the four prediction models on four large imbalanced data-sets, $LetterA$, $Sick$, $Splice1$, and $Tic-tac-toe$. We observe that Single$J48$ and SBagging achieve a very high TNR and a very low TPR on those data-sets. This is why Single$J48$ and SBagging achieve a high F_{value} with lower G_{mean} on those larger data-sets.

Table 3.2 shows that on three highly imbalanced data-sets, $Abalone19$, $Car3$ and $FlagWhite$, Single$J48$ and SBagging get 0 on G_{mean} - the reason is that their TPR =0, which is also why their F_{value} is not available (NA).

For the most balanced data-set, *Kr-vs-kp*, the four methods produce fairly similar results. This is not so surprising, although the results for the other fairly balanced data-sets (e.g. *Heart-c* and *Sonar*) show more differences, indicating that UBagging performs better under these nearly

balanced conditions too. It can also be seen that for the largest data set, *LetterA*, all four classifiers perform well, even though there are only 3.9% positive examples, presumably because there are enough positive examples in such a large data set for them to have sufficient influence on the model.

Figure 3.1 presents a comparison of the performance of the prediction models with the Nemenyi test, where the x-axis indicates the average rank of F_{value}, the y-axis indicates the ranking order of the four prediction models, and the horizontal bars indicate the "critical difference". If the horizontal bars between prediction models overlap, it means there is no statistically significant difference between the prediction models at a 95% confidence interval. The results indicate that based on F_{value}, our proposed UBagging is statistically superior to the other three prediction models, BBagging, SBagging and Single$J48$; BBaging and SBagging are statistically superior to Single$J48$; however, there is no statistically significant difference between BBagging and SBagging.

Figure 3.2 presents a comparison of the performance of the four prediction models with the Nemenyi test based on the evaluation metric, G_{mean}, where the x-axis indicates the average rank of G_{mean}, the y-axis indicates the ranking order of G_{mean} for the four prediction models, and the horizontal bars indicate the "critical difference". If the horizontal bars between prediction models do not overlap, it means there is a statistically significant difference between the prediction models at a 95% confidence interval. The results indicate that based on G_{mean}, our proposed UBagging is statistically superior to the other three prediction models, BBagging, SBagging and Single$J48$; BBagging is superior to the other two prediction models, SBagging and Single$J48$; however, there is no statistically significant difference between SBagging and Single$J48$.

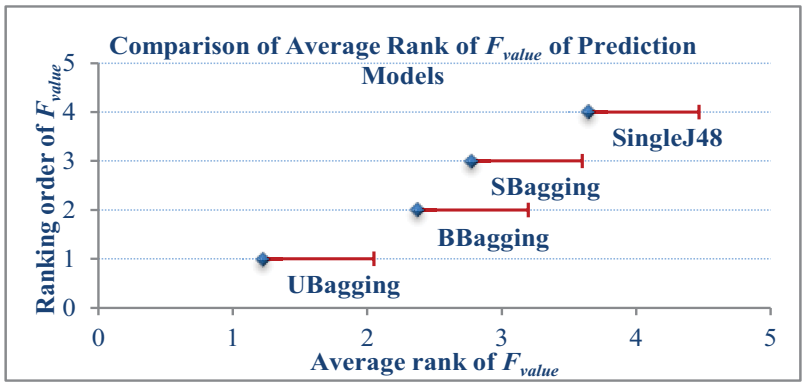

Figure 3.1: Comparison of average rank of F_{value} of the performance of four prediction models with the Nemenyi test, where the x-axis indicates the average rank of F_{value}, the y-axis indicates the ranking order of the four prediction models, and the vertical bars indicate the "critical difference".

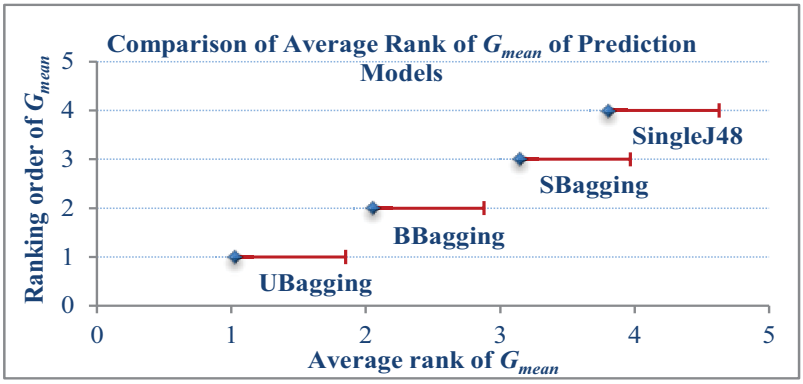

Figure 3.2: Comparison of average rank of G_{mean} of the performance of four prediction models with the Nemenyi test, where the x-axis indicates the average rank of G_{mean}, the y-axis indicates the ranking order of the four prediction models, and the horizontal bars indicate the "critical difference".

3.6 Conclusion

This chapter proposes a new approach, called UBagging, to boost the performance of the prediction model for solving extremely imbalanced problems. This approach generates unevenly balanced bootstraps samples with unequal bag sizes and varying ratios of class distribution, which is different from previous approaches, which to the best of our knowledge all use identical or near-identical sized bags to improve the performance of the bagging predictor to solve imbalanced classification problems. We compared the performance of the new approach, UBagging, with three other prediction models, Single$J48$, SBagging, and BBagging, based on two evaluation metrics, F_{value} and G_{mean}. The experimental results demonstrate that our new approach is statistically significantly superior to the other three prediction models at 95% confidence interval on two evaluation metrics over 32 imbalanced data-sets. We observe that this new approach, UBagging, performs well on both extremely imbalanced and almost balanced binary classification tasks. We have examined our approach using an unstable base learner, $J48$. We believe the success of these results will also apply to other base learners (initial experiments with an SVM indicate support for this hypothesis). Our future research will focus on the performance of UBagging with other base learning algorithms on imbalanced data-sets. There are also other aspects that could be investigated too, such as altering the sizes over which N_i varies, and the increment between successive bags – here we use an increment of 5% of $|P|$ having found this to be better than 10%, but a more thorough investigation of these aspects could be made.

Chapter 4

An Empirical Study of Bagging Predictors with Different Learning Algorithms

This chapter investigates the performance of ensemble learning systems with respect to different learning algorithms in terms of learning from natural class distribution. Bagging is a simple yet effectively designed ensemble learning method for generating a set of bootstrap samples with replacement for training a set of classifiers and aggregating a set of individual classifiers to improve the performance of the resulting prediction model. Despite the popular usage of bagging in many real-world applications, existing research is mainly concerned with studying unstable learners as the key to ensuring the performance gain of a bagging predictor, with many key factors remaining unclear. We assert that both stability and robustness are key requirements for building a high performance bagging predictor. In addition, the definitions of robustness and stability are formally defined. A novel approach, two-dimensional robustness and stability decomposition, is proposed to rank the base learners into different

categories to investigate the performance of bagging predictors with respect to different learning algorithms.

In this chapter, we carry out comprehensive empirical studies to evaluate the performance of bagging predictors by using 12 different learning algorithms and 48 benchmark data-sets. The experimental results demonstrate that bagging is influenced by the combination of robustness and instability, and indicate that robustness is important for bagging to achieve a highly accurate prediction model. In addition, our studies demonstrate that bagging is statistically significantly superior to most single learners, except for KNN and Naïve Bayes (NB). Multi-layer perceptron (MLP), Naïve Bayes Trees (NBTree), and PART are the learning algorithms with the best bagging performance.

The chapter is organized as follows. Section 4.1 gives an introduction. Section 4.2 outlines the designed framework. Section 4.3 presents base learner characterization. Section 4.4 presents the experimental setting. Section 4.5 analyzes the experimental results. Section 4.6 concludes the chapter and discusses future research directions.

4.1 Introduction

The aim of ensemble learning is to improve the performance of the prediction model by combining a set of multiple base learners. The high accuracy and diversity of ensemble learning have captured the interest of the data mining and machine learning community for about two decades. Techniques of effective ensemble methods have been investigated theoretically and empirically by many previous researchers (Breiman 1996*a*, Freund & Schapire 1996, Kittler 1998, Schapire 1990). Bagging (Breiman 1996*a*) is one of the most popular and effective parallel ensemble learning methods. Bagging uses a set of bootstrap samples to

train a set of classifiers, and the prediction of a new example in the test set is made by the majority votes of a set of trained classifiers.

As a result of its simple yet effective design, bagging has been popularly used in many real-world applications (West et al. 2005, Kim & Kang 2010, Lopes et al. 2008, Tu et al. 2009*b*, Tu, Shin & Shin 2009*a*, Xu, Zuo, Zhang & He 2010, Hothorn et al. 2004, Hu et al. 2006). Bagging is widely accepted as a variance-reduction technique, so it is mostly applied to unstable, high variance algorithms (Tuv 2006). Breiman heuristically defined instability as an unstable classifier, which means that a small change in the training data can lead to large change in the resulting prediction model (Breiman 1996*a*). Existing research is mainly concerned with studying unstable learners as the key to improving the performance of a bagging predictor, with many key factors remaining unclear. This study asserts that both stability and robustness are key requirements to ensure high performance for building a bagging predictor.

Previous empirical studies (Breiman 1996*a*, Quinlan 1996, Opitz & Maclin 1999, Bauer & Kohavi 1999, Dietterich 2000*b*, Breiman 1996*b*) have demonstrated that bagging is often more accurate than any of the individual learners in the ensemble if the base learners are unstable. However, most previous experimental research only focuses on one or two base learners, or only considers one factor: instability. There are implicit measures to group the base learners into different categories to analyze bagging; moreover, it is not clear from the literature which bagging predictor performs best on a application.

Existing studies have demonstrated the effectiveness of the bagging predictor, but a comprehensive study of bagging predictors with respect to different learning algorithms has not been undertaken. Given a large body of learning algorithms, existing research is limited in its ability to answer practical questions such as (1) when should we expect a bagging predictor

to outperform a single learner? (2) which learning algorithms are expected to achieve the maximum accuracy gain? (3) does bagging improve the performance of low variance but highly accurate base learners? and (4) which bagging predictor performs best on randomly selected databases from an application point of view? Answering these questions poses the following research challenges: (1) how to rank/group the base learners into different categories, and (2) how to conduct a fair and rigorous study to evaluate multiple algorithms over multiple data-sets (Demšar 2006).

This chapter presents a comprehensive study on bagging predictors which uses 12 learning algorithms and 48 benchmark data-sets. The main contributions of this chapter are as follows:

1. A novel approach is proposed, asserting that both stability and robustness are key requirements for building a high performance bagging predictor. Definitions of robustness and stability are formally defined to investigate the performance of bagging predictors with respect to different learning algorithms on 48 data-sets.

2. A novel approach, the two-dimensional decomposition of robustness and stability, is proposed to rank based learners into different categories: strong, weak, stable and unstable learners.

3. Statistical tests are used to compare the classifiers to draw valid conclusions.

4.2 Designed Framework

Figure 4.1 presents the designed framework, which is divided into three tasks: (1) two important factors, robustness and stability are defined; a novel two-dimensional robustness and stability decomposition is proposed to categorize base learners into different categories; (2)the use of Friedman test with post-

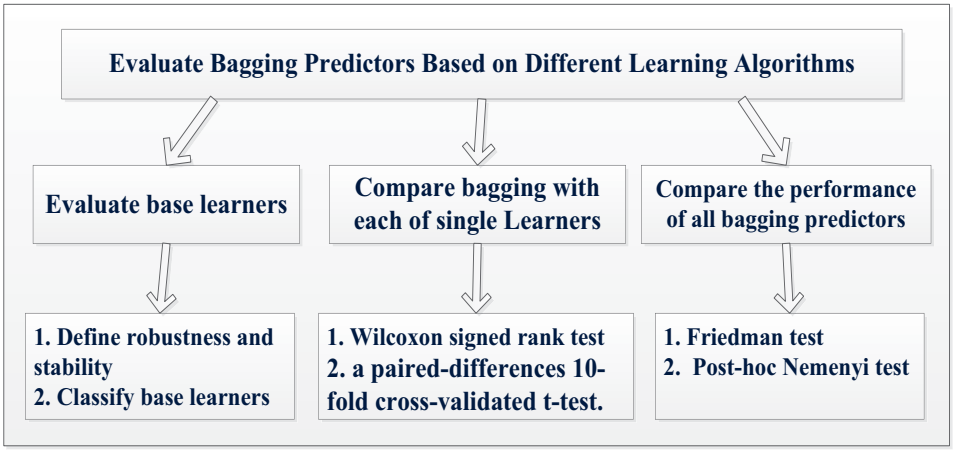

Figure 4.1: Designed framework

hoc Nemenyi test to compare multiple learners (e.g., comparison of bagging predictors with one another) to determine which bagging predictors have the best performance; and (3) the comparison of bagging with single learners, two statistical tests are used: (a) using Wilcoxon signed-rank test to compare two learners (e.g., bagging SVM and the single learner SVM) to determine when bagging will outperform a single learner, and (b) using a paired-difference cross-validated t-test to determine which bagging predictor on average has the largest performance gain across all the benchmark data-sets.

4.3 Base Learner Characterization

To investigate the bagging predictors with respect to different learning algorithms, a two-dimensional robustness and stability decomposition is proposed to characterize base learners based on estimated error rate, and variance as a performance measure to assess the base learners.

Definition 1 *Robustness refers to the ranking of the average performance of a base learner among a set of learners. For example, if we assume the*

estimated error rate is a performance measure, we rank all base learners according to their estimated error rate performance on each data-set to obtain the average rank over all benchmark data-sets; the normalized ranking order of the estimated error rate is then used to capture the robustness of a learner, with a smaller ranking number denoting a more robust learner.

Definition 2 *Stability refers to the ranking of the variance of a base learner in a set of learners. For example, if we assume the variance of the error rate is a performance measure, we rank all base learners according to their variance on each data-set to obtain the average rank over all benchmark data-sets; the normalized ranking order of variance is then used to capture a learner's stability, with a smaller ranking number denoting a more stable learner.*

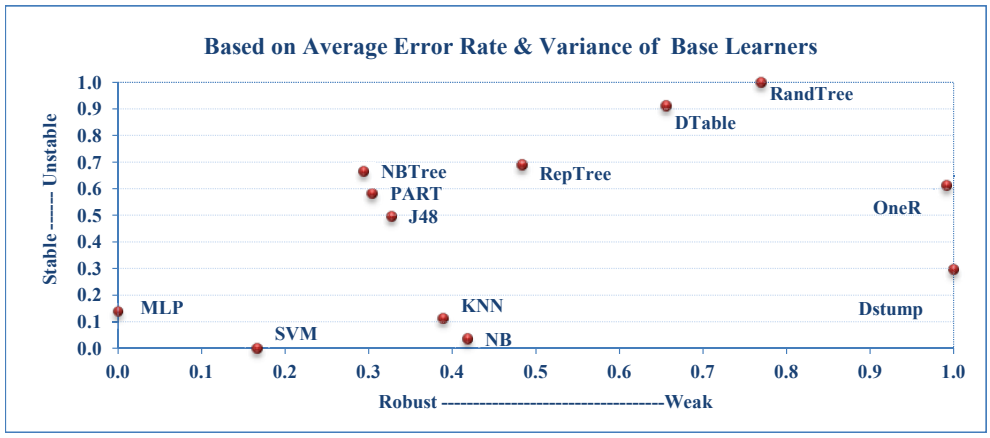

Figure 4.2: Two-dimensional robustness and stability decomposition of the base learners based on estimated error rate and variance, where the x-axis denotes the robustness of the base learners from robust to weak, and the y-axis denotes the stability of the base learners from stable to unstable.

Figure 4.2 illustrates two-dimensional Robustness and Stability decomposition in assessing base learners based on estimated error rate and variance as performance measures. There are three steps to plotting Figure 4.2 as follows:

1. All base learners are ranked based on their estimated error rate and variance. The base learner with the best performance (the lowest error rate and variance) is ranked as 1, while the worst performance (the highest error rate and variance) is ranked as 12.

2. The average ranks of their estimated error rate and variance are calculated.

3. The normalized ascending order of average ranks of estimated error rate and variance is used to create the two-dimensional plot to represent the robustness and stability of base learners, respectively.

In Figure 4.2, MLP and SVM with a smaller value of robustness denote more robust learners, while OneR and Dstump with a larger value of robustness denote more robust learners. On the other hand, NB and SVM with a smaller value of stability denote more stable learners, while RandTree and DTable with a larger value of stability denote more unstable learners. In addition, we observe that MLP and SVM, both having relatively lower variance, are similar to and have more robustness than KNN and NB respectively.

4.4 Experimental Setting

We use WEKA implementation of the 12 algorithms with default parameter settings in this empirical study (Witten & Frank 2005). To reduce uncertainty and obtain reliable experimental results, all the evaluations are

assessed under the same test conditions by using the same randomly selected bootstrap samples with replacements in each fold of 10-trial 10-fold cross-validation on each of the 48 data-sets in Table 2.2 collected from the UCI Machine Learning Repository (Merz & Murphy 2006).

An experimental study of model selection comparing the two common methods, cross-validation and bootstrap, demonstrate that ten-fold cross validation is the best method to use to select a good model from a set of classifiers (model selection) (Kohavi 1995). Therefore, we use the 10-trial 10-fold cross-validation technique for this study.

On each data-set, 20 bootstrap samples are drawn from each iteration training fold of 10-fold cross-validation with replacements. The integer k is the number of bootstrap samples. Each bootstrap sample D_k of size $|M|$ has the same number of instances as the original training set. We trained the classifiers of different base learners on the same bootstrap sample D_k of a ten-fold cross validation respectively, and tested the unseen instances on the test set T_k. For bagging, the prediction C^* is made by the majority vote of the k classifiers $(C_1, C_2, ..., C_k)$ which were previously trained from the k bootstrap samples, $D_1, D_2, ..., D_k$ in each training set; while for the single base learner the classifier is trained from the original training set. The misclassification error rate is averaged over ten-trial ten-fold cross-validation, and we therefore we predicted the measures of their performance based on the average error rate and standard deviation over ten trials. We utilized 0/1 loss to estimate the bias and variance to measure the robustness and stability of the base learners.

4.5 Experimental Analysis

Section 4.5 analyzes the experimental results as follows: (1) Subsection 4.5.1 is based on the Friedman test with the Post-hoc Nemenyi test to

compare all bagging predictors with each other to determine which group of bagging predictors performs the best and the worst, respectively; (2) Subsection 4.5.2 is based on the Wilcoxon signed-rank test to compare each pair of bagging predictors and single learners to determine which bagging predictors are statistically significantly superior to their single learners; and (3) Subsection 4.5.3 is based on a paired-difference cross-validated t-test to compare the average improvement of each pair of bagging predictors and single learners over multiple data-sets.

4.5.1 Comparison of All Bagging Predictors

The Friedman test is used for the comparison of multiple bagging predictors. First, we perform the Friedman test to compare 12 bagging predictors on each data-set, and then to obtain their mean ranks over multiple data-sets. If the Null Hypothesis is rejected, the test indicates that there is at least a difference between the mean ranks of bagging predictors, and the corresponding post-hoc Nemenyi test for the additional exploration of the differences between mean ranks provides specific information on which mean rank is significantly different from another.

Table 4.1: Mean rank of Friedman test for error rate of bagging predictors

Mean rank of *ErrorRate* of bagging predictors from Friedman Test						
Learners	B_MLP	B_NBTree	B_PART	B_J48	B_SVM	B_RandTree
Mean Rank	3.61	3.82	3.96	5.24	5.42	5.57
Learners	B_RepTree	B_NB	B_KNN	B_DTable	B_OneR	B_DStump
Mean Rank	6.59	7.47	7.56	7.56	10.38	10.81

Table 4.1 shows the statistical results of the Friedman test and the average rank of the *errorrate* performance of bagging predictors. The second and

fourth rows indicate the name of the bagging predictors, and the third and last rows indicate the mean rank of the *errorrate* performance of bagging predictors from the Friedman test results.

Figure 4.3 reports the results of the Friedman with post-hoc Nemenyi test for comparison of the mean rank of all bagging predictors over 48 datasets. The x-axis indicates the mean rank of each algorithm, while the y-axis indicates the ascending ranking order of the bagging predictors. The horizontal error bars indicate the "critical difference". If the horizontal bars between bagging predictions do not overlap, that means there is a statistically significant difference between the bagging predictions at a 95% confidence interval. We observe that the group of most robust base learners, MLP, NBTree, and PART contributes to the best bagging predictors, whereas the group of weakest learners, OneR and DStump, leads to the worst bagging predictors. There is a statistically significant difference between the two groups. As a result, one can conclude that the robustness of the base learners is an important factor for building accurate bagging predictors.

The ranking order of most robust base learners is MLP, SVM, NBTree, PART and $J48$, while the ranking order of the most robust bagging predictors is: B_MLP, B_NBTree, B_PART, B_$J48$, and B_SVM. SVM is more robust than $J48$, but B_$J48$ is superior to B_SVM according to the mean rank of bagging predictors. The possible reason is that the instability of SVM is lower than the instability of $J48$. We conclude that bagging is influenced by the combination of instability and robustness, and point out that robustness is an important factor for achieving a highly accurate prediction model.

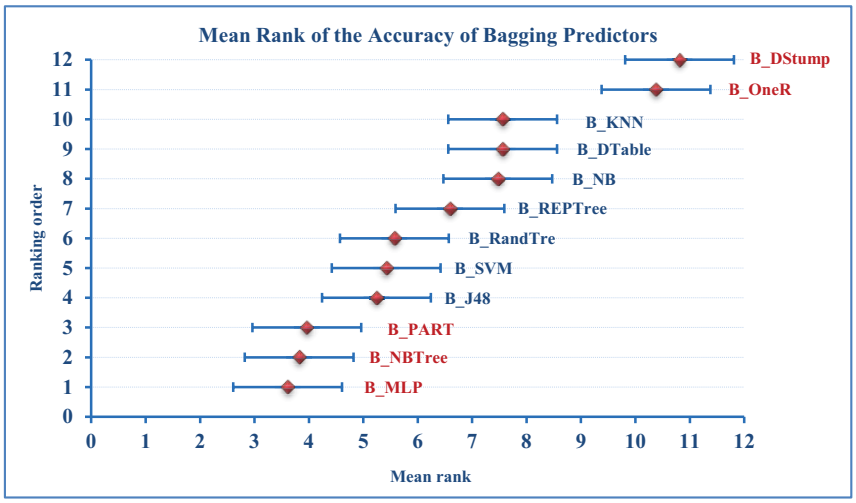

Figure 4.3: Friedman and post-hoc Nemenyi test results of comparison of all bagging predictors, where the x-axis indicates the mean rank of bagging predictors, the y-axis indicates the ranking order of the bagging predictors, and the horizontal error bars indicate the "critical difference".

Table 4.2: The results of the Wilcoxon signed-rank test to compare the estimated *ErrorRate* of bagging and single learners. The significance level is .05.

Wilcoxon signed-rank test to compare bagging and single learners						
Learners	NB	KNN	SVM	MLP	DStump	NBTree
p-values	**.555**	**.110**	.001	.000	.000	.000
Learners	DTable	OneR	J48	PART	RepTree	RandTree
p-values	.000	.000	.000	.000	.000	.000

4.5.2 Comparison of Two Learners Bagging and Single Learner

The Wilcoxon signed-rank test is considered to be safe from the statistical point of view and is more powerful than t-test when the test conditions cannot meet the assumption requirements of a parametric test. Therefore, we performed this test to determine whether there really is an improvement in performance between the two learners, bagging and single learner. The Null Hypothesis is that the median of differences between bagging and a single learner equals 0. Therefore, the Null Hypothesis of this test states that both learners perform equally well, while the alternative Null Hypothesis states that the performance difference between the two learners is significant.

Table 4.2 presents the summarized results of the Wilcoxon signed-rank test for the performance between all pair-wise combinations of the comparisons, bagging and individual single learners. If a calculated *p-value* is greater than α value, 0.05, then the *p-values* are highlighted and we accept the null hypothesis, e.g., both KNN and NB. For all other cases, the *p-value* is less than the α value, 0.05 and we reject the rest of the null hypothesis. Therefore, the Wilcoxon signed-rank test indicates that bagging performs statistically significantly better than most of the single learners, except for KNN and NB. Previous studies have concluded that KNN and NB are stable learners, so their performance in bagging predictors is not supposed to be good. It is consistent with previous research.

4.5.3 Comparison of Average Improvement of Bagging

Figure 4.4 presents the average improved performance of bagging over single learner on multiple data-sets. Each point plots the averaged difference in the performance of the two algorithms, bagging and single learner, on all benchmark data-sets. The vertical axis indicates the

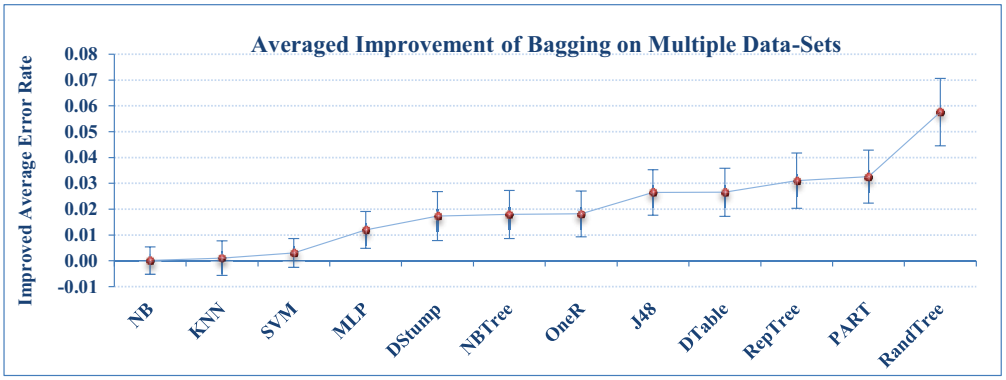

Figure 4.4: The improved accuracy between bagging predictors and individual base learners on average over multiple data-sets. The error bars present a 95% confidence interval based on the cross-validated t-test.

observed difference, where the error bars indicate the statistical significance of the observed difference. In the plot, the 12 learners are sorted in an ascending order of their averaged differences.

Figure 4.4 demonstrates that bagging RandTree gains nearly 6% improvement in Error Rate on average over 48 data-sets, while there is almost no gain for bagging NB and KNN. These findings are consistent with Breiman's theories and our experimental results of the Wolcoxon signed-rank test in Table 4.2. However, bagging MLP and SVM have better performance gain than bagging KNN and NB over 48 data-sets. According to Breiman's theories (Breiman 1996*a*), if they have similar variance with KNN and NB, respectively, they should not have a better gain than KNN and NB. A possible reason is that both MLP and SVM are stronger than KNN and NB.

4.6 Conclusions

This chapter empirically studies the performance of bagging predictors with respect to different algorithms in terms of learning from natural class distribution. It reports a comprehensive empirical study to evaluate the performance of bagging for 12 algorithms on 48 data-sets, in which first both robustness and stability were defined to investigate when bagging is influenced by different types of base learners. Base learners were then ranked into different categories by using two dimensional robustness and stability decomposition to analyze bagging predictors. Surprisingly, we found that MLP and SVM have relatively low variance with KNN and NB, while each of them is stronger than KNN and NB, respectively. However, our results demonstrate that bagging is most often statistically superior to a single learner, except for KNN and NB. In addition, we found that MLP is the strongest base learner of all 12 base learners, and that bagging MLP is the best predictive model of a total of 24 algorithms across numerous domains on 48 data-sets. Furthermore we observed that the strongest base learners such as MLP, NBTree and PART can be used to build the best bagging predictors. This finding is consistent with previous empirical studies and will provide a useful guideline for real world applications in selecting a suitable prediction model for projects. There was a distinct advantage to including different categories of base learners in this study and a new method is provided to rank those base learners into different categories, especially when two important factors are defined to analyze base learners and investigate when bagging is influenced by different types of base learners. We query why and how bagging would improve low variance algorithms, such as SVM and MLP.

Chapter 5

An Empirical Study of the Sensitivity of Bagging on Imbalanced Class Distribution

This chapter investigates the effect of varying levels of class distribution on the sensitivity of bagging. The sensitivity refers to whether there is large change in the performances of the bagging predictor when the levels of class distribution are varied; if there is a large change, it means that the bagging predictor is sensitive, whereas if there is no change or less change in the performance of the bagging predictor, it means that the bagging predictor is insensitive.

As growing numbers of real world applications involve imbalanced class distribution, learning from imbalanced class distribution is considered to be one of the most challenging issues in data mining and machine learning research. However, it is unclear which bagging predictors are sensitive when levels of class distribution are changed. The assessment of the sensitivity of a bagging predictor is based on the ranking of the changed G_{mean} (CG) performance of bagging predictors between two adjacent levels of class

distribution, and the average rank of the CG performance of bagging predictors is compared. The largest mean rank of a bagging predictor indicates that the bagging predictor is sensitive; by contrast, the lowest mean rank of a bagging predictor indicates that the bagging predictor is insensitive to changing levels of class distribution.

This chapter empirically investigates the sensitivity of bagging with respect to 12 learning algorithms at nine levels of class distribution on 14 imbalanced data-sets by using statistical and graphical methods to address the important imbalanced problem and the effect of varying levels of class distribution on bagging predictors. In addition, statistical analyses are performed to ensure the results are validated. The experimental results demonstrate that bagging MLP and bagging NB are insensitive when the levels of imbalanced class distribution vary. Furthermore, we observe that there is no statistically significant difference between bagging MLP and bagging NB, and both have statistically significant differences when compared with the remaining ten bagging predictors.

The chapter is organized as follows. Section 5.1 gives an introduction. Section 5.2 presents the outline of the designed framework and two evaluation metrics: ROC curve and G_{mean}. Section 5.3 presents the experimental results analysis which uses both statistical and graphical methods. Section 5.4 concludes the chapter.

5.1 Introduction

Imbalanced class distribution refers to the training samples that are non-uniformly distributed among classes. Typically, in a binary classification, the minority class samples are much smaller than the majority class samples; the minority class and majority class are regarded as a positive class and a negative class, respectively.

A growing number of researchers focus on solving imbalanced class distribution problems in real world applications in a variety of domains, such as credit card fraud detection, medical diagnosis, and biological data analysis (Chen & Wasikowski 2008). However, it is unclear which bagging predictors with which learning algorithms are insensitive to imbalanced class distribution, and which may perform well if they are directly applied on the imbalanced data-sets.

(Weiss & Provost 2001) evaluated the effect of class distribution on classifier learning by assessing the relationship between training class distribution and the performance of the decision trees $C4.5$ learner to draw their conclusions as to which distribution is best for training, based on two evaluation measures: error rate and Area Under the ROC curve (AUC). However, they did not evaluate which learner is sensitive when the levels of class distribution vary. Moreover, imbalanced class distribution often causes learning algorithms to perform poorly on the minority class; the mis-classification error rate cannot determine the accuracy of the minority class (He & Garcia 2009, Weiss & Provost 2001). Two evaluation measures, Receiver Operating Characteristic (ROC) Curve and Geometric mean (G_{mean}) are therefore adopted for this study.

Bagging (Breiman 1996a) utilizes bootstrap sampling and majority vote techniques to improve the performance of the prediction models. It has been applied to a variety of real world applications, such as micro-array expression (Hothorn et al. 2004), financial decision applications (West et al. 2005), and Natural Language Processing (Wang et al. 2010), but it is unclear which bagging predictors are sensitive when the levels of class distribution vary. Our previous studies investigated the performance of bagging predictors with respect to different learning algorithms (Liang et al. 2011a) and with respect to different levels of imbalanced class distribution (Liang et al. 2011b), but we did not investigate the sensitivity of bagging, so it is unclear which bagging

predictors are sensitive to varying levels of class distribution.

Investigating the sensitivity of bagging predictors presents the following research challenges: (1) how to evaluate the sensitivity of the bagging predictors between the adjacent two different levels of class distribution over multiple data-sets, and (2) how to use graphical methods to visualize the sensitivity of the bagging predictors.

Our main contribution is to conduct a comprehensive evaluation of the sensitivity of bagging predictors to understand the effect of varying levels of class distribution by using two evaluation methods: (1) statistical methods to draw validated conclusions, and (2) graphical methods to further visualize the sensitivity of bagging predictors. As a result, our research provides both graphical and statistical comparisons of the sensitivity of bagging predictors with the underlying 12 base learners when the levels of class distribution vary. The experimental results provide a useful guide for data mining practitioners to understand the sensitivity of the bagging predictors and solve imbalanced class distribution problems for their applications.

5.2 Designed Framework

Figure 5.1 represents a designed framework for investigating the sensitivity of bagging predictors as follows: (1) a random under-sampling (RUS) method is used to change the original data-set into nine new data-sets with different imbalanced class distribution; (2) a 10-trial 10-fold cross-validation (CV) is performed on each altered data-set; (3) statistical methods are applied to draw validated conclusions; and (4) two evaluation metrics are adopted to further visualize the sensitivity of bagging predictors.

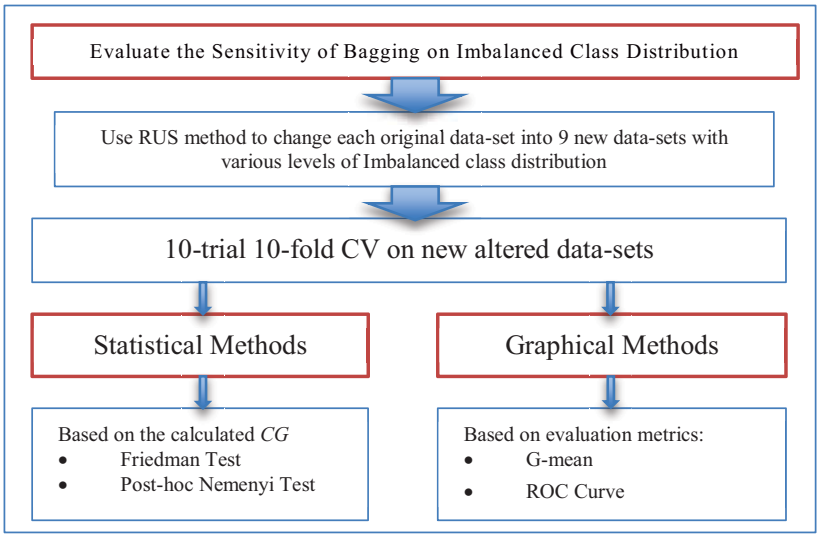

Figure 5.1: Designed framework

5.2.1 Sensitivity of Bagging Predictor

Definition 3 *Sensitivity refers to whether there is large change in the performance of bagging predictors, when the levels of imbalanced class distribution vary. If there is a large change in the performances of the bagging predictor, it means the bagging predictor is sensitive; on the other hand, if there is little or no change in the performances of the bagging predictor, it means the bagging predictor is insensitive.*

The assessment of the sensitivity of bagging predictors is based on the ranking of the changed G_{mean} (CG) of bagging predictors between two adjacent levels of class distribution, and the average rank of bagging predictors is compared. The bagging predictor with the largest mean rank indicates that the bagging predictor is sensitive; by contrast, the bagging predictor with the lowest mean rank indicates that the bagging predictor is insensitive to changing levels of class distribution.

The details of how to calculate the mean rank of the sensitivity of bagging predictors and how perform the statistical analysis to assess the sensitivity of bagging predictors are as shown below.

5.2.2 Friedman Test with Post-hoc Nemenyi Test

In the statistical method, the Friedman test with corresponding post-hoc Nemenyi test (Demšar 2006) is used to compare multiple bagging predictors:

1. the changed G_{mean} (CG) between two adjacent levels of class distribution is calculated; there are eight CG values for each bagging predictor on each original imbalanced data-set.

2. CG is used to rank bagging predictors on each data-set; the lowest CG value is ranked as 1, the second lowest is ranked as 2, and so on. If there is a tie, the average value of their ranks is used. For each bagging predictor there are eight ranks on each data-set, and 112 ranks over 14 of the data-sets.

3. The Friedman test is used to obtain the average rank of CG among 12 bagging predictors over all data-sets.

4. The post-hoc Nemenyi test is used to calculate "critical difference" (CD), and to determine where there is a significant difference among bagging predictors. The bagging predictors with the smallest value of average ranks are insensitive, meaning that the bagging predictors work well in imbalanced class distribution situations. The bagging predictors with the largest value of average ranks are sensitive to different levels of class distribution, meaning that the bagging predictors do not work well in extremely imbalanced class distribution situations.

5.2.3 Evaluation Metrics

The graphical method visualizes the performance of selected bagging predictors to further examine the statistical results:

1. A *ROC* curve is used to plot the pairs of False Positive Rate (FPR) and True Positive Rate (TPR) on the x-axis and y-axis, respectively. In this study, a *ROC* curve is used to represent the performance of each bagging predictor at nine different levels of class distribution.

2. G_{mean} is calculated by equation 2.8 to summarize and to monitor the accuracy rates of both TPR and TNR for the minority and majority classes, respectively.

5.3 Experimental Results

5.3.1 Statistical Analysis

Table 5.1: Statistical results of Wilcoxon signed-rank test

	Wilcoxon Signed-Rank Test											
	MLP	NB	KNN	PART	DStump	SVM	NBTree	J48	RdTree	OneR	RepTree	DTable
MLP .	-	**.881**	.004	.001	.018	.000	.000	.000	.000	.000	.000	.000
NB			.027	.032	.042	.008	.001	.000	.000	.000	.000	.000
KNN				**.528**	**.684**	**.398**	**.112**	.010 .	.000	.000	.000	.000
PART					**.813**	**.519**	**.198**	.001	.002 .	.000	.000	.000
DStump						**.444**	**.441**	**.079**	**.057**	.000	.000	.000
SVM							**.825**	**.188**	**.086**	.001	.001	.000
NBTree								**.282**	**.115.**	.000	.000	.000
J48									**.855**	.016	.000	.000
RdTree										.018	.002	.000
OneR											**.837**	.002
RepTree												.000
DTable												-

Table 5.1 presents the results of the Wilcoxon signed-rank test to indicate which bagging predictors are more sensitive to the different imbalanced levels of class distribution. The experimental results are based on the absolutely different value of the G_{mean} between two adjacent levels of class distribution. For each original data-set, we compared the two adjacent levels of the nine levels of class distribution, and there were eight groups with an absolutely different value of G_{mean} for every bagging predictor on each data-set. We ranked all 12 bagging predictors on each of eight groups on an original data-set, so there are 112 ranks for each bagging predictor on 14 data-sets.

The Wilcoxon signed-rank test is used to compare the 112 pairs of ranks of two bagging predictors to determine whether there is a significant statistical difference between the two bagging predictors. If the statistical value is equal to or greater than value .05, they are highlighted in red and there is no significant difference between the two bagging predictors. Otherwise, there is a significant difference between the two bagging predictors. The test results indicate that there is no statistically significant difference between bagging MLP and bagging NB. Both are statistically significantly different from the remaining 10 bagging predictors. However, the Wilcoxon signed-rank test only shows whether their is a statistically significantly difference between two bagging predictors and does not show which bagging predictor is more sensitive. The Friedman test is required to show the sensitive ranking order of bagging predictors.

Figure 5.2 presents a comparison of all bagging predictors with the Nemenyi test, where the x-axis indicates the average rank of CG performance of the bagging predictors; the y-axis indicates the ascending order of the average rank of CG performance, which represents bagging predictors from insensitive to sensitive; and the horizontal bars indicate the CD. If the horizontal bars between bagging predictors do not overlap, it means there is a statistically significant difference between the bagging

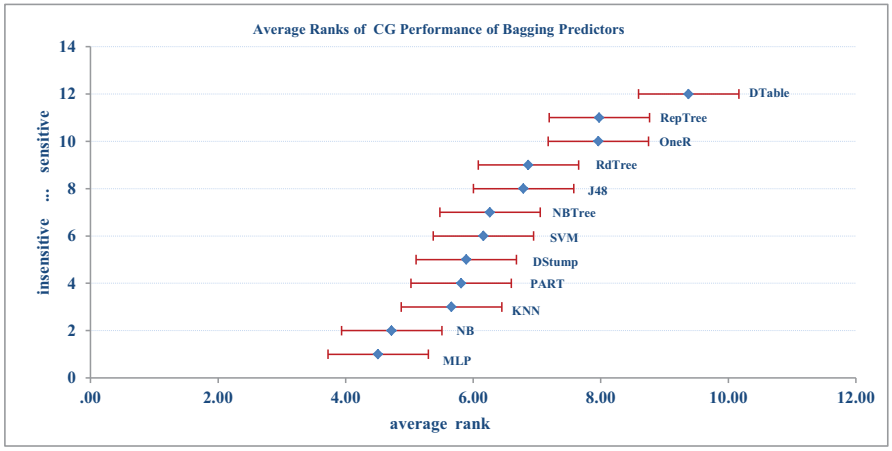

Figure 5.2: Comparison of all bagging predictors with the Nemenyi test, where the x-axis indicates the average rank of the bagging predictors, the y-axis indicates the ascending order of the average rank of CG performance, and the horizontal bars indicate the CD.

predictors at a 95% confidence interval. The results indicate that the bagging predictors Multi-layer Perceptron (MLP) and Naïve Bayes (NB) are the most insensitive predictors, which means that the performance of the bagging predictors changes gradually between adjacent levels of class distribution, so they are insensitive to varying levels of class distribution; By contrast, the bagging predictors Decision Table (DTable), RepTree and OneR are the most sensitive predictors, which means that the performance of those bagging predictors changes sharply between adjacent levels of class distribution, so they are sensitive to varying levels of class distribution. The ranking order of the CG performance of the sensitive bagging predictors is therefore greater than that of the insensitive bagging predictors when the levels of class distribution change. There are statistically significant differences between the two groups.

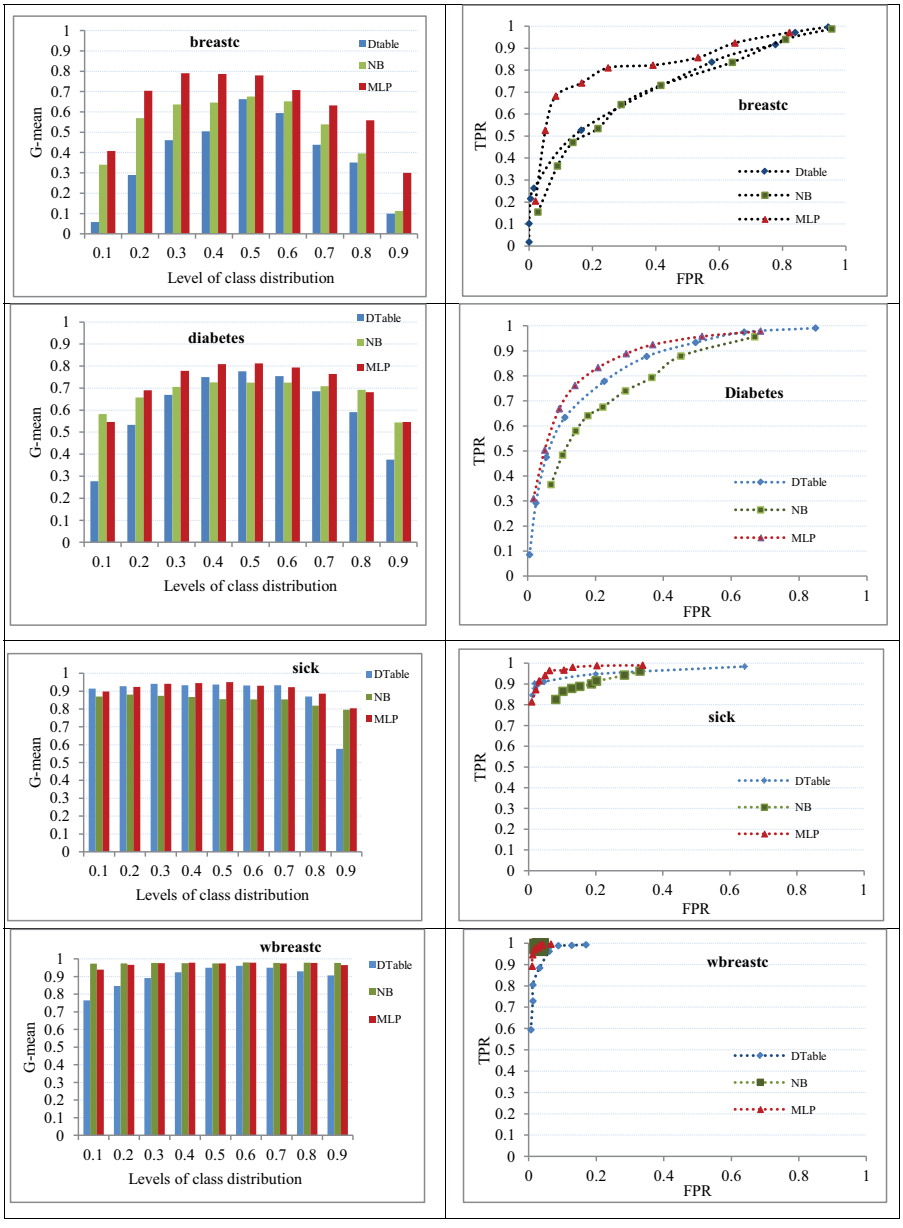

Figure 5.3: Comparison of *ROC* curve and G_{mean} among selected bagging predictors and data-sets.

5.3.2 Graphical Analysis

Graphical methods are used to further assess the sensitivity of bagging predictors, thus we only select three bagging predictors: two insensitive bagging predictors and one highly sensitive bagging predictor for comparisons over four original data-sets.

The first column of Figure 5.3 presents the G_{mean} comparisons of three bagging predictors at nine levels of class distribution on four original data-sets, where the x-axis indicates the levels of class distribution, and the y-axis indicates the G_{mean} value. The second column of Figure 5.3 presents the ROC curves of three bagging predictors at nine levels of class distribution on four original data-sets, where the x-axis indicates the FPR, and the y-axis indicates the TPR of the ROC curve. Each plot shows the ROC performance of three bagging predictors.

The first and second rows of the four sub-figures present graphical comparisons of G_{mean} and ROC curves of three selected bagging predictors at nine levels of class distribution on the *Breastc* and *Diabetes* data-sets, respectively. When the levels of class distribution are changed, the G_{mean} performance of bagging NB and MLP changes gradually, while bagging DTable changes sharply. The ROC curves indicate that Bagging MLP and NB have more points close to the " perfect point" and better performance than bagging DTable at same level of extremely imbalanced class distribution, eg., at 10%, 20%, 80%, and 90% levels of imbalanced class distribution. The range of TPR and FPR for bagging DTable varies from 0 to 1, and it is larger than the range for bagging MLP and NB, so bagging DTable is more sensitive than bagging MLP and NB. All three bagging predictors perform well around balanced class distribution, e.g., at 0.5 level. On the other hand, in the last two rows, for sub-figures for *Sick* and *WBreastc* data-sets, the G_{mean} performances of bagging MLP and NB show almost no change when the levels of class distribution change, while

the G_{mean} performance of bagging DTable changes gradually; therefore, bagging MLP and NB are more insensitive than bagging DTable. The graphical results are thus consistent with the statistical results. For the second last plot on the *Sick* data-set, the range of FPR for bagging DTable is larger than the range for bagging MLP and NB; thus bagging DTable is more sensitive than bagging MLP and NB. For the last plot on $WBreastc$ data-set, the performance of bagging predictors MLP and NB is extremely good, nearly reaching the perfect point (0, 1); therefore bagging predictors MP and NB are more insensitive than bagging DTable.

The graphical observations confirm that bagging predictors, MLP and NB are insensitive to various levels of class distribution and perform relatively well with extremely imbalanced class distribution. Our observations are consistent with the statistical tests.

5.4 Conclusion

This chapter empirically investigates the sensitivity of bagging predictors with respect to various levels of class distribution, using statistical and graphical methods. Our statistical results demonstrate that bagging MLP and NB are insensitive to different levels of imbalanced class distribution, which means the performances of bagging MLP and NB do not change much when the levels of class distribution change. Moreover, our observations by graphical methods are consistent with our statistical results. Regarding the sensitivity of bagging predictors when class distribution varies, there is no statistically significant difference between bagging MLP and bagging NB; however, there are statistically significant differences between the group of two bagging predictors MLP and NB and the other group of bagging predictors. This finding provides a useful guide for data mining practitioners to understand the sensitivity of bagging predictors for imbalanced class distribution applications.

Chapter 6

The Impact of Class Distribution on Bagging

This chapter investigates the effect of imbalanced class distributions on bagging performance. Imbalanced class distribution refers to the number of training samples that are unevenly distributed among different classes. Research into learning from imbalanced class distribution has increasingly captured the attention of both academia and industry. Many real world applications involve highly imbalanced class distribution; however, most traditional classification learning algorithms are designed to maximize the overall accuracy rate and assume that training instances are uniformly distributed. Despite the popularity of bagging in many real-world applications, some questions have not been clearly answered in the existing research, such as the effect of varying the levels of class distribution on the performance of bagging predictors, e.g., whether the performance of bagging is superior to single learners when the levels of class distribution change. This chapter proposes a unique approach to investigate the effects of varying levels of imbalanced class distribution on the performance of bagging predictors with 12 underlying base learners by using statistical and

graphical methods to assess three evaluation metrics, Geometric mean (G_{mean}), True Positive Rate (TPR), Receiver Operating Characteristic (ROC) graph, and Area Under the ROC curve (AUC) on imbalanced data-sets.

The experimental results based on both G_{mean} and TPR evaluation metrics over multiple imbalanced data-sets demonstrate that PART and Multi-layer Proceptron (MLP) are the learning algorithms with the best bagging performance on imbalanced data-sets; moreover, only four out of 12 bagging predictors are statistically superior to single learners. The experimental results based on AUC indicate that Decision Table (DTable) and RepTree are the learning algorithms with the best bagging performance; in addition, the performances of the bagging predictors is statistically superior to single learners, with the exception of Support Vector Machines (SVM) and Decision Stump (DStump).

This chapter is organized as follows. Section 6.1 provides an introduction. Section 6.2 presents the outline of the designed framework. Section6.3 provides the experimental setting. Section 6.4 presents the experimental results analysis to compare the performance of bagging with each of the single learners and to rank all bagging predictors with respect to different imbalanced levels of class distribution. Section 6.5 concludes the chapter.

6.1 Introduction

Imbalanced class distribution refers to the number of training samples that are unevenly distributed among different classes. It is considered to be one of ten challenging problems in data mining research (Yang & Wu 2006). Research into learning from imbalanced class distribution has increasingly captured the attention of both academia and industry, due to the increasing

number of real world applications involving extremely skewed class distribution, e.g., fraud detection (Chan & Stolfo 1998, Phua et al. 2004), text classification (Chawla et al. 2003), medical diagnostics (Mena & Gonzalez 2006, Rao, Krishnan & Niculescu 2006, Mazurowski et al. 2008), and detection of oil spills using satellite images (Kubat, Holte & Matwin 1998).

In those extremely imbalanced class distribution situations, the number of instances in the majority class (called negative samples) are much greater than those in the minority class (called positive samples), and the minority class is more important and has a notably higher cost than the majority class. High accuracy of a minority class is therefore required; however, the overall accuracy is an ineffective measure for extremely imbalanced data (Weiss & Provost 2003, Qin 2005, Chawla et al. 2003, Weiss 2004, Sun, Kamel, Wong & Wang 2007), because the most traditional learning algorithms attempt to maximize the overall accuracy rate, which results in a higher prediction rate for correctly classifying the majority class and a lower prediction rate for correctly classifying the minority class (Maloof 2003, Chawla, Japkowicz & Kotcz 2004, Chawla et al. 2003, Koknar-Tezel & Latecki 2009, Su & Hsiao 2007). Therefore, a simple estimated error rate has limitations in evaluating the performance of a classifier on a minority class (Fawcett 2006); the Area Under the Receiver Operating Characteristic (ROC) curve (AUC) is a commonly used evaluation metric for imbalanced class distribution (Bradley 1997), and it is considered to be an alternative measure for comparing the performance of classifiers across the entire range of class distributions and error costs (Provost & Fawcett 1997, Provost et al. 1998, Ling et al. 2003).

In the literature, techniques for solving the imbalanced class distribution problem have been proposed at data-level (Chawla et al. 2002, Han et al. 2005, Estabrooks, Jo & Japkowicz 2004, Batista et al. 2004, He, Han &

Wang 2005, Bunkhumpornpat et al. 2009) and at algorithm-level (Cieslak & Chawla 2008, Liu, Chawla, Cieslak & Chawla 2010). The commonly used techniques are sampling methods at data-level, e.g., over-sampling and under-sampling methods. Under-sampling is considered to be an efficient method for imbalanced class distribution learning (Liu et al. 2009).

Chapter 4 reports on statistical comparisons to investigate the performance of bagging intensively across a rich set of base learners in general terms (Liang et al. 2011*a*). However, error rate is not an appropriate performance measure for imbalanced learning, because the experimental results cannot be applied to imbalanced situations; in addition, a comprehensive study of the effect of varying levels of class distribution on the performance of bagging predictors has not been undertaken. Bagging has been widely applied in many real world applications, but some practical questions have not been clearly answered; for example, which bagging predictors are the best learning algorithms based on their average performance, when the imbalanced levels of class distribution change, and in such situations, whether bagging is superior to single learners. Answering these questions poses the following research challenges: (1) how should the performance of bagging be evaluated at different degrees of imbalanced class distribution ; and (2) how should a valid and rigorous study be conducted to evaluate multiple algorithms over multiple imbalanced data-sets.

This chapter investigates the impact of class distribution on the performance of bagging predictors on imbalanced data, and the investigation utilizes an under-sampling technique to alter the class distribution at different imbalanced levels. Both statistical and graphical methods have been adopted to analyze the impact of class distribution on bagging performance over imbalanced data-sets. The statistical analyses (Demšar 2006) performed instil confidence in the validity of the conclusions of this research.

Our main contribution is to conduct statistical and graphical comparisons to investigate the impact of class distribution on the performance of bagging predictors based on the evaluation metrics, G_{mean}, TPR, ROC graph, and AUC of ROC curve on imbalanced data-sets. Overall, the experimental results provide a useful guide for data mining practitioners to choose the best or the most effective learners when using bagging predictors for imbalanced applications.

6.2 Designed Framework

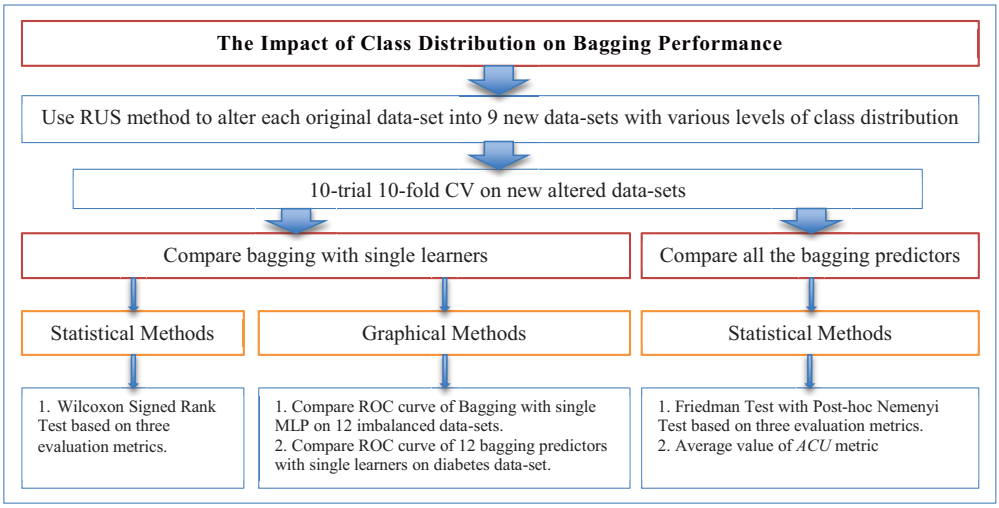

Figure 6.1: Designed framework

Figure 6.1 presents the designed framework of the evaluation of the impact of class distribution on the performance of bagging predictors, which is divided into tasks as follows:

1. Utilize the under-sampling technique to alter each original imbalanced data-set into nine new data-sets with different levels of imbalanced class distribution.

2. Perform 10-trial 10-fold cross-validation evaluation on the nine altered data-sets to obtain nine pairs of False Positive Rate (FPR) and True Positive Rate (TPR) for each learner to form a ROC curve, which represents the performance of a prediction model at nine different levels of class distribution.

3. The calculated AUC of ROC curve represents the average performance of a prediction model at nine different levels of class distribution, so the AUC of ROC curve is employed to compare the performance of bagging predictors with single learners;

4. Statistically compare the evaluation metrics, G_{mean}, TPR, and AUC as follows:

   (a) using the Wilcoxon signed-rank test for comparing two learners, e.g., comparing bagging and single learner to determine whether bagging is superior to single learner, and

   (b) using the Friedman test for comparing multiple learners, e.g., comparing all the bagging predictors to determine which predictors have the best performance.

   (c) using the post-hoc Nemenyi test for determine whether there is a statistically significant difference between bagging prediction models

5. Graphically compare the ROC graphs to determine whether bagging MLP is superior to the single MLP at different levels of class distribution on 12 data-sets; also graphically examine the ROC curve to compare 12 bagging predictors with single learners on the Diabetes data-set.

6.2.1 Random Under-sampling Technique Varying the Levels of Class Distribution

Table 6.1: Under-sampling technique altering the levels of class distribution

Class Distribution P_i%	Data Set	Sample Size		
		Total	Positive Samples	Negatives Samples
Original Data	D	$M = P + N$	P	N
10%	D_1	M_1	$P_1 = P = 10\%M_1$	$N_1 = 90\%M_1$
20%	D_2	M_2	$P_2 = P = 20\%M_2$	$N_2 = 80\%M_2$
.	.	.	.	.
.	.	.	.	.
10*i%	D_i	M_i	$P_i = P = 0.1 * i * M_i$	$N_i = (1 - 0.1 * i)M_i$
.	.	.	.	.
90%	D_9	M_9	$P_9 = P = 90\%M_9$	$N_9 = 10\%M_9$

For this work, the random under-sampling technique is used to vary the levels of class distribution of the original data to investigate the performance of bagging predictors, i.e., to alter each original imbalanced data-set, D with sample size, M into nine new data-sets, $D_1, D_2, ..., D_9$ with new sample size, $M_1, M_2, ..., M_9$, respectively.

First, all the minority class samples are considered as a positive class, sample size, P and the proportions of the positive class are as follows: 10%, 20%, ..., 90% of $M_1, M_2, ..., M_9$, respectively. Next, the majority class samples are selected randomly without replacement as a negative class, with samples size $N_1, N_2, ..., N_9$, and the proportions of the negative class are as follows: 90%, 80%, ..., 10% of $M_1, M_2, ..., M_9$, respectively. Then the nine new data-sets, D_i (integer $i = 1$ to 9) are formed. Each original data-set D is thereby altered into nine new data-sets with nine different levels of class distribution.

A 10-trial 10-fold cross-validation is performed on each of the new data-sets D_i, so that the test-set has the same distributions as the training-set.

6.3 Experimental Setting

We implement the bagging prediction model in Java, and use WEKA implementations of the 12 algorithms with default parameter settings in this empirical study (Witten & Frank 2005). We investigate the *AUC* performance of bagging predictors with respect to nine different imbalanced levels of class distribution on multiple imbalanced binary-class data-sets collected from the UCI Machine Learning Repository (Merz & Murphy 2006).

Under-sampling technique is used to alter each original imbalanced data-set D, sample size $|M|$ into nine new data-sets D_i (integer $i = 1$ to 9) samples size $|M_i|$ with nine different imbalanced levels of class distribution. The original minority class (sample size $|P|$) is used as the positive class in the new data-sets ($|P_i| = |P|$). Table 6.1 describes the under-sampling technique for altering different levels of class distribution.

For each learning algorithm, we perform a 10-trial 10-fold cross-validation evaluation on each of the new altered nine data-sets to obtain nine pairs of FPR and TPR for the single learner and nine pairs of FPR and TPR for bagging to form *ROC* curves.

Table 6.2 presents the sampling process to generate new altered data-sets, build prediction models, and form *ROC* curve, for each algorithm, e.g., for SVM, on the new altered nine data-sets with different class distribution, we build nine single SVM prediction models and nine bagging SVM prediction models, respectively to form two *ROC* curves: one for the single learner and one for the bagging predictor. Overall, for each learning algorithm on each original data-set, we build 18 models to form two *ROC* curves. We investigate the *AUC* performance of bagging predictors with 12 algorithms at nine levels of sample distributions on 14 data-sets. As a result, we built

Table 6.2: Sampling techniques are used to change each original data-set into 9 altered data-sets with 9 levels of class distribution for building 9 single and bagging final prediction models, respectively. These prediction models produce 9 pairs (FPR, TPR) to form a *ROC* curve for single and bagging prediction models, respectively.

	$N^{\#}$ Algorithms	$N^{\#}$ Original Data-Sets	$N^{\#}$ levels Class Distribution	$N^{\#}$ Altered New Data-sets	$N^{\#}$ Models		$N^{\#}$ pairs (FPR, TPR)	$N^{\#}$ *ROC* curves
					Single	Bagging		
	1	1	9	9	9	9	18	2
	12	1	9	9	108	108	216	24
	12	14	9	126	1512	1512	3024	336
Total	12	14	9	3024	3024		3024	336

3024 prediction models in total to evaluate the *AUC* performance of bagging predictors.

To reduce uncertainty and obtain reliable experimental results, all the evaluations are assessed under the same test conditions using the same randomly selected bootstrap samples (with replacements) in each fold of the 10-trial 10-fold cross-validation on each data-set.

6.4 Experimental Results Analysis

The experimental results analysis includes three subsections as follows: (1) Subsection 6.4.1 statistically compares bagging with each of the single learners based on the Wilcoxon signed-rank test; (2) Subsection 6.4.2 graphically compares *ROC* curves between a bagging MLP and single MLP on 12 selected imbalanced data-sets, and between 12 bagging predictors and single learners on Diabetes data-set, respectively; and (3) Subsection 6.4.3 compares the performance of bagging predictors based on the Friedman test with post-hoc Nemenyi test.

6.4.1 Statistical Comparison Bagging Predictors with Single Learners

This subsection compares the performance of bagging and single learners based on the Wilcoxon signed-rank test by using three different evaluation metrics: G_{mean}, TPR, and AUC of ROC curve.

The Wilcoxon signed-rank test is used to determine whether there is statistically significant difference between two learners, i.e., AUC performance of bagging SVM and single learner SVM.

- **The Null Hypothesis** shows that the median of difference between Bagging and each single learner equals 0.
- **Rule**: Reject the Null Hypothesis if the p-value Test Statistic W is less than $\alpha = .05$ at the 95% confidence level of significance.

Table 6.3: The statistical results of the Wilcoxon signed-rank test for comparison of the G_{mean} performance of bagging and single learners. The significance level is .05.

Wilcoxon signed-rank test based on G_{mean}						
Learners	J48	RepTree	RandTree	NB	SVM	Dstump
p-values	.005	015	.008	**.610**	**.131**	**1.000**
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	.037	**.814**	.005	**.657**	**.136**	.019

Tables 6.3, 6.4 and 6.5 present the summarized results of the Wilcoxon signed-rank test for the comparison of the performance of bagging predictors with each of single learners based on the evaluation metrics, G_{mean}, TPR,

and AUC. If a calculated p-value is equal to or greater than α value, 0.05, then the p-values are highlighted in red and the Null Hypothesis is accepted.

Table 6.3 indicates that only 50% bagging predictors perform statistically significantly better than single learners at 95% confidence interval, including $J48$, RepTree, RandTree, OneR, PART and MLP learners, when using G_{mean} as an evaluation metric on multiple imbalanced data-sets.

Table 6.4 indicates that less than 50% of bagging predictors perform statistically significantly better than the single learners at 95% confidence interval, including $J48$, RepTree, RandTree, PART and NBTree learners, while using TPR as an evaluation metric on imbalanced data-sets.

Overall, based on both G_{mean} and TPR evaluation metrics, only four out of 12 bagging predictors are statistically significantly superior to single learners on multiple imbalanced data-sets. The four bagging predictors are the tree family learners, $J48$, RepTree and RandTree, and the rule learner, PART.

Table 6.5 indicates that the p-values of SVM, and DStump are greater than α value, .05, and the Null Hypothesis is accepted, so there are no statistically significant differences between bagging SVM and bagging DStump with single learners SVM and DStump, respectively; while the p-value of the remaining cases is smaller than α value, .05, and the Null Hypothesis is rejected, so there are statistically significant differences between bagging predictors with the rest of the single learners. In addition, we observe that the AUC of the majority of bagging predictors is larger than that of the single learners. The experimental results therefore demonstrate that the AUC performances of bagging are statistically significantly better than most single learners at 95% confidence interval, except for SVM and DStump learners.

Table 6.4: The statistical results of the Wilcoxon signed-rank test for comparison of the TPR performance of bagging and single learners. The significance level is .05.

Wilcoxon signed-rank test based on TPR						
Learners	J48	RepTree	RandTree	NB	SVM	DStump
p-values	.015	.002	.005	**.906**	**.722**	**.263**
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	**.110**	**.272**	.006	**.575**	.002	**.136**

Table 6.5: The statistical results of the Wilcoxon signed-rank test for comparison of the AUC performance of bagging and single learners. The significance level is .05

Wilcoxon signed-rank test to compare the AUC performance						
Learners	J48	RepTree	RandTree	NB	SVM	Dstump
p-values	.004	.035	.004	.001	**.096**	**.074**
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	.031	.001	.004	.008	.026	.019

6.4.2 Graphical Comparison of *ROC* Curves

Subsection 6.4.2 compares the group of *ROC* graphs of 12 bagging predictors with single learners to further graphically examine whether bagging MLP (B_MLP) is superior to single MLP on 12 selected imbalanced data-sets, and whether 12 bagging predictors are superior to single learners on the single Diabetes data-set, respectively.

Figure 6.2 presents the group of comparisons of *ROC* curves between the B_MLP and the single learner MLP on 12 imbalanced data-sets; each sub-figure presents two *ROC* curves, one for bagging and the other for a single learner on each data-set, and each *ROC* curve is formed by nine pairs of FPR and TPR, which represent the average performance of bagging or single learners at nine degrees of class distribution.

Figure 6.2 indicates that B_MLP is superior to a single learner MLP on two out of 12 imbalanced data-sets, *bupa* and *Diabetes*. B_MLP has a greater area than a single learner MLP, and therefore B_MLP has better average performance than the single learner MLP on these two data-sets. The experimental results are consistent with the result of Wilcoxon signed-rank test on the metric of TPR in Table 6.4, that B_MLP is not superior to single MLP.

Even though the average performance of B_MLP is not superior to a single learner MLP, it has similar results to the single learner MLP on 10 imbalanced data-sets. On the $WDBC$ data-set especially, both B_MLP and the single learner MLP perform extremely well on nine different imbalanced levels, as all nine pairs of FPR and TPR are close to the "*ROC* Heaven", the upper left point (0, 1), and present almost 100% true positive and zero false positive.

Figure 6.3 indicates the group of comparisons of *ROC* curves between bagging and single learners to examine whether bagging is superior to single learners on the *Diabetes* data-set. Each sub-figure shows two *ROC* curves:

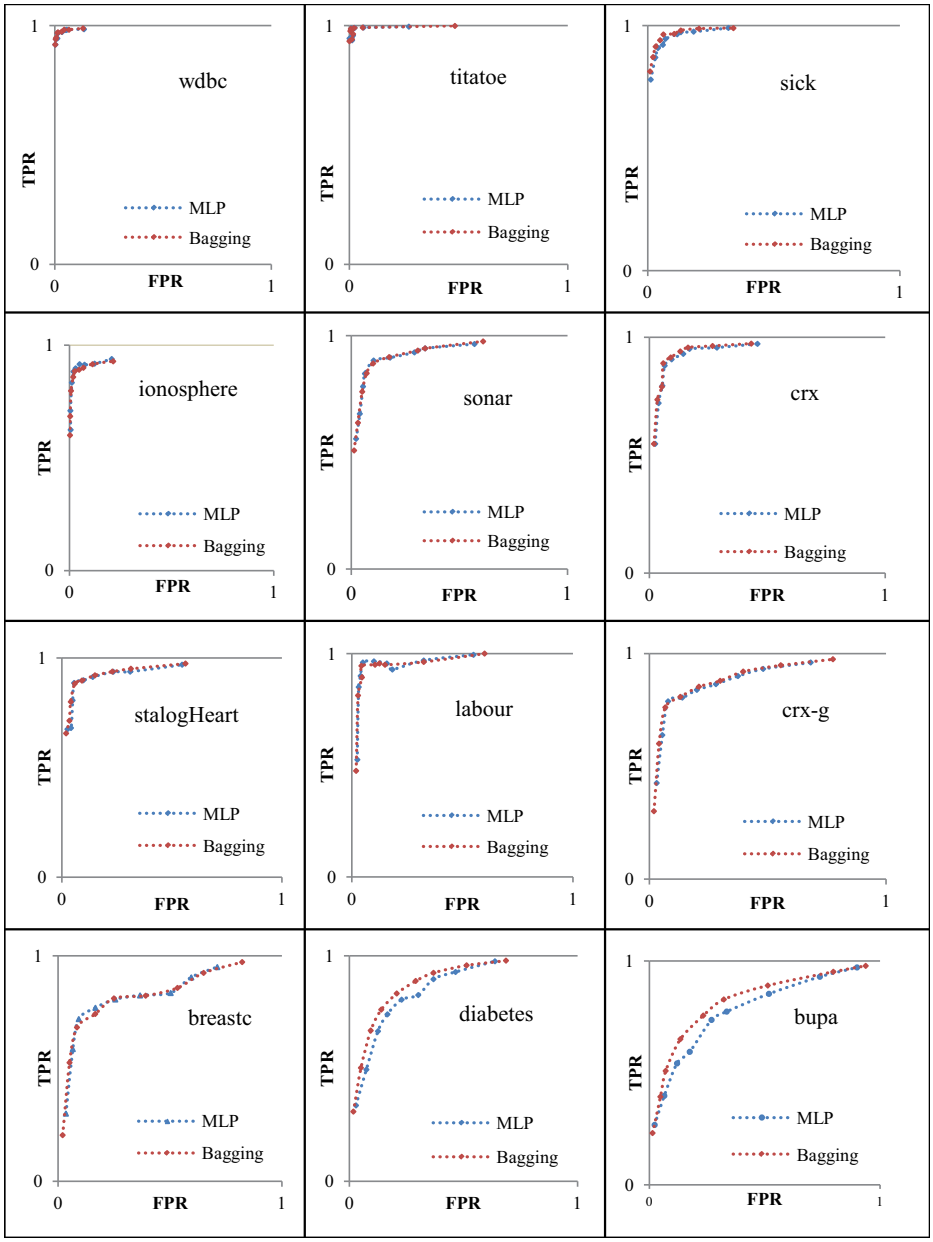

Figure 6.2: Comparisons of *ROC* curves between a B_MLP and a single learner MLP on 12 imbalanced data-sets, where the x-axis denotes FPR, the y-axis denotes TPR for each sub-figure.

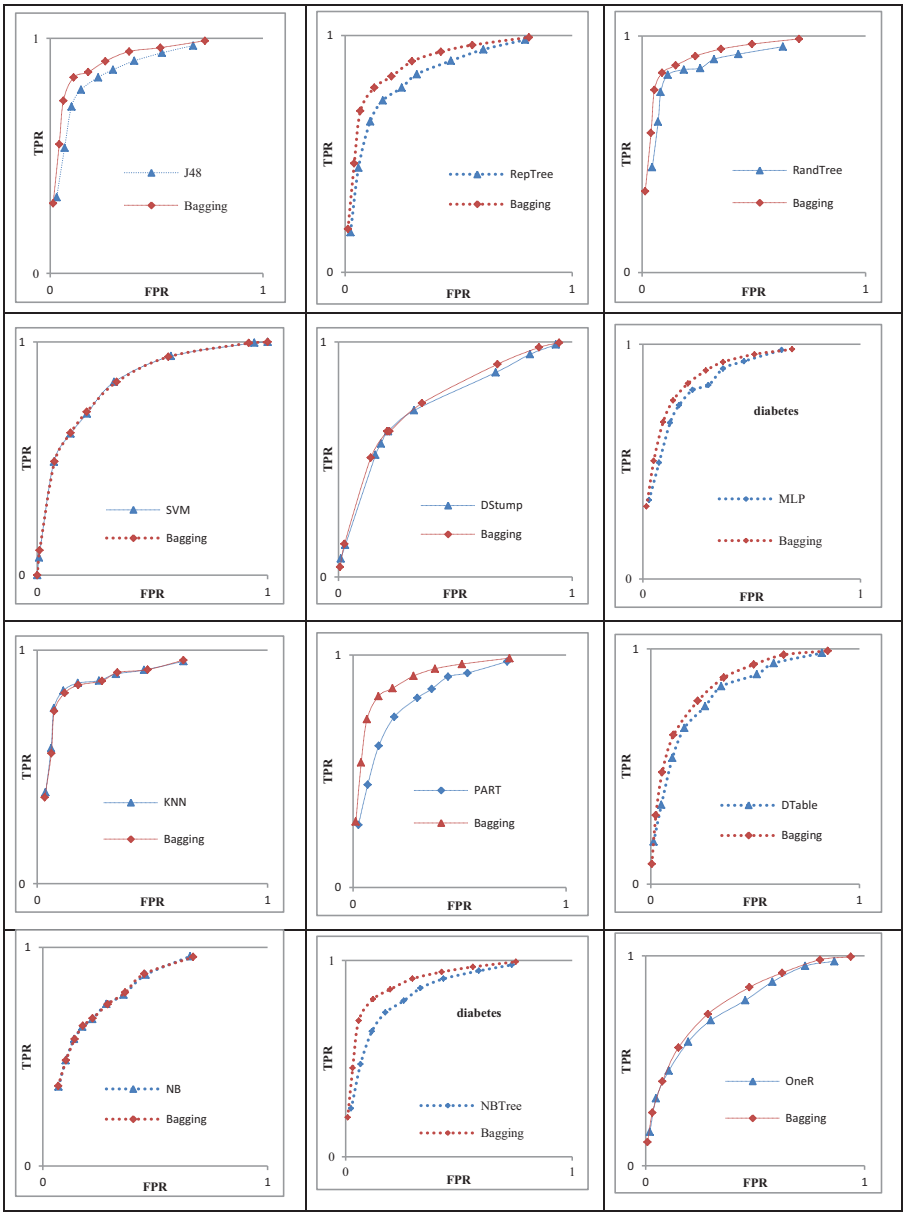

Figure 6.3: The group of comparisons of *ROC* curves between 12 bagging predictors and single learners on the *Diabetes* data-set, where the x-axis denotes FPR, the y-axis denotes TPR for each sub-figure.

for example, bagging NB (B_NB) and single NB, on the same data-set. Each *ROC* curve represents the average performance of bagging or a single learner at nine different levels of class distribution, plotted by nine pairs of FPR and TPR. It can be seen that the bagging performance of *ROC* is not superior to the single learners SVM, DStump, NB, KNN, and OneR. Even though the average performance of bagging is not superior to the five single learners, it has similar results to the five single learners and is superior to the rest of the seven single learners on the *Diabetes* data-set. These results are consistent with the results of the Wilcoxon signed-rank test on the metric of TPR in Table 6.4, expect that for bagging DTable (B_DTable) and B_MLP are superior to the single learner.

6.4.3 Comparison of the Performance of All Bagging Predictors

Subsection 6.4.3 compares bagging predictors based on Friedman with post-hoc Nemenyi tests for comparison of all bagging predictors with three evaluation measures, G_{mean}, TPR, and AUC of *ROC*, respectively.

The Friedman test is used to compare of multiple learners over multiple data-sets. For example, the bagging predictors are first ranked on each data-set according to their G_{mean} metric from 1 to 12, respectively, e.g., the best performance of the bagging predictor with the largest value of the G_{mean} is signed as ranking 1, with the second largest value signed as ranking 2, and so on; if there are ties, the averaged value of their ranking orders is signed as their ranking. The Friedman test is then performed to obtain the mean rank of all bagging predictors, which are presented in ascending order in Table 6.6.

Table 6.6 indicates the ascending order the average rank of bagging predictors based on an evaluation measure G_{mean}. The third and the last

rows indicate the mean rank of bagging predictors from the Friedman test results. The Friedman test is used to compare of multiple learners. First, the G_{mean} measure of 12 bagging predictors on each data-set is ranked, then the Friedman test is performed to obtain the mean rank of bagging predictors over multiple imbalanced data-sets. As the Null Hypothesis is rejected, the Friedman test indicates there is at least a difference between the mean ranks of bagging predictors; therefore, the corresponding post-hoc Nemenyi test is required for additional exploration of the differences between the mean rank to provide specific information on which mean rank is significantly different from another.

Table 6.6: Mean rank of the Friedman test for G_{mean} performance of bagging predictors

Mean rank of the Friedman test for G_{mean} performance of bagging predictors						
Bagging	B_PART	B_MLP	B_NBTree	B_J48	B_RdTree	B_SVM
Mean Rank	3.83	3.92	4.67	5.58	5.67	6.33
Bagging	B_NB	B_KNN	B_RepTree	B_DTable	B_Dstump	B_OneR
Mean Rank	6.5	6.83	7.08	8.75	9	9.93

Figure 6.4 reports the results of the Friedman with post-hoc Nemenyi tests to compare the performance of all bagging predictors based on the average rank of G_{mean} metric on multiple imbalanced data-sets. The group of the most robust base learners, PART and MLP, contributes to the best bagging predictors, whereas the group of the weakest learners, OneR and DStump, leads to the worst bagging predictors. The performance of two bagging predictors is significantly different when the horizontal bars do not overlap. There is a statistically significant difference between the two groups: the group of bagging predictors based on the most robust base

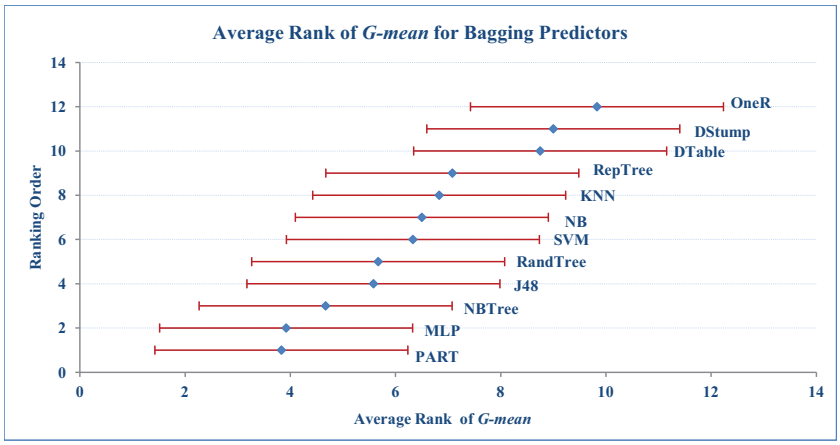

Figure 6.4: Comparison of the performance of all bagging predictors with post-hoc Nemenyi test, where x-axes indicate the mean rank of G_{mean} for bagging, the y-axes indicate the ascending ranking order of the bagging predictors and the horizontal error bars indicate the "critical difference".

Table 6.7: Mean rank of the Friedman test for TPR performance of bagging predictors

Mean rank of the Friedman test for TPR performance of bagging predictors						
Bagging	B_NBTree	B_MLP	B_PART	B_SVM	B_RdTree	B_RepTree
Mean Rank	3.92	4.92	5.92	6.08	6.23	6.54
Bagging	B_DTable	B_J48	B_KNN	B_NB	B_OneR	B_Dstump
Mean Rank	6.65	6.65	7.54	7.62	7.65	8.27

learners, PART and MLP and the group of bagging predictors based on the weakest learners, OneR and DStump. As a result, one can conclude that the robustness of the base learners is an important factor for building accurate bagging predictors, at different degrees of imbalanced class distribution.

Table 6.7 indicates the average rank of bagging predictors from the results of the Friedman test based on an evaluation metric TPR. As the Null Hypothesis is accepted, the Friedman test indicates that there is no difference between the mean ranks of the bagging predictors. Therefore, the corresponding post-hoc Nemenyi test is not required for additional exploration of the differences between mean ranks to provide specific information on which mean ranks are significantly different from one another.

Figure 6.5 presents the mean ranks of TPR for all bagging predictors and demonstrates that NBTree, MLP, and PART are the learning algorithms with the best bagging performance on imbalanced data-sets, while DStump and OneR are the learning algorithms with the worst bagging performance on imbalanced data-sets. However, the mean ranks of TPR for bagging predictors are not significantly different from one another.

Table 6.8: Mean rank of the Friedman test for AUC performance of bagging predictors

Mean rank of the Friedman test for AUC performance of bagging predictors						
Bagging	B_DTable	B_RepTree	B_OneR	B_RdTree	B_J48	B_PART
Mean rank	3.00	4.14	4.17	5.36	5.43	6.36
Bagging	B_NBTree	B_DStump	B_SVM	B_KNN	B_MLP	B_NB
Mean rank	6.50	7.07	7.64	9.00	9.21	9.57

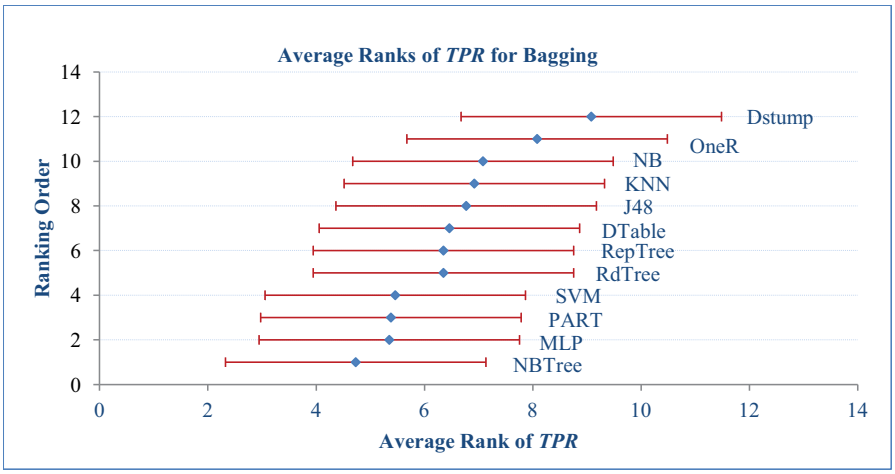

Figure 6.5: Comparison of the TPR performance of all bagging predictors with the Nemenyi test, where the x-axes indicate the mean rank of TPR for bagging predictors, the y-axes indicate the ascending ranking order of the bagging predictors, and the horizontal bars indicate the "critical difference".

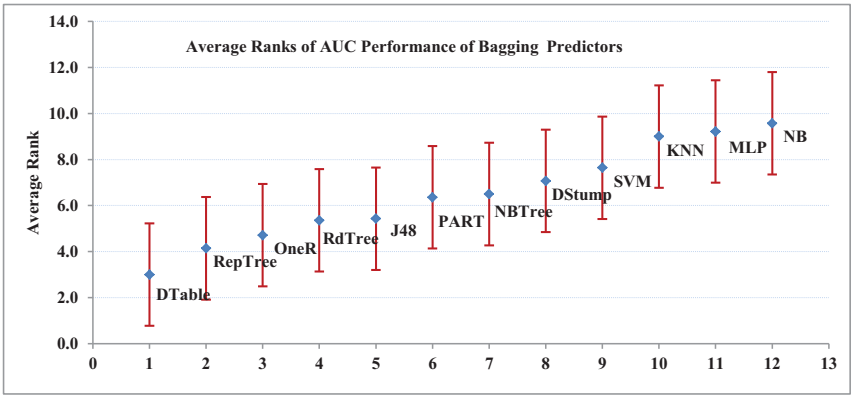

Figure 6.6: Average ranks of AUC performance for 12 bagging predictors with the Nemenyi test, where the x-axis denotes the ranking order of the bagging predictors, while the y-axis denotes the average rank of the AUC performance of the bagging predictors. The error bars present the "critical difference" of the Nemenyi test.

Table 6.8 reports the average rank of *AUC* of the bagging predictors. As the Null Hypothesis is rejected, the Friedman test indicates there is at least a difference between the mean ranks of bagging predictors. Therefore, the corresponding post-hoc Nemenyi test for additional exploration of the differences between the mean ranks provides specific information on which mean rank is significantly different from another.

Figure 6.6 presents an empirical comparison of the *AUC* performances of 12 bagging predictors, where the x-axis denotes the ranking order of the bagging predictors, and the y-axis denotes the average rank of the *AUC* value of the bagging predictors. The error bars indicate the "critical difference" of the Nemenyi test. The AUC performance of two bagging predictors is significantly different if the corresponding error bars do not overlap. Overall, the group of Bagging DTable and RepTree has the best average ranks of *AUC* performance, while the group of bagging NB, MLP and KNN has the worst average ranks of *AUC* performance. In addition, there are statistically significant differences between the two groups. The two dimensional robustness and stability decomposition is introduced in Chapter 4 to classify base learners into different categories. According to the experimental results, DTable and RepTree are categorized as unstable base learners, while NB, MLP and KNN are categorized as stable learners. We therefore demonstrate that based on *AUC* performance, the unstable base learners, DTable and RepTree contribute to the best bagging predictors; while the stable base learners, NB, MLP and KNN lead to the worst bagging predictors, when the imbalanced levels of class distribution are changed at nine different levels on each data-set, over all data-sets.

Table 6.9 presents the average *AUC* performance of bagging predictors on 14 data-sets. Mean and Variance indicate the average value of the *AUC* performance of bagging predictors and the corresponding value of error bars in Figure 6.7, respectively.

Table 6.9: Average *AUC* performance of bagging predictors on 14 imbalanced data-sets

Average *AUC* performance of bagging predictors						
Bagging	B_DTable	B_RepTree	B_OneR	B_RdTree	B_J48	B_PART
Mean	.668	.598	.585	.561	.552	6.36
vaiance	.48	.097	.037	.065	.082	.071
Bagging	B_NBTree	B_DStump	B_SVM	B_KNN	B_MLP	B_NB
Mean	.519	.511	.510	.453	.435	.420
vaiance	.064	.075	.084	.042	.082	.055

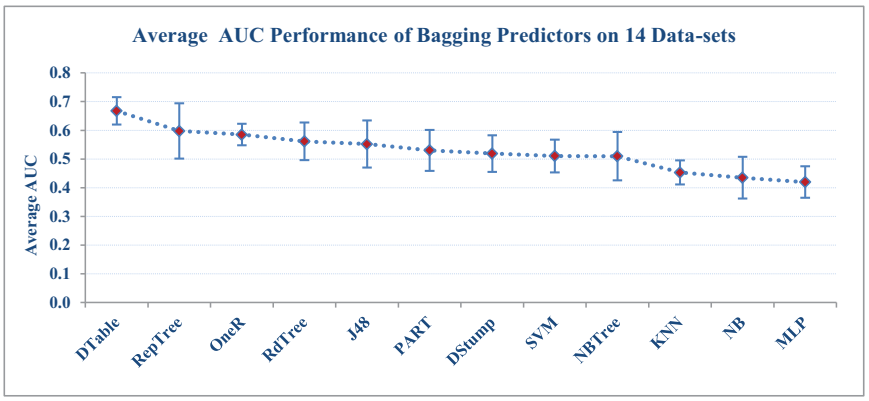

Figure 6.7: The average *AUC* performance of bagging predictors over 14 data-sets, where the x-axis indicates the name of the bagging predictors, the y-axis indicates the average value of *AUC* and the error bar indicates the variance value.

Figure 6.7 presents the summary of the observed average AUC performance of bagging predictors over multiple imbalanced data-sets in Table 6.9. In this plot, the vertical axis indicates the average value of the AUC performance of bagging predictors, while the horizontal axis indicates the sorted average AUC performance of bagging predictors in descending order over the total benchmark of imbalanced data-sets, and the error bar indicates the variance of the observed average AUC performance. We note that Figure 6.6 and Figure 6.7 present a similar ranking order of bagging predictors, with the exception of NBTree.

6.5 Conclusion

We empirically investigated the impact of class distribution on 12 bagging predictors based on three evaluation metrics on imbalanced data-sets. The under-sampling technique was utilized to alter the class distribution at different imbalanced levels. This chapter has used both graphical and statistical methods to analyze the experimental results to provide a full comparison of the performances of bagging predictors with 12 underlying base learners at different levels of class distribution.

Based on both G_{mean} and TPR evaluation metrics, we observe that 4 out of 12 bagging predictors are statistically superior to single learners, including tree family learners, $J48$, RepTree and randTree, and a rule learner PART. In addition, the experimental results indicate that the AUC performances of bagging are statistically superior to single learners, except for SVM and DStump over multiple data-sets. Comparing the AUC performances of bagging shows that, the group of unstable learners, DeciTable and RepTree is the learning algorithms with the best average bagging performance, while the group of stable learners, NB, MLP and KNN, leads to the worst bagging predictors; there also are significant

differences between the two groups. The graphical comparison of the *ROC* curve between bagging and single learners demonstrates that the *ROC* performance of bagging is not superior to single learners SVM, DStump, NB, KNN, and OneR, but it has similar results to the five single learners on the Diabetes data-set, while it is superior to the rest of the seven single learners. In addition, we observe that the strongest base learners PART, MLP and NBTree can be used to build the best bagging predictive models, whereas the weakest learners, OneR and DStrump, result in the worst bagging predictive models in the context of imbalanced learning.

Chapter 7

An Empirical Investigation of Bagging on Domain Specific Data

This chapter investigates the performance of bagging predictors in terms of learning from medical data, which includes two parts: (1) part one presents a graphic comparison of the performance between 12 bagging predictors and 12 single learners with respect to learning from the natural class distribution, and reports the best performance of the prediction model based on the G_{mean} evaluation measure on eight individual medical data-sets, and (2) part two presents a comparison of the performance of bagging predictors of natural class distribution and the best achieved bagging performance of altered class distribution on individual medical data-sets.

It is important for data miners to achieve highly accurate prediction models, and this is especially true for imbalanced medical applications. In these situations, practitioners are more interested in the minority class than the majority class; however, it is hard for most of the traditional supervised learning algorithms to achieve a highly accurate prediction on the minority

class, even though it might achieve better results according to the most commonly used evaluation metric, overall accuracy (Acc). Bagging is a simple yet effective ensemble method which has been applied to many real-world applications. However, some questions have not been well answered, e.g., whether bagging outperforms single learners on medical data-sets; which learners are the best predictors for each medical data-set; and what is the best predictive performance achievable for each medical data-set when we apply sampling techniques.

This study empirically investigates the performance of bagging predictors with 12 underlying base learners on eight medical data-sets based on four performance measures: True Positive Rate (TPR), True Negative Rate (TNR), Geometric Mean (G_{mean}) of the accuracy rate of the majority class and the minority class, and overall accuracy (Acc) as evaluation metrics. In addition, the statistical analyses performed instil confidence in the validity of the conclusions of this research.

The chapter is organized as follows. Section 7.1 provides an introduction to the chapter. Section 7.2 presents the outlines of the designed framework. Section 7.3 presents the experimental setting and Section 7.4 presents the experimental results analysis. Section 7.5 concludes this chapter.

7.1 Introduction

Bagging (Breiman 1996*a*) is a simple and effective ensemble learning method. Due to its promising capabilities in improving the performance of classification prediction models using a combination of sampling and voting techniques, it has been widely used in many applications. The effectiveness of bagging has been investigated empirically and it has been demonstrated that bagging is very effective for decision trees (Breiman 1996*a*, Opitz & Maclin 1999, Quinlan 1996, Bauer & Kohavi 1999, Dietterich 2000*b*), and

neural networks (Opitz & Maclin 1999, West et al. 2005, Kim & Kang 2010). Even though the existing studies demonstrate the effectiveness of the bagging predictor, it is not clear whether bagging is superior to single learners in the context of imbalanced medical data-sets, nor which predictor is the best performing learning method on individual imbalanced medical data-sets.

Our previous works investigated the performance of bagging predictors in general terms (Liang et al. 2011*a*) and in imbalanced class distribution terms (Liang & Zhang 2011*a*, Liang et al. 2011*b*). However, the previous conclusions are based on statistical tests that aggregate over multiple data-sets and do not show which learners are the best prediction models for individual medical data-sets, as various prediction models might behave differently for different kinds of data-sets. They also do not show the best achievable predictive performance for each medical data-set using a sampling technique.

In the literature, an empirical study of combined classifiers on medical data (Lopes et al. 2008) compared the performance of three classification methods, C4.5 (Quinlan 1986), bagging (Breiman 1996*a*), and boosting (Freund & Schapire 1996) on 16 medical data-sets and 16 generic data-sets. The evaluation was based on the accuracy of these learning methods as a performance measure; their research did not address the challenging issues of medical data-sets, namely, imbalanced class distribution and the unequal costs of mis-classification errors in different classes. Moreover, accuracy is an inappropriate performance measure for evaluating imbalanced data-sets (Chawla 2010, Chawla et al. 2002).

The majority of medical applications involve learning from imbalanced binary classification data-sets in which the proportion of the class distribution is skewed, the number of instances of the majority class is higher than those of the minority class, and practitioners are more interested in the minority class than the majority class, such as Breast cancer early detection, in which the

minority class is quite small with an unequal high cost associated with mis-classification errors in different classes. If a patient with Breast cancer is mis-classified as normal, the patient will miss the opportunity for his/her earlier stage cancer detection and treatment, while if a patient without Breast cancer is mis-classified as having cancer, it will cause unnecessary stress and the extra cost of treatment. Traditional supervised learning algorithms perform poorly in predictive accuracy over the minority class, even though they may produce high overall accuracy (Chawla 2010, Maloof 2003, Ng & Dash 2006, Phua et al. 2004, Su & Hsiao 2007). We therefore employ four evaluation metrics, True Positive Rate (TPR), True Negative Rate (TNR), geometric mean (G_{mean}) of the accuracy rate of the majority class and minority class, and overall accuracy (Acc) to assess the performance of bagging in terms of learning from medical data-sets.

To solve the problem of imbalanced class distribution and increase the Acc of the prediction model, the most commonly used methods are sampling-oriented methods and algorithms-oriented methods (Liu & Chawla 2011).

In this study, we utilize random under-sampling (RUS) techniques to investigate the performance of bagging predictors at different levels of class distribution and report the best achieved performance of bagging by using sampling techniques based on the G_{mean} evaluation metrics.

The main contributions of this chapter are threefold: (1)to determine whether bagging is superior to single learners in the context of specific domain, imbalanced medical data-sets; (2)to determine which learners give the best performance on each medical data-set with natural class distribution; and (3)to report the best achieved performance of the bagging predictors on each medical data-set by using sampling techniques.

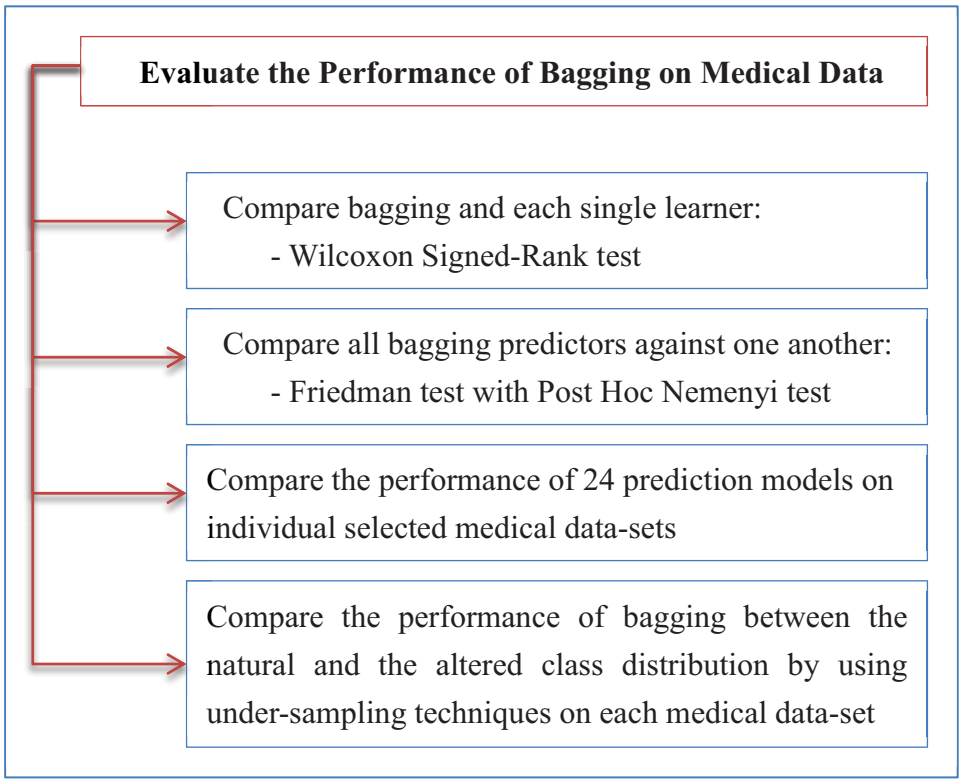

Figure 7.1: Designed framework

7.2 Designed Framework

Figure 7.1 presents the designed framework. The evaluation of bagging predictors on medical data-sets is broken down into four tasks as follows:

- Compare bagging predictors with single learners: the Wilcoxon signed-rank test is used to compare two learners to determine whether bagging outperforms a single learner on multiple medical data-sets.
- Compare the performance of all bagging predictors against each another: the Friedman test with the corresponding post-hoc Nemenyi test are used to compare multiple learners to determine which bagging predictors have the best performance over multiple medical data-sets.
- Compare the performance of 24 prediction models and report the best performance models with natural class distribution on individual medical data-sets based on four evaluation metrics: G_{mean}, TPR, TNR and Acc.
- Compare the performance of bagging predictors between the natural class distribution and the best achieved bagging performance with certain levels of altered class distribution on individual data-sets.

7.3 Experimental Setting

This section includes data-sets, software and parameter settings. 10-trial 10-fold cross-validations are performed to evaluate bagging and single learners on eight medical data-sets in Table 2.4, which are collected from the UCI Machine Learning Repository (Merz & Murphy 2006). The bagging predictor is implemented in Java platform for this study. WEKA implementation of the 12 algorithms with their default parameter settings is used in this empirical study (Witten & Frank 2005).

To reduce uncertainty and obtain reliable experimental results, all the evaluations of bagging performance are assessed under the same test conditions using the same randomly selected bootstrap samples with replacements in each fold of 10-trial 10-fold cross-validation on each data-set.

7.4 Experimental Results Analysis

Section 7.4 presents the experimental results analysis including four subsections as follows: Subsection 7.4.1 presents a comparison of bagging with single learners; Subsection 7.4.2 presents a comparison of all bagging predictors against one another on medical data-sets; Subsection 7.4.3 provides a graphical comparison of the performance of 24 prediction models and a report on the best prediction model on individual medical data-sets; and Subsection 7.4.4 presents a comparison of the performance of bagging predictors between natural class distribution and the altered class distribution on individual medical data-sets.

7.4.1 Comparison of Bagging with Single Learners

Subsection 7.4.1 compares the performance of bagging with single learners over multiple medical data-sets to determine whether bagging is superior to single learners based on G_{mean}.

The Wilcoxon Signed-Rank Test is used to compare two learners, for example, to compare bagging SVM and a single learner SVM over multiple data-sets to determine whether bagging is superior to a single learner.

Table 7.1 presents the summarized results of the Wilcoxon signed-rank test based on G_{mean} for the comparison of the two learners, bagging and each of the single learners, i.e., to compare bagging J48 and single learner $J48$ to

determine whether bagging $J48$ is superior to single learner $J48$ over multiple imbalanced medical data-sets. If the *p-value* is equal to or greater than α value .05, we accept the Null Hypothesis and the*p-value* are highlighted and marked in red. Table 7.1 indicates that bagging is statistically superior to the single learners $J48$, RandTree, OneR, PART and MLP on eight medical data-sets based on the evaluation metric, G_{mean}.

Table 7.1: Compare bagging with each single learner based on Wilcoxon signed-rank test on G_{mean}. The significance level is .05.

Wilcoxon Signed-Rank Test on G_{mean}						
Learners	J48	RepTree	RandTree	NB	SVM	Dstump
p-values	.036	**.161**	.036	**.069**	**.093**	**.866**
Learners	OneR	DTable	PART	KNN	NBTree	MLP
p-values	.017	**.779**	.036	**.327**	**.484**	.012

7.4.2 Comparison of All Bagging Predictors

Friedman test and post-hoc Nemenyi test: Both tests are non-parametric for comparing multiple bagging predictors over multiple imbalanced medical data-sets.

The Friedman test is used to compare the average rank of all bagging predictors, and the post-hoc Nemenyi test is used to check whether there is a statistically significant difference between the mean ranks as follows:

1. All the bagging predictors are ranked on each data-set, giving the best performing algorithm the rank of 1, the second best rank 2, and so on. If there are ties, average values are assigned.

2. The Friedman test is used to calculate the average rank of all the bagging predictors.

3. The Null Hypothesis of this test states that the performances of all bagging predictors are equivalent. If the Null Hypothesis is rejected, it does not determine which particular algorithms differ from one another. Because the test result does not show exactly where that significant difference occurs, a post-hoc Nemenyi test is required to calculate the "critical difference".

The post-hoc Nemenyi test is required for additional exploration of the differences between mean ranks to provide specific information on which mean ranks are significantly different from other ranks. The critical difference is calculated. If the mean ranks are different by at least the critical difference, the performance of learners is statistically significantly different. More detail how to calculate the critical difference of the Nemenyi test is presented in Chapter 2 Subsection 2.3.2.

Table 7.2 presents the ranking order of the performance of G_{mean}. The first and 11th rows present the ascending order of the name of bagging predictors according to their mean rank of the G_{mean} measure in the 10th and 20th rows. The second to ninth and the 12th to 19th rows present the ranking order of the bagging predictors on each individual medical data-set, e.g., bagging MLP performs best on the *Diabetes* data-set ranking as 1, followed by bagging NBTree ranking as 2, and bagging OneR ranked 12 is the worst bagging predictor on the same data-set. The 10th and 20th rows present the mean rank of the G_{mean} metric of the bagging predictors over all eight medical data-sets. On the other hand, we observe that different bagging predictors behave differently for different medical data-sets, e.g., bagging MLP performs well on most of these medical data-sets, except for *Sick* data-set which is an extremely imbalanced, high dimensional, and

Table 7.2: Ranking order of the performance of bagging based on G_{mean} and mean ranks.

G_{mean}	MLP	NB	NBTree	SVM	PART	RdTree
Breastc	2	1	7	6	5	8
Diabetes	1	3	2	8	7	5
Sick	10	9	5	12	3	7
Heart-c	3	2	4	1	5	7
Heart-h	3	1	4	2	5	6
StaHeart	3	1	4	2	5	7
WBreastc	3	1	2	4	6	5
WDBC	2	10	4	1	3	6
Mean Rank	3.375	3.5	4	4.5	4.875	6.375

	J48	RepTree	DStump	KNN	DTable	OneR
Breastc	9	11	3	4	12	10
Diabetes	4	6	10	11	9	12
Sick	1	4	2	11	8	6
Heart-c	6	8	10	11	9	12
Heart-h	10	11	7	8	12	9
StaHeart	10	6	12	9	8	11
WDBC	7	8	11	5	9	12
Mean Rank	6.75	7.875	8.375	8.375	9.625	10.375

large data-set; bagging NB performs best (ranking as 1) on four medical data-sets, *Breastc*, *StatlogHeart*, *Heart-h* and *WBreastc*, but performs poorly on the other two data-sets, *Sick* and *WDBC*, which have high dimensional attributes or extremely imbalanced class distribution data-sets; while bagging *J*48 and DStump perform well on the extremely imbalanced, high dimensional, and largest medical data-set, *Sick*.

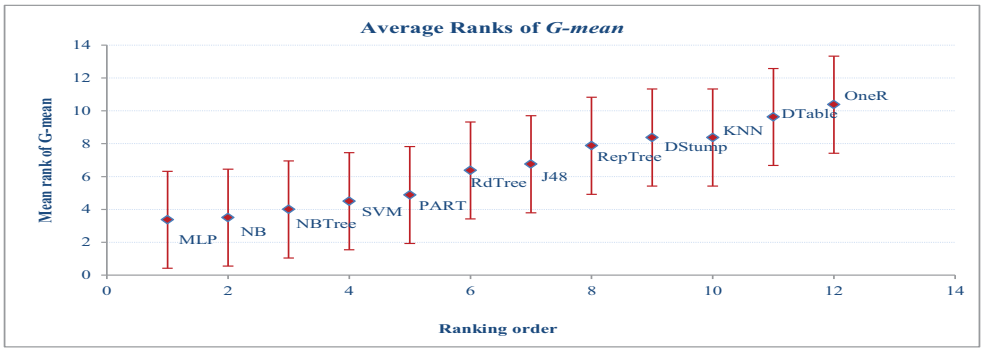

Figure 7.2: Comparison of the G_{mean} performance of all bagging predictors with post-hoc Nemenyi test, where the x-axes indicate the mean rank of each bagging predictor, the y-axes indicate the ascending ranking order of the bagging predictors, and the vertical error bars indicate the "critical difference".

Figure 7.2 presents the results of the mean rank of G_{mean} metric of bagging predictors over all eight medical data-sets based on the Friedman and post-hoc Nemenyi tests. The results indicate that bagging MLP and NB are the best bagging predictors, while bagging OneR is the worst bagging predictor. The performances of bagging predictors are statistically significantly different if the vertical bars do not overlap; therefore, there is a statistically significant difference between the two best bagging predictors, MLP and NB with the worst bagging predictor OneR. However, there is not a statistically significant difference between the remaining bagging predictors.

7.4.3 Comparison of the Performance of Prediction Models on Individual Medical Data-sets

Subsection 7.4.3 graphically compares the performance of the 24 prediction models on eight individual medical data-sets.

Figures 7.3 to 7.10 inclusive present a graphical comparison of the performance of the 24 prediction models on eight individual medical data-sets. Each graph presents the summarization of the observed performance of the prediction models based on four evaluation metrics, G_{mean}, TPR, TNR and Acc on individual data-sets. For each plot, the horizontal axis indicates the descending ranking order of the G_{mean} metric, while the vertical axis indicates the value of the four performance measures.

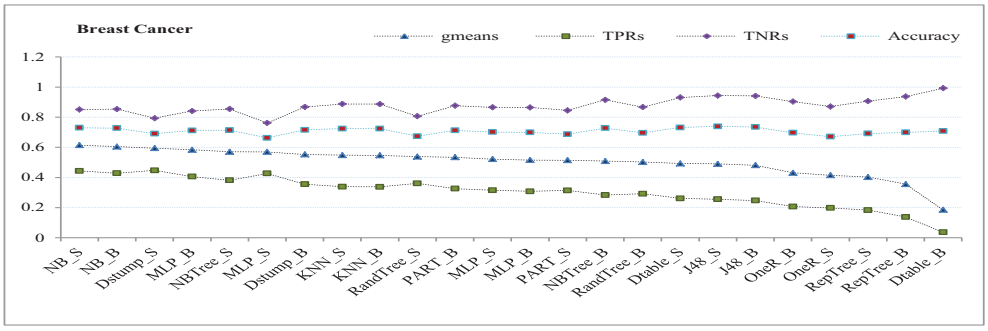

Figure 7.3: The performance of prediction models on *Breastc* data-set.

Figure 7.3 shows that both single learner NB and bagging NB perform better than the other prediction models, followed by the simple learner DStrump and bagging MLP. By contrast, bagging DTable, bagging RepTree, RepTree, OneR and bagging OneR are the worst prediction models for the *Breastc* data-set based on the evaluation metrics, G_{mean} and TPR. Even though the performance of Acc seems reasonably good for all the prediction models, but it is influenced by the TNR, this observation is consistent with the existing research.

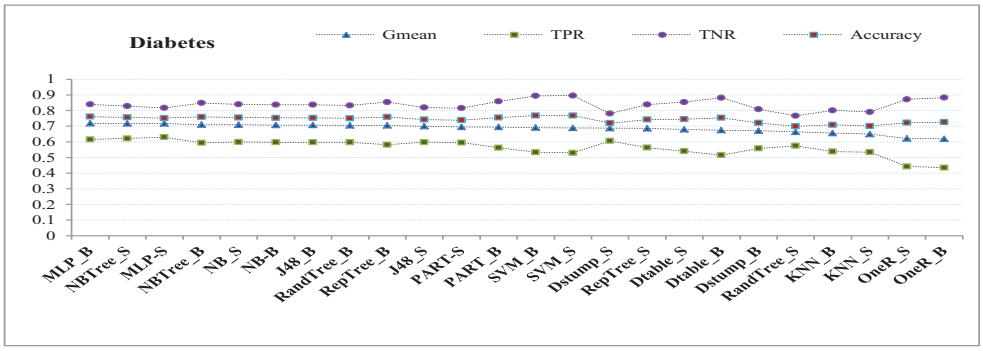

Figure 7.4: Comparison of the performance of prediction models on *Diabetes* data-set.

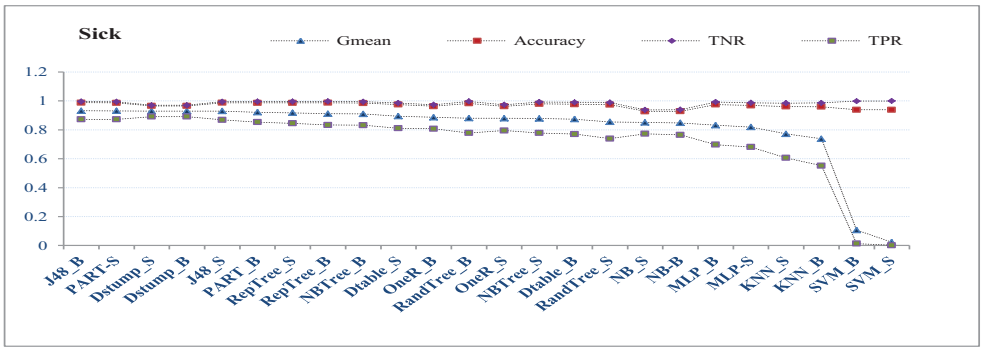

Figure 7.5: Comparison of the performance of prediction models on *Sick* data-set.

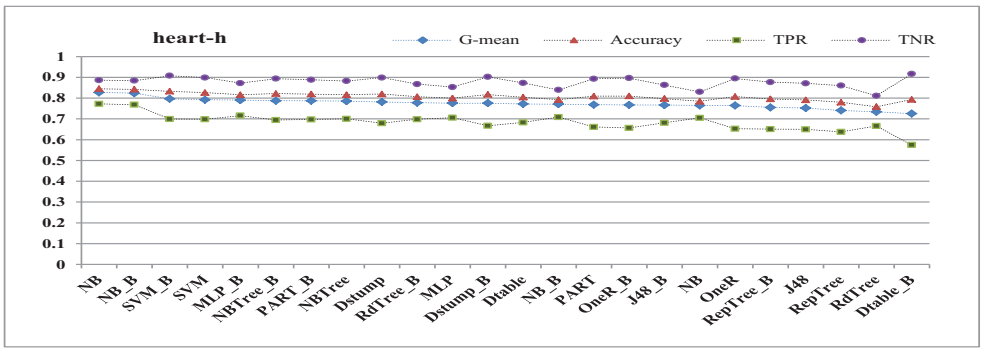

Figure 7.6: Comparison of the performance of bagging predictors and single learners on *Heart-h* data-set.

Figure 7.4 presents the comparison of the performance of the prediction models on the *Diabetes* data-set. Bagging MLP, NBTree, MLP and bagging NBTree are the best prediction models on this data-set, followed by NB and bagging NB. By contrast, bagging KNN, KNN, OneR and bagging OneR are the worst prediction models on this data-set.

Figure 7.5 presents a comparison of the performance of the prediction models on the extremely imbalanced *Sick* data-set. We observe that *Acc* and TNR perform well for all the prediction models, because *Acc* is influenced by TNR on this extremely imbalanced data-set. However, regarding the performance measures, TPR and G_{mean}, we observe that bagging $J48$ and PART perform best, followed by single DStump, bagging DStump, $J48$ and bagging PART. By contrast, SVM, bagging SVM, bagging KNN and KNN are the worst prediction models for this medical data-set.

Figure 7.6 presents a comparison of the performance of prediction models on the almost balanced *Heart-h* data-set. Most prediction models perform well on this data-set, e.g., NB, bagging NB, bagging SVM, and SVM are the best prediction models on this data-set.

Figure 7.7 presents a comparison of the performance of prediction models on the moderately imbalanced $WDBC$ data-set. Most prediction models perform well on this data-set, except for the weak learners, DStump, OneR, bagging OneR, and bagging DStump. By contrast, bagging SVM, bagging MLP and MLP are the best prediction models on this data-set.

Figure 7.8 presents a comparison of the performance of prediction models on *Heart-c* data-set. Bagging SVM, bagging NB, and NB are the best prediction models on this data-set, followed by SVM and bagging MLP. By contrast, weak learners, OneR and Dstump are the worst prediction models on this data-set.

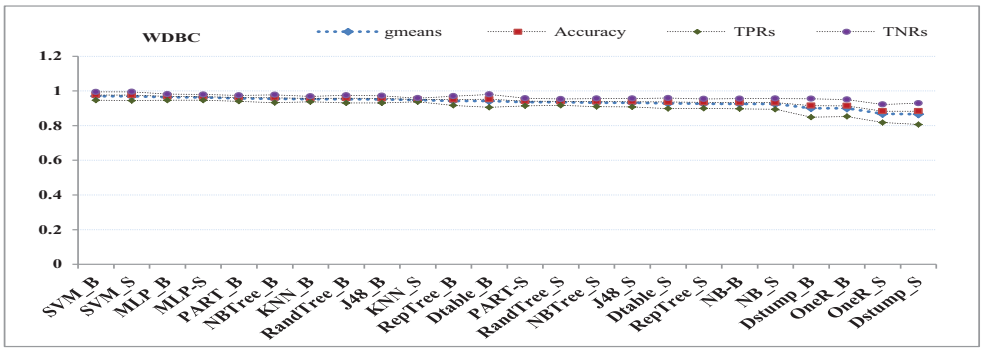

Figure 7.7: Comparison of the performance of the bagging predictors and single learners on $WDBC$ data-set.

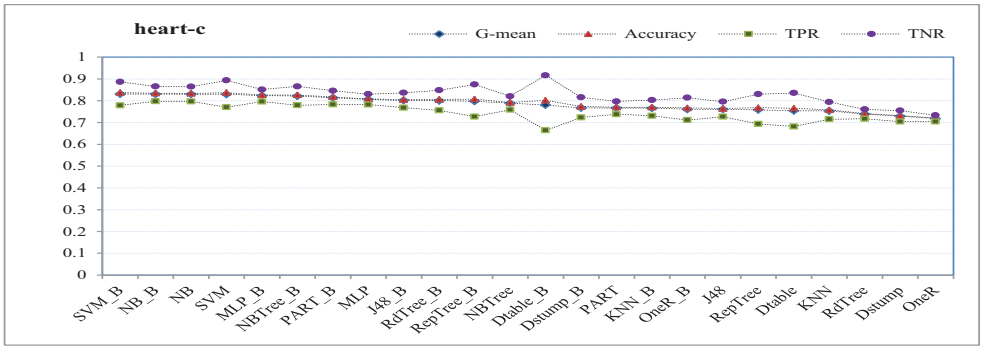

Figure 7.8: Comparison of the performance of the bagging predictors and single learners on *Heart-c* data-set.

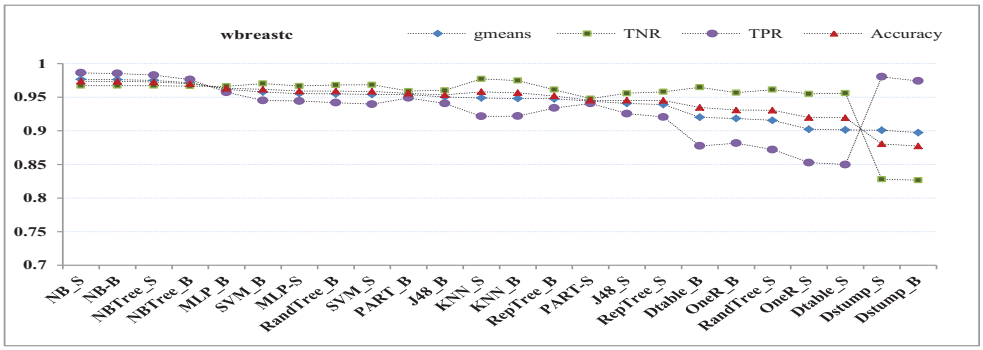

Figure 7.9: Comparison of the performance of the bagging predictors and single learners on $WBreastc$ data-set.

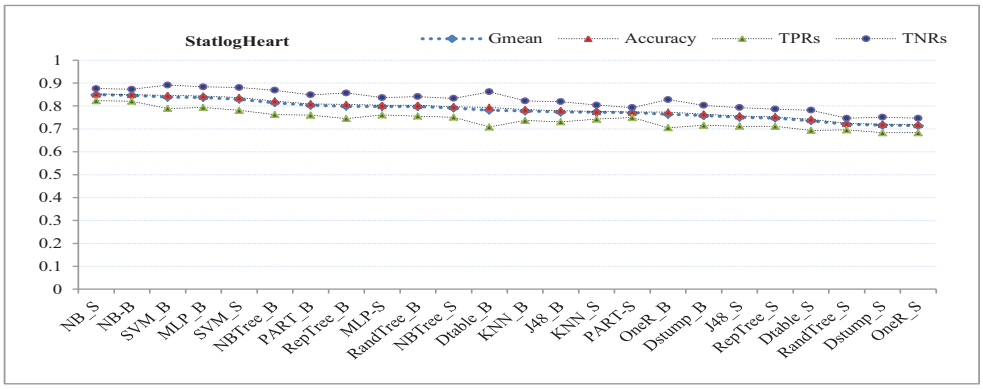

Figure 7.10: Comparison of the performance of the bagging predictors and single learners on *StatlogHeart* data-set.

Figure 7.9 presents a comparison of the performance of the prediction models on the *WBreastc* data-set. NB, bagging NB, NBTree, and bagging NBTree are the best prediction models on this data-set, followed by bagging MLP. By contrast, Dstump and bagging Dstump are the worst prediction models on this data-set.

Figure 7.10 presents a comparison of the performance of prediction models on the almost balanced *StatlogHeart* data-set. NB and bagging NB are the best prediction models on this data-set, followed by bagging SVM and bagging MLP. By contrast, weak learners, OneR and Dstump are the worst prediction models on this data-set.

Table 7.3 reports the best prediction models for the natural class distribution on individual medical data-sets. Bagging predictors MLP and *J*48 are the best prediction models for *Diabetes*, and *Sick* data-sets, respectively; moreover, bagging SVM is the best prediction model for *Heart-c* and *WDBC* data-sets; by contrast, single learner NB is the best prediction model for *Heart-h*, *StatlogHeart*, *WBreastc* and *Breastc* data-sets.

Table 7.3: Best G_{mean} performance prediction models for the natural class distribution on individual medical data-sets

Data-sets		Best Performance Models					
Index	Name	G_{mean}	ErrorRate	TPR	TNR	Learners	P%
1	Breastc	0.6142	0.2703	0.4435	0.8507	NB	0.29
2	Diabetes	0.7188	0.2384	0.6153	0.84	MLP_B	0.34
3	Heart-c	0.831	0.1624	0.779	0.8867	SVM_B	0.45
4	Heart-h	0.8239	0.1578	0.7679	0.8840	NB	0.36
5	StatlogHeart	0.8492	0.1474	0.8233	0.876	NB	0.44
6	Sick	0.932	0.0117	0.9959	0.8723	J48_B	0.06
7	WBreastc	0.9767	0.0262	0.9672	0.9863	NB	0.34
8	WDBC	0.97	0.0236	0.9462	0.9944	SVM_B	0.37

Table 7.4 presents the best achieved bagging prediction models with the altered class distribution on individual medical data-sets. Bagging RdTree performs best on four out of eight medical data-sets, *Diabetes*, *Heart-c*, *Heart-h*, and $WDBC$. Bagging NBTree performs best on three of eight medical data-sets, *Sick*, $WBreastc$, and $WDBC$. Bagging MLP performs best on two of eight medical data-sets, $StatlogHeart$ and $WDBC$. Bagging KNN performs best on $Breastc$ data-set. Bagging predictors, $PART$, $NBTree$, $RdTree$, and MLP perform best with G_{mean} metric, 0.979 on $WDBC$ data-set.

Table 7.4: The best G_{mean} performance of the bagging prediction models achieved with altered class distribution on individual data-sets

Data-sets		Best Performance Model				
Index	Name	G_{mean}	TPR	TNR	Learners	P%
1	Breastc	0.802	0.782	0.822	KNN_B	0.50
2	Diabetes	0.876	0.844	0.909	RdTree_B	0.40
3	Heart-c	0.913	0.905	0.921	RdTree_B	0.50
4	Heart-h	0.904	0.873	0.935	RdTree_B	0.40
5	StatlogHeart	0.912	0.883	0.941	MLP_B	0.40
6	Sick	0.974	0.964	0.984	NBTree_B	0.30
7	WBreastc	0.983	0.983	0.984	NBTree_B	0.30
8	WDBC	0.979	0.977	0.981	PART_B	0.40
		0.979	0.976	0.981	NBTree_B	0.50
		0.979	0.972	0.985	MLP_B	0.50
		0.979	0.971	0.986	RdTree_B	0.40

7.4.4 Comparison of the Performance of Bagging between Natural Class Distribution and Altered Class Distribution on Individual Medical Data-sets

Subsection 7.4.4 reports the performance of bagging predictors between the natural class distribution and the best achieved results by using sampling techniques on each medical data-set.

Tables 7.5 to 7.8 present a comparison of the performance of the bagging predictors between natural class distribution and the best achieved results of altered class distribution on eight medical data-sets. The first column indicates the name of a medical data-set and the bagging predictors, respectively; the second, fourth, and sixth columns present the results from the natural class distribution and the third, fifth, and seventh columns present the best achieved results from altered class distribution for evaluation metrics, G_{mean}, TPR, and TNR, respectively; the eighth column presents the proportion of the positive instances (P%), which refers to the level of the altered class distribution when bagging achieves the best performance on the G_{mean} measure. We also note that if the proportion of positive instances increases, the TPR will also increase but the G_{mean} may reduce.

The experimental results in the second and third columns indicate the comparison of G_{mean} performance of bagging predictors between natural class distribution and the best achieved results from altered class distribution: the level of the class distribution is mostly about 50% on *Breastc*, *Heart-c*, and *StatlogHeart* data-sets. This finding is consistent with previous research. However, the levels of class distribution are mostly 40% on *Diabetes*, $WDBC$ and *Heart-h* data-sets, 30% on *Sick* data-set, and 60% on $WBreastc$ data-set, when the best bagging performance on the G_{mean} measure is achieved.

Table 7.5: Comparison of the performance of bagging predictors on *Breastc* and *Heart-c* data-sets

Breastc	G_{mean}		TPR		TNR		
Bagging	Natural	Sampling	Natural	Sampling	Natural	Sampling	P%
J48	0.481	0.724	0.247	0.717	0.941	0.737	50%
RepTree	0.356	0.678	0.138	0.651	0.937	0.709	50%
RdTree	0.503	0.796	0.292	0.837	0.867	0.580	40%
NB	0.605	0.675	0.428	0.644	0.854	0.709	50%
SVM	0.516	0.690	0.308	0.695	0.865	0.685	50%
DStump	0.552	0.630	0.355	0.486	0.868	0.824	50%
OneR	0.431	0.619	0.207	0.505	0.904	0.769	50%
DTable	0.186	0.662	0.037	0.527	0.993	0.835	50%
PART	0.534	0.746	0.326	0.760	0.877	0.733	50%
KNN	0.546	0.802	0.338	0.782	0.887	0.822	50%
NBTree	0.509	0.731	0.284	0.732	0.915	0.732	50%
MLP	0.584	0.790	0.406	0.682	0.841	0.916	70%
Heart-c							
*J*48	0.801	0.887	0.768	0.880	0.836	0.894	50%
RepTree	0.798	0.850	0.728	0.82	0.875	0.881	50%
RdTree	0.801	0.913	0.756	0.905	0.849	0.921	50%
NB	0.831	0.840	0.798	0.802	0.866	0.882	40%
SVM	0.831	0.846	0.779	0.883	0.887	0.811	60%
DStump	0.768	0.778	0.723	0.745	0.816	0.812	40%
OneR	0.761	0.764	0.711	0.741	0.814	0.789	50%
DTable	0.780	0.847	0.665	0.790	0.917	0.908	50%
PART	0.814	0.906	0.783	0.896	0.846	0.917	50%
KNN	0.766	0.898	0.731	0.901	0.803	0.896	50%
NBTree	0.822	0.904	0.780	0.88	0.866	0.929	50%
MLP	0.823	0.909	0.796	0.907	0.852	0.911	70%

Table 7.6: Comparison of the performance of bagging predictors on *Heart-h* and *StatlogHeart* data-sets

Heart-h	G_{mean}		TPR		TNR		
Bagging	Natural	Sampling	Natural	Sampling	Natural	Sampling	P%
*J*48	0.767	0.855	0.681	0.854	0.863	0.856	50%
RepTree	0.756	0.828	0.650	0.881	0.877	0.779	60%
RdTree	0.778	0.904	0.698	0.873	0.868	0.935	40%
NB	0.824	0.837	0.768	0.808	**0.884**	**0.868**	50%
SVM	0.797	0.835	0.699	0.879	0.908	0.794	60%
DStump	0.776	0.796	0.667	0.741	0.902	0.859	60%
OneR	0.767	0.795	0.657	0.801	0.896	0.790	60%
DTable	0.725	0.830	0.575	0.805	0.917	0.857	50%
PART	0.787	0.867	0.697	0.828	0.888	0.908	40%
KNN	0.771	0.895	0.709	0.873	0.839	0.918	40%
NBTree	0.788	0.873	0.694	0.835	0.894	0.913	40%
MLP	0.790	0.893	0.716	0.860	0.872	0.928	40%
StatlogHeart							
*J*48	0.773	0.870	0.731	0.878	0.819	0.863	50%
RepTree	0.799	0.860	0.745	0.859	0.857	0.862	50%
RdTree	0.797	0.900	0.756	0.895	0.841	0.905	50%
NB	0.846	0.854	0.821	0.844	0.873	0.865	50%
SVM	0.839	0.865	0.789	0.853	0.891	0.877	50%
DStump	0.758	0.780	0.716	0.716	0.803	0.854	30%
OneR	**0.764**	**0.740**	0.705	0.726	0.828	0.756	50%
DTable	0.781	0.836	0.708	0.872	0.862	0.803	50%
PART	0.803	0.892	0.760	0.855	0.849	0.931	40%
KNN	0.778	0.891	0.737	0.863	0.822	0.919	40%
NBTree	0.814	0.900	0.763	0.856	0.869	0.946	40%
MLP	0.837	0.912	0.793	0.883	0.883	0.941	40%

Table 7.7: Comparison of the performance of bagging predictors on *Diabetes* and *Sick* data-sets

Diabetes	G_{mean}		TPR		TNR		
Bagging	Natural	Sampling	Natural	Sampling	Natural	Sampling	P%
*J*48	0.707	0.861	0.597	0.834	0.837	0.890	40%
RepTree	0.705	0.824	0.581	0.878	0.855	0.871	40%
RdTree	0.705	0.876	0.597	0.844	0.833	0.909	40%
NB	0.707	0.726	0.597	0.740	0.837	0.712	60%
SVM	0.691	0.741	0.534	0.700	0.894	0.785	50%
DStump	0.672	0.696	0.558	0.620	0.809	0.892	40%
OneR	0.620	0.719	0.435	0.723	0.883	0.716	50%
DTable	0.674	0.776	0.515	0.778	0.882	0.774	50%
PART	0.695	0.852	0.563	0.824	0.859	0.881	40%
KNN	0.656	0.848	0.538	0.816	0.801	0.881	40%
NBTree	0.710	0.840	0.593	0.803	0.849	0.879	40%
MLP	0.719	0.812	0.615	0.833	0.840	0.793	50%
Sick							
*J*48	0.932	0.973	0.872	0.967	0.996	0.979	30%
RepTree	0.912	0.965	0.834	0.954	0.997	0.976	30%
RdTree	0.881	0.972	0.778	0.964	0.997	0.980	40%
NB	0.848	0.880	0.765	0.864	0.939	0.898	20%
SVM	0.107	0.892	0.013	0.857	0.999	0.930	30%
DStump	0.930	0.934	0.892	0.896	0.970	0.974	70%
OneR	0.887	0.934	0.807	0.898	0.974	0.971	30%
DTable	0.874	0.941	0.771	0.902	0.991	0.982	30%
PART	0.922	0.973	0.854	0.967	0.995	0.979	30%
KNN	0.738	0.912	0.552	0.908	0.986	0.915	40%
NBTree	0.910	0.974	0.833	0.964	0.995	0.984	30%
MLP	0.832	0.951	0.698	0.964	0.993	0.938	50%

Table 7.8: Comparison of the performance of bagging predictors on $WDBC$ and $WBreastc$ data-sets

$WDBC$	G_{mean}		TPR		TNR		
Bagging	Natural	Sampling	Natural	Sampling	Natural	Sampling	P%
$J48$	0.951	0.974	0.931	0.968	0.972	0.981	40%
RepTree	0.943	0.968	0.917	0.959	0.970	0.978	40%
RdTree	0.953	0.979	0.931	0.971	0.975	0.986	40%
NB	0.926	0.937	0.897	0.905	0.956	0.970	10%
SVM	0.970	0.977	0.946	0.966	0.994	0.987	50%
DStump	0.900	0.925	0.849	0.940	0.955	0.910	70%
OneR	0.900	0.929	0.852	0.900	0.950	0.959	40%
DTable	0.942	0.958	0.906	0.961	0.980	0.954	40%
PART	0.957	0.979	0.940	0.977	0.975	0.981	40%
KNN	0.953	0.980	0.937	0.980	0.969	0.981	50%
NBTree	0.955	0.979	0.932	0.976	0.978	0.981	50%
MLP	0.964	0.979	0.947	0.972	0.982	0.985	50%
$WBreastc$							
$J48$	0.950	0.964	0.941	0.967	0.990	0.961	40%
RepTree	0.948	0.961	0.934	0.964	0.961	0.958	50%
RdTree	0.955	0.982	0.942	0.983	0.968	0.981	40%
NB	0.976	0.981	0.986	0.985	0.967	0.976	60%
SVM	0.958	0.979	0.945	0.988	0.971	0.970	80%
DStump	0.897	0.908	0.974	0.981	0.827	0.840	30%
OneR	0.918	0.935	0.882	0.962	0.957	0.908	60%
DTable	0.920	0.960	0.878	0.981	0.965	0.940	60%
PART	0.954	0.964	0.949	0.969	0.959	0.958	40%
KNN	0.948	0.981	0.922	0.982	0.975	0.981	60%
NBTree	0.972	0.983	0.976	0.983	0.967	0.984	30%
MLP	0.962	0.979	0.957	0.990	0.966	0.968	60%

In addition, there are interesting findings on both $WDBC$ and $Sick$ data-sets in that when bagging NB achieves the best performance on the G_{mean} measure, the levels of class distributions are 10% and 20%, respectively. Both data-sets have high dimensional features. This finding may be inconsistent with existing research, which assumes that traditional learning algorithms will perform better in a balanced situation than in an imbalanced situation.

The experimental results demonstrate that the sampling techniques can improve the performance of bagging predictors on the G_{mean} metric over most medical data-sets, except for bagging OneR on the $StatlogHeart$ data-set whose result is highlighted and marked in red. The bagging performance on the TPR and TNR measures also improves at the same level of class distribution, except for NB on *Heart-h* data-set whose TNR measure is highlighted and marked in red.

7.5 Conclusions

This research investigates the performance of bagging predictors with 12 underlying different base learners on eight medical data-sets. We address the imbalanced class distribution and unequal cost of mis-classification error issues on medical data which may have high accuracy but poor performance on the TPR of the minority class. We report the best performance prediction model for the natural class distribution on each individual medical data-set by comparing 12 single learners and 12 bagging predictors. In addition, we utilize sampling techniques to alter the class distribution at different imbalanced levels and report the comparison of the bagging performance between the natural class distribution and the best achieved performance based on the G_{mean} measure at a certain level of class distribution. We note that by using sampling techniques to improve the performance of the bagging predictors, the level of the class distribution is

mostly at a 50% balanced level for three data-sets, *Breastc*, *Heart-c*, and *statlogHeart*; however, it is mostly at 40% for the *Diabetes*, *WDBC* and *Heart-h* data-sets, at 30% for the *Sick* data-set, and at 60% for the *WBreastc* data-set. We also observe that the levels of class distribution for bagging NB to achieve the best performance on the G_{mean} measure are 10% for the *WDBC* data-set and 20% for the *Sick* data-set, while for bagging NBTree to achieve the best G_{mean} performance they are 30% for the *Sick* and *WBreastc* data-sets.

We investigated the effectiveness of bagging by using statistical tests. We also compared the performance of 12 bagging predictors on each of the medical data-sets; we observed that different bagging predictors behave differently for different medical data-sets. Bagging MLP performs well on most of these medical data-sets, except for the extremely imbalanced class distribution and high dimensional attributes large data-set *Sick*; Bagging NB has the best performance on four out of eight medical data-sets but performs poorly on two medical data-sets: *Sick* and *WDBC*; Bagging *J*48 and Dstump perform well on the extremely imbalanced and large, high dimensional *Sick* data-set. This full comparison of the performance of bagging predictors will allow data mining practitioners to choose the most favorable learners and to understand what to expect when using bagging predictors for imbalanced medical applications.

Chapter 8

An Effective Method for Imbalanced Time Series Classification: Hybrid Sampling

This chapter investigates effective methods for highly imbalanced time series classification problems. Mining time series data and imbalanced data are two of ten challenging problems in data mining research. Most traditional supervised classification learning algorithms are ineffective for highly imbalanced time series classification, which has received considerably less attention than imbalanced data problems in data mining and machine learning research. The structure-preserving over-sampling method (SPO) (Cao et al. 2011) has been proposed for solving highly imbalanced time series classification problems and has been reported as achieving better performance than other over-sampling methods and the state-of-the-art methods in a time series classification. The authors did not provide statistical analysis for their experimental results, and it can be argued that their claim is inappropriate from a statistical point of view. In addition, they did not compare their results with any under-sampling methods.

Bagging is one of the most effective ensemble learning methods, yet it has drawbacks on highly imbalanced data. Sampling methods are considered to be effective to tackle highly imbalanced data problem, but both over-sampling and under-sampling have disadvantages; thus it is unclear which sampling schema will improve the performance of bagging predictor for solving highly imbalanced time series classification problems. This chapter has addressed the limitations of existing techniques of the over-sampling and under-sampling, and proposes a new approach, hybrid-sampling technique to enhance bagging (HBagging), for solving these challenging problems. Comparing this new approach, HBagging with previous approaches, over-sampling methods, SPO, and under-sampling with various algorithms (Liang 2012) on benchmark data-sets, the experimental results demonstrate that HBagging is able to dramatically improve on the performance of previous approaches. Friedman and Post-hoc Nemenyi statistical tests are used to draw valid conclusions. Note: The over-sampling methods include repeating (REP), SMOTE (SMO) (Chawla et al. 2002), Borderline SMOTE (BoS) (Han et al. 2005), ADASYN (ADA) (He, Bai, Garcia & Li 2008), and DataBoost (DB) (Guo & Viktor 2004); and state-of-the-art methods in TSC include Easy Ensemble (Easy) (Liu et al. 2009), BalanceCascade (Bal) (Liu et al. 2009), One nearest neighbor classifier using Euclidean distance (1NN) (Wei & Keogh 2006), and One nearest neighbor classifier using dynamic time warping distance (1NN DW) (Xi, Keogh, Shelton, Wei & Ratanamahatana 2006).

The chapter is organized as follows. Section 8.1 gives an introduction. Section 8.2 outlines the proposed new approach. Section 8.3 provides the experimental setting and Section 8.4 presents the experimental results analysis. Section 8.5 concludes by summarizing the significant results of the chapter.

8.1 Introduction

Imbalanced time series classification (ITSC) involving time serial classification (TSC) and imbalanced problems can be widely observed in many real-world applications in various domains, such as financial stock market data analysis (Kim 2003), bio-informatics (Zavaljevski, Stevens & Reifman 2002), and ECG beats classification (Acır 2005). Most traditional supervised classification learning algorithms are ineffective for highly imbalanced time series classification (HITSC). Due to its challenging issues of high dimensionality, large scale, and uneven class distribution among different classes, and considering the sequence of the numerical attributes carrying special information as whole instead of individual attributes (Hidasi & Gáspár-Papanek 2011), it has received considerably less attention than imbalanced data problems in data mining and machine learning research. HITSC refers to a situation in which the proportions of the training examples of time series data are varied significantly among different classes. This study mainly focuses on imbalanced binary TSC, e.g., the proportion of positive examples that are far fewer than the proportion of negative examples in the training data of the TSC.

Bagging (Breiman 1996*a*) is one of several effective methods for classification, but it has limitations for solving highly imbalanced data problems. Sampling techniques are considered to be one of the most effective ways to tackle highly imbalanced problems, but since both over-sampling and under-sampling techniques have their limitations, it is unclear which sampling schema is able to enhance the performance of bagging. These challenging issues have motivated me to propose a new approach, hybrid-sampling techniques to enhance bagging (HBagging), for solving HITSC problems.

The proposed new approach, HBagging, randomly over-samples the positives and under-samples the negatives to half of the original training size, $\frac{|P|+|N|}{2}$, respectively, to generate a set of balanced bootstrap samples from the original training set to enhance bagging. Comparing the performance of this new approach, HBagging with previous approaches, the over-sampling methods and state-of-the-art methods in TSC, SPO (Cao et al. 2011), and under-sampling with various algorithms (Liang & Zhang 2012*a*) on the benchmark data-sets, the experimental results demonstrate that the proposed new approach, HBagging is superior to, and dramatically improves the performance of previous approaches. Friedman and post-hoc Nemenyi statistical tests for comparing the performance of multiple learning methods over multiple benchmark data-sets are applied to draw valid conclusions.

The key contributions of this chapter are as follows. (1) This chapter proposes a new approach, HBagging for improving the performance of prediction models to solve the HITSC problems. (2) Empirically comparing the performance of this new approach with previous approaches (Cao et al. 2011, Liang & Zhang 2012*b*, Liang & Zhang 2012*a*) on the benchmark data-sets, the experimental results demonstrate that the new approach, HBagging integrating the unstable base learner, decision trees *J*48, is effective for solving the HITSC problems and is dramatically superior to previous approaches.

8.2 HBagging approach

Algorithm 3 outlines the proposed new approach, HBagging integrating unstable learner decision trees *J*48. This new approach is different from previous approaches (Cao et al. 2011, Liang & Zhang 2012*a*) because H-sampling reduces the disadvantage of under-sampling, loosing to much

Algorithm 3: HBagging

Input:

D, original training set, containing $|P|$ positive
and $|N|$ negative instances;
a learning scheme (algorithm, e.g., $J48$);

Output: A composite model, C^*.

Method:

for $i = 1$ **to** k **do**

Create balanced bootstrap samples of
size $|D_i|$ sub-sets, $|D_i| = |P_i| + |N_i|$ where
P_i and N_i are randomly drawn with replacement
from original training set, P and N, respectively:
$|P_i| = |N_i| = \frac{(|P|+|N|)}{2}$ and;

end

return *a set of bootstrap samples D_i (containing k bootstrap samples);*

Train each base classifier model C_i from D_i;

To use the composite model, C^* for a test set T on an instance x
where its true class label is y:

$C^*(x) = \arg\max_y \sum_i \delta\left(C_i(x) = y\right)$
Delta function $\delta(\cdot) = 1$ if argument is true, else 0.

important information for training, and the disadvantages of over-sampling, over-fitting, high computational cost and longer training time. This new approach, H-sampling, randomly selects the positives and the negatives to the balanced point at half of the original training size, $\frac{|P|+|N|}{2}$. For example, the positives are randomly selected with replacement from the entire positive class to the size of the balanced point; the negatives are randomly selected with replacement from the negative class of original training set to the size of the balanced point.

For the proposed prediction model, suppose the size of an ensemble is k, a set of classifiers C_i (for $i = 1$ to k) is built from a set of balanced bootstrap samples D_i; each new test example is classified by a set of classifiers C_i, and the final prediction is made by majority votes to aggregate the predictions of the set of classifiers C_i by using a delta function $\delta(\cdot) = 1$ if the prediction of C_i is a true class label, else the delta function $\delta(\cdot) = 0$. Majority votes, aggregating the set of predicted class labels, use the delta function to vote for a class and the class label obtaining the highest number of votes is considered as the output of the final prediction.

8.2.1 Statistical Tests

Friedman and post-hoc Nemenyi statistical tests are applied to compare the performance of the multiple learning methods on multiple data-sets, where it is inappropriate to compare their average value, because the average values are susceptible to outliers (Demšar 2006, Liang & Zhang 2012*a*). Therefore, average rank is preferred for evaluating the performance of multiple learning methods. This work therefore performs statistical tests to evaluate the performance of the multiple learning methods on multiple data-sets. The Friedman test is utilized to obtain the average rank of the performance of the multiple learning methods on multiple data-sets; the

post-hoc Nemenyi test is utilized to check whether there is a statistically significant difference between the learning methods at a 95% confidence interval.

8.3 Experimental Setup

Java platform is used to implement the new approach, HBagging technique integrated unstable learner, decision trees J48 (Witten & Frank 2005), and to investigate the performance of the new approach and previous approaches. 31 bootstrap samples are used in the ensemble. A 10-trial 10-fold cross-validation evaluation is performed for this study. The Friedman test is used for the calculation of average rank.

Table 8.1: Time series data-sets

Data-sets		Data Information					Class Information	
		TS	Instances			Ratio	Previous	Altered
Index	Name	Length	(P^+&N^-)	P^+	N^-	P^+/N^-	Class	Class
1	Adiac	176	781	23	758	0.0303	37	2
2	S-Leaf	128	1125	75	1050	0.0714	15	2
3	Wafer	152	7164	762	6402	0.0119	2	2
4	FaceAll	131	2250	112	2138	0.0524	14	2
5	Yoga	426	3300	1530	1770	0.8644	2	2

8.3.1 Data-sets

Table 8.1 shows a summary of the characteristics of the five time series data-sets from the public UCR time series repository (Keogh, Zhu, Hu, Hao, Xi, Wei & Ratanamahatana 2011), which were used as the benchmark data-sets.

These data-sets were selected using different criteria, for instance the number of instances from 781 up to 3300, the number of attributes from 128 up to 426, and the frequency of each class from almost balanced 0.86 to extremely imbalanced 0.0303. The first column indicates the ID number and the name of the data-sets. The second column presents the information about the original time series data which includes the number of instances, the TS Length referring to the number of attributes (excluding the class) and the number of classes in each data-set, respectively. The third column presents the information of the altered imbalanced time series data-sets, which includes the number of positive class samples (P+), negative samples (N-), ratio of positive samples with negative samples, and the number of classes in each data-set.

We altered three out of five data-sets from multi-class change to binary-class as follows. For the Adiac data-set, the second class with 23 samples is considered as the positive class, and the remaining samples are considered as the negative class. For FaceAll and S-Leaf data-sets, the first class is considered as the positive class with 112 and 75 samples, respectively.

8.4 Experimental Results Analysis

This section contains three sub-sections: Subsection 8.4.1 evaluation of the performance of SVM on HITSC, Subsection 8.4.2 comparison of the performance of over-sampling methods, under-sampling with various algorithms and HBagging method on HITSC; and Subsection 8.4.3 comparison of the performance of other state-of-the-art methods in TSC, SPO, under-sampling with various algorithms, and HBagging method for HITSC.

8.4.1 Evaluation of the Performance of SVM

Table 8.2: Results of SVM on imbalanced time series data-sets

Data-sets		Results of SVM on ITSC data-sets (without sampling method)					
Index	Name	Error Rate	TPR	TNR	F_{value}	G_{mean}	Ratio P+/N-
1	Adiac	0.029	0	1	0	0	0.030
2	FaceAll	0.014	.769	0.997	0.845	0.876	0.052
3	S-Leaf	0.067	0	1	0	0	0.071
4	Wafer	0.037	0.666	0.998	0.792	0.815	0.012
5	Yoga	0.423	0.538	0.611	0.541	0.573	0.864

Table 8.2 shows the performance of SVM (using SMO from WEKA [26]) to examine whether SVM performs well for imbalanced time series classification. The experimental results indicate that SVM does not perform well on four out of five imbalanced time series classification data-sets. In the literature, overall estimated error rate is considered as an ineffective evaluation measure for imbalanced classification (Liang et al. 2011*b*, Maloof 2003, Qin 2005, Chawla et al. 2003, Weiss 2004, Sun et al. 2007).

Both data-sets Adiac and S-Leaf, for example, have low error rate (or high overall accuracy), however, their true positive rate and true negative rate are 0 and 1, respectively. This means that none of the minority class samples have been correctly predicted and all of majority class samples have been correctly predicted. These experimental results demonstrate that estimated error rate is an ineffective measure for HITSC. Moreover, SVM is not a suitable learning algorithm for HITSC, which is true for these HITSC, such as Adiac and S-Leaf data-sets. In addition, SVM is not suitable for large and high dimensional almost balanced TSC, such as Yoga data-set.

8.4.2 Comparison of Over-sampling, Under-sampling, and Hybrid-sampling Methods

Table 8.3: Comparison of the performance of over-sampling methods, under-sampling with various algorithms, and HBagging method based on the evaluation metrics F_{value} and G_{mean}.

Metrics	Data-set	Results from Previous Research (Cao et al. 2011)						Results from previous Work (Liang & Zhang 2012*a*)					This Work
		Over-sampling Methods						Under-sampling					H-sampling
	Name	REP	SMO	BoS	ADA	DB	SPO	SVM	J48	RTree	KNN	MLP	HBagging
F_{value}	Adiac	0.375	0.783	0.783	0.783	0.136	0.963	0.967	0.883	0.903	0.918	0.947	**0.975**
	S-Leaf	0.761	0.764	0.764	0.759	0.796	0.796	0.841	0.820	0.849	0.836	0.786	**0.932**
	Wafer	0.962	0.968	0.968	0.967	0.977	0.982	0.891	0.929	0.956	**0.999**	0.933	0.980
	FaceAll	0.935	0.935	0.935	0.935	0.890	0.936	0.957	0.876	0.863	0.909	0.919	**0.995**
	Yoga	0.710	0.729	0.721	0.727	0.689	0.702	0.744	0.771	0.811	0.807	0.780	**0.926**
	AverageValue	0.740	0.836	0.834	0.834	0.698	0.876	0.880	0.856	0.876	0.894	0.873	**0.962**
	STD	0.236	0.108	0.110	0.109	0.332	0.122	0.110	0.061	0.055	0.075	0.083	**0.031**
	AverageRank	8.90	6.90	7.30	7.70	8.70	4.50	7.40	7.80	6.40	4.40	6.60	**1.40**
	CD	7.45											
G_{mean}	Adiac	0.480	0.831	0.831	0.831	0.748	**0.999**	0.957	0.910	0.920	0.958	0.975	0.989
	S-Leaf	0.800	0.861	0.861	0.849	0.898	0.898	0.902	0.809	0.812	0.887	0.856	**0.976**
	Wafer	0.965	0.969	0.970	0.970	0.980	0.984	0.903	0.907	0.956	**0.998**	0.937	0.988
	FaceAll	0.950	0.950	0.950	0.950	0.948	0.957	0.966	0.870	0.860	0.929	0.925	**0.997**
	Yoga	0.741	0.756	0.750	0.755	0.724	0.735	0.630	0.807	0.803	0.808	0.774	**0.976**
	AverageValue	0.787	**0.783**	0.872	0.871	**0.860**	**0.915**	0.872	0.861	0.870	0.916	0.893	**0.985**
	STD	0.197	0.088	0.090	0.089	0.117	0.108	0.138	0.051	0.067	0.073	0.079	**0.009**
	AverageRank	9.30	6.80	6.90	7.20	7.50	4.10	6.60	8.60	8.20	4.20	7.20	**1.40**
	CD	7.45											

Table 8.3 presents a comparison of the performance of this new approach, HBagging with previous approaches, over-sampling methods and the under-sampling with various algorithms based on the F_{value} and G_{mean} measures. The experimental results indicate that this new approach, HBagging achieves the best performance with F_{value} across all over-sampling methods and the under-sampling with various algorithms on average value and average rank of F_{value}. This new approach achieves the

highest average value 0.962 with smallest standard deviation (STD) 0.031 and the best average rank 1.4, respectively, which are the best results across all methods; while KNN with the under-sampling method achieves the average value 0.894 with STD 0.075 and average rank 4.40, respectively, which is the second best across all methods on F_{value}.

On average value and average rank of the G_{mean} measure, this new approach, HBagging achieves the highest average value 0.985 with smallest STD 0.009 and lowest average rank 1.40, respectively, which is the best across all the compared methods; while, the SPO over-sampling method achieves average value 0.915 with STD 0.108 and average rank 4.1, respectively, which is the second best across all the compared methods on average rank of the G_{mean} measure, whereas KNN with the under-sampling method achieves average value 0.916 with STD 0.073 and average rank 3.4, respectively, which is the second best across all the compared methods on average of the G_{mean} measure. The results highlighted in red indicate the correction of the previous work (Cao et al. 2011, Liang & Zhang 2012*a*).

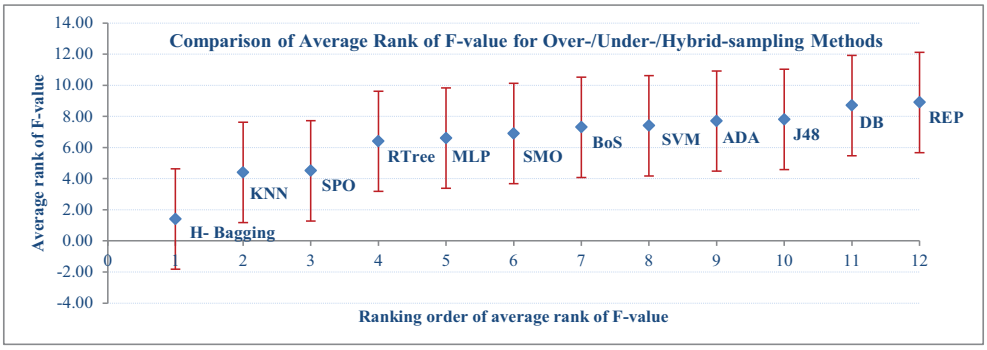

Figure 8.1: Comparison of average rank of the F_{value} with the Nemenyi test for the over-sampling methods, under-sampling with with various algorithms, and HBagging, where the x-axis indicates the ranking order of the average rank of the F_{value}, the y-axis indicates the average rank of the F_{value}, and the vertical bars indicate the "critical difference".

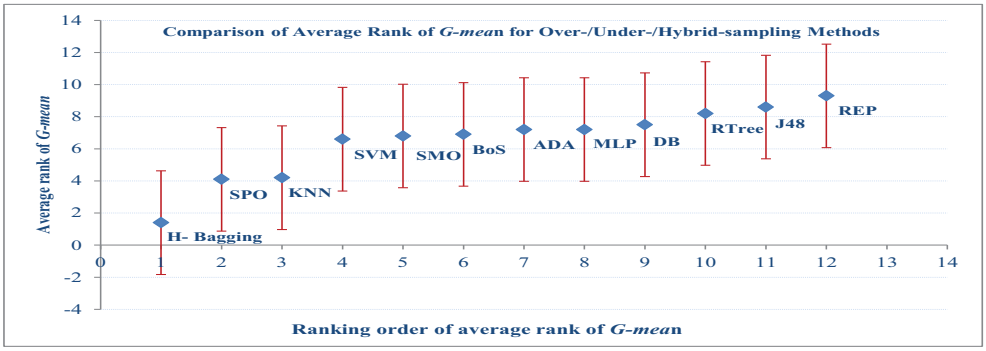

Figure 8.2: Comparison of average rank of the G_{mean} with the Nemenyi test for all the over-sampling methods, under-sampling with various algorithms, and HBagging method, where the x-axis indicates the ranking order of the average rank of the G_{mean}, the y-axis indicates the average rank of the G_{mean}, and the vertical bars indicate the "critical difference".

Figs 8.1 and 8.2 present a comparison of this new approach, HBagging with previous approaches, over-sampling and under-sampling with various algorithms, with the Nemenyi test, where the x-axis indicates the ranking order of the average rank of two evaluation measures, the y-axis indicates the average rank of the F_{value} and G_{mean} performance, respectively, and the vertical bars indicate the "critical difference". Groups of sampling methods that are no significantly different at a 95% confidence interval are indicated when the vertical bars overlap. Comparing the performance of this new approach with previous approaches, over-sampling (Cao et al. 2011) and under-sampling with various algorithms (Liang et al. 2011*b*), based on F_{value} and G_{mean}, HBagging has the best average rank on both measures. KNN with the under-sampling method has the second best average rank of F_{value}; while the SPO over-sampling method has the second best average rank of G_{mean}. Statistical tests indicate that there is no statistically significant difference at a 95% confidence interval between over-sampling SPO, under-sampling KNN, and HBagging on the average rank of F_{vlaue} and G_{mean}; however, there

is a statistically significant difference at a 95% confidence interval between HBagging and two over-sampling methods, DB and REP on F_{vlaue} measure, and between HBagging and two over-sampling methods, $J48$ and REP on G_{mean} measure.

8.4.3 Comparison of the Performance of State-of-the-art Methods in TSC, SPO, Under-sampling, and H-sampling Methods

Table 8.4 presents a comparison of the performance of previous work (state-of-the-art methods in TSC, SPO (Cao et al. 2011), under-sampling with various algorithms (Liang & Zhang 2012a)), and this work, HBagging based on F_{value} and G_{mean} evaluation measures. The experimental results indicate that HBagging achieves the best performance on F_{value} and G_{mean} across all previous approaches. HBagging achieves an average value of 0.962 and 0.985, and average rank of 1.40 and 1.40, respectively, which is the best average value and average rank of F_{value} and G_{mean} across all previous methods. KNN achieves an average value of 0.894 and 0.916, and an average rank of 3.0 and 2.4, respectively, which is the second best average value and average rank of F_{value} and G_{mean} across all the remaining methods. The results highlighted in red indicate the correction of the previous work (Cao et al. 2011).

Figs. 8.3 and 8.4 present a comparison of the performance of previous work (state-of-the-art methods in TSC, SPO, and the under-sampling method with various algorithms) and this new approach, HBagging, using the Nemenyi test, where the x-axis indicates the ranking order of the average rank of two evaluation measures; the y-axis indicates the average rank of F_{value} and G_{mean} performance, respectively, and the vertical bars indicate the "critical difference". Groups of learning methods that have no statistically significant difference at a 95% confidence interval are indicated

Table 8.4: Comparison of the performance of state-of-the-art methods in TSC, SPO, under-sampling with various algorithms, and HBagging based on evaluation metrics: F_{value} and G_{mean}.

Metrics	Data-set	Results from Previous Research (Cao et al. 2011)					Results from Previous Work (Liang & Zhang 2012*a*)					This Work
		state-of-the-art methods in TSC					Under-sampling					H-sampling
	Name	Easy	Bal.	1NN	1NN_DW	SPO	SVM	J48	RTree	KNN	MLP	HBagging
F_{value}	Adiac	0.534	0.348	0.800	0.917	0.963	0.967	0.883	0.903	0.918	0.947	**0.975**
	S-Leaf	0.521	0.578	0.716	0.429	0.796	0.841	0.820	0.849	0.836	0.786	**0.932**
	Wafer	0.795	0.954	0.949	0.857	0.982	0.891	0.929	0.956	**0.999**	0.933	0.980
	FaceAll	0.741	0.625	0.802	0.959	0.936	0.957	0.876	0.863	0.909	0.919	**0.995**
	Yoga	0.356	0.689	0.652	0.710	0.702	0.744	0.771	0.811	0.807	0.780	**0.926**
	AverageValue	0.589	0.639	0.784	**0.774**	0.876	0.880	0.856	0.876	0.894	0.873	**0.962**
	STD	0.179	0.218	0.112	0.215	0.122	0.092	0.061	0.055	0.075	0.083	**0.031**
	AverageRank	10.4	9	8.4	7.2	4.6	4.6	6.6	4.6	3.8	5.4	**1.4**
	CD	7.00										
G_{mean}	Adiac	0.782	0.897	0.875	0.920	**0.999**	0.957	0.910	0.920	0.958	0.975	0.989
	S-Leaf	0.721	0.898	0.798	0.572	0.898	0.902	0.809	0.812	0.887	0.856	**0.976**
	Wafer	0.817	0.970	0.953	0.870	0.984	0.903	0.907	0.956	**0.998**	0.937	0.988
	FaceAll	0.792	0.918	0.983	0.985	0.957	0.966	0.870	0.860	0.929	0.925	**0.997**
	Yoga	0.464	0.688	0.695	0.741	0.735	0.630	0.807	0.803	0.808	0.774	**0.976**
	AverageValue	0.713	0.874	**0.861**	**0.818**	**0.915**	0.872	0.861	0.870	0.916	0.893	**0.985**
	STD	0.145	0.108	0.117	0.164	0.108	0.113	0.051	0.067	0.073	0.079	**0.009**
	AverageRank	10.80	6.50	7.20	7.10	3.50	7.10	7.20	6.50	3.20	5.50	**1.40**
	CD	7.00										

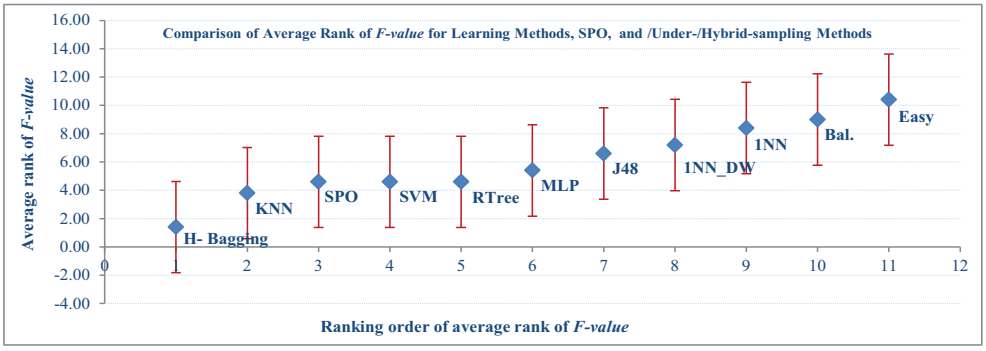

Figure 8.3: Comparison of average rank of the F_{value} metric with the Nemenyi test for the state-of-the-art methods in TSC, SPO, under-sampling with various algorithms, and HBagging, where the x-axis indicates the ranking order of the average rank of F_{value}, the y-axis indicates the average rank of F_{value}, and the vertical bars indicate the "critical difference".

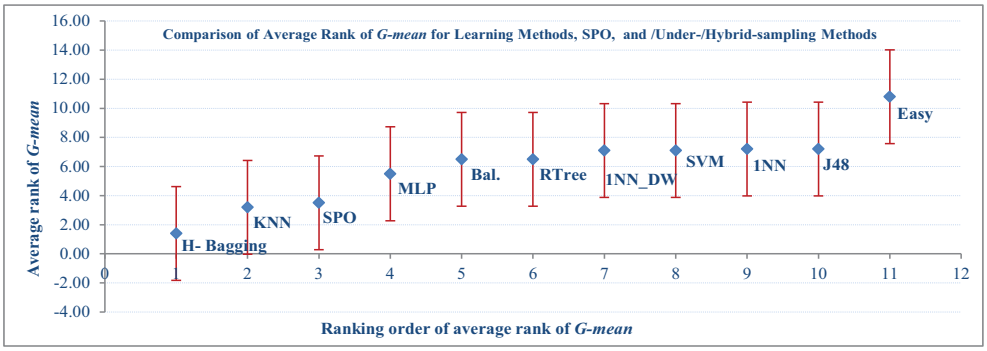

Figure 8.4: Comparison of average rank of the G_{mean} metric with the Nemenyi test for the state-of-the-art methods in TSC, SPO, under-sampling with various algorithms, and HBagging, where the x-axis indicates the ranking order of the average rank of G_{mean}, the y-axis indicates the average rank of G_{mean}, and the vertical bars indicate the "critical difference".

when the vertical bars overlap. Comparing the previous approaches (Cao et al. 2011, Liang & Zhang 2012*a*) and this approach, HBagging based on F_{value} and G_{mean}, H-sampling with bagging has the best average rank, and KNN with under-sampling method has the second best average rank. The statistical Nemenyi test results demonstrate that HBagging is statistically significantly better than 1NN, Bal. and Easy on F_{value}, and better than Easy on G_{mean} at a 95% confidence interval; however, there is no statistically significant difference between HBagging and previous approaches, SPO, and under-sampling KNN at a 95% confidence interval.

8.5 Conclusion

This chapter has proposed a new approach, H-sampling schema to enhance bagging for improving the performance of extremely imbalanced time series classification. This new schema reduces the computational cost and training time of over-sampling by using fewer positives in training, and increases the capability of under-sampling by using more negatives for training. We have empirically compared this new approach, HBagging with the previous approaches of over-sampling methods, state-of-the-art methods in TSC, SPO and under-sampling with various algorithms based on two evaluation measures, F_{value} and G_{mean} on benchmark data-sets. The experimental results demonstrate that HBagging dramatically improves the performance of the previous approaches. HBagging achieves the highest average value with the lowest STD and the lowest average rank on both evaluation measures, and it is dramatically superior to previous approaches on both evaluation measures. For future work, I would like to investigate the impact of the performance of HBagging with other base learners: unstable learners and stable learners.

Chapter 9

Conclusions and Future Work

9.1 Conclusions

This dissertation has proposed novel supervised ensemble learning approaches, UBagging and HBagging to boost the prediction model for solving extremely imbalanced classification and HITSC problems. It has also empirically investigated the performance of bagging predictors with respect to different algorithms on various types of data environments in terms of learning from natural class distribution, and in terms of learning from varying levels of class distribution approaches.

The novel supervised ensemble learning approach, UBagging, has been tested on 32 public data-sets, and the experimental results have demonstrated that UBagging is statistically significantly superior to Single$J48$, SBagging, and BBagging. HBagging has been tested on the benchmark data-sets of HITSC problems, and the experimental results have demonstrated that HBagging performs the best and is superior to the complex over-sampling SPO approach, to the under-sampling with various learning algorithms, and the state of the art methods in time series classification.

In terms of learning from natural class distribution, a comprehensive study was conducted to evaluate the performance of bagging with respect to 12 different algorithms on 48 data-sets. A novel two dimensional stability and robustness decomposition was proposed to rank base learners into different categories to assess the performance of the bagging predictors, which provided a clear picture of categorized base learners. The existing research mainly concerns the fact that instability is an important factor for bagging to improve the performance of a prediction model. This dissertation has asserted that both stability and robustness are important factors for ensuring that bagging predictors achieve high performance. The experimental results demonstrate that bagging is influenced by the combination of robustness and instability, and illustrates that robustness is important for bagging to achieve a highly accurate prediction model. Our observations support our claims: the most robust base learners, MLP, NBTree and PART contribute to the best bagging prediction models; in contrast, the weakest learners, OneR and DStump, lead to the worst bagging prediction models.

In terms of learning from varying levels of class distribution, the estimated error rate has been addressed as an ineffective evaluation metric for learning from the imbalanced class distribution problem; consequently, other evaluation metrics, such as AUC of ROC curve, TPR and G_{mean} have been adopted to investigate the performance of bagging predictors on imbalanced data as follows:

- The ROC curves present the average performance of bagging predictors at 9 different levels of class distribution. Based on AUC of ROC curve as an evaluation measure, the experimental results demonstrated that the AUC performance of bagging is statistically superior to that of single learners, except for SVM and DStump; when comparing the AUC performance of bagging predictors, bagging with the unstable

learners, DTable and RepTree, are the learning algorithms with the best bagging average performance, while bagging with the stable learners, NB, MLP and KNN, lead to the worst bagging average performance of bagging predictors.

- Based on TPR and G_{mean} as evaluation metrics, the experimental results demonstrate that robustness is important for bagging to achieve a highly accurate prediction model. For example, the most robust and unstable base learners, PART, MLP and NBTree, can be used to build the best bagging prediction models, whereas the weakest learners, OneR and DStump, result in the worst bagging prediction models.

- Consequently, this dissertation investigated the effect of varying levels of class distribution on the sensitivity of bagging, and the experimental results demonstrate that bagging MLP and bagging NB are the most insensitive predictors when the levels of class distribution vary.

This dissertation investigated the performance of bagging on medical data as follows:

- This work compared 24 prediction models, 12 bagging predictors and 12 single learners in terms of learning from natural class distribution based on four evaluation metrics, TPR, TNR, G_{mean} and Acc, and reported the best G_{mean} performance of prediction models on eight individual medical data-sets. The experimental results indicate that single learner NB performed best on four of eight medical data-sets; bagging SVM performed best on three of eight medical data-sets; and bagging $J48$ performed best on one of eight medical data-sets.

- This work compared the performances of bagging between natural class distribution and the altered class distribution, and reported the best achieved performance by using altered class distribution. It was

observed that when the best performance was achieved at certain levels of class distribution, which are varied depending on individual medical data-sets.

Two limitations of this study are that only the default parameter settings of 12 learning algorithms in WEKA are used, and only the empirical study is adopted for this investigation.

9.2 Future Work

Although a comprehensive empirical study has been performed in terms of learning from natural class distribution and altered class distribution, it is unclear whether different parameter settings would affect the experimental results and lead to different conclusions. In addition, although the proposed new approaches UBagging and HBagging with underlying unstable learner $J48$ perform well on imbalanced classification and ITSC problems, respectively, we did not investigate the impact of the performance of these approaches with other learners, such as KNN and MLP as base learners with UBagging on these imbalanced problems. We also did not provide theoretical justification for why these approaches perform well on imbalanced problems.

In future work, we would like to explore different parameter settings of algorithms in WEKA to identify whether they will affect the experimental results; we would also like to empirically investigate whether these novel approaches with other underlying learners are superior to the existing work on these imbalanced problems; as well as making theoretical investigations into why UBagging and HBagging approaches perform well on these imbalanced problems.

Bibliography

Acır, N. (2005), 'Classification of ECG beats by using a fast least square support vector machines with a dynamic programming feature selection algorithm', *Neural Computing & Applications* **14**(4), 299–309.

Banfield, R., Hall, L., Bowyer, K. & Kegelmeyer, W. (2007), 'A comparison of decision tree ensemble creation techniques', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(1), 173–180.

Batista, G., Prati, R. & Monard, M. (2004), 'A study of the behavior of several methods for balancing machine learning training data', *ACM SIGKDD Explorations Newsletter* **6**, 20–29.

Bauer, E. & Kohavi, R. (1999), 'An empirical comparison of voting classification algorithms: Bagging, boosting, and variants', *Machine Learning* **36**(1), 105–139.

Bradley, A. (1997), 'The use of the area under the ROC curve in the evaluation of machine learning algorithms', *Pattern Recognition* **30**(7), 1145–1159.

Breiman, L. (1996*a*), 'Bagging predictors', *Machine Learning* **24**(2), 123–140.

Breiman, L. (1996*b*), Bias, variance and arcing classifiers, Technical report, Department of Statistics, University of California.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Buciu, I., Kotropoulos, C. & Pitas, I. (2006), 'Demonstrating the stability of support vector machines for classification', *Signal Processing* **86**(9), 2364–2380.

Bühlmann, P. & Yu, B. (2002), 'Analyzing bagging', *The Annals of Statistics* **30**(4), 927–961.

Buja, A. & Stuetzle, W. (2000), 'Smoothing effects of bagging', *AT&T Labs-Research* .

Buja, A. & Stuetzle, W. (2006), 'Observations on bagging', *Statistica Sinica* **16**(2), 323.

Bunkhumpornpat, C., Sinapiromsaran, K. & Lursinsap, C. (2009), 'Safe-Level-SMOTE: Safe-Level-Synthetic Minority Over-Sampling TEchnique for handling the class imbalanced problem', *Advances in Knowledge Discovery and Data Mining* pp. 475–482.

Bunkhumpornpat, C., Sinapiromsaran, K. & Lursinsap, C. (2011), 'DB-SMOTE: Density-Based Synthetic Minority Over-sampling TEchnique', *Applied Intelligence* pp. 1–21.

Cao, H., Li, X., Woon, Y. & Ng, S. (2011), SPO: Structure preserving oversampling for imbalanced time series classification, *in* 'Proceeding of the IEEE 11th International Conference on Data Mining', pp. 1008–1013.

Chan, P. & Stolfo, S. (1998), Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection, *in* 'Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining', pp. 164–168.

Chawla, N. (2010), 'Data mining for imbalanced datasets: An overview', *Data Mining and Knowledge Discovery Handbook* pp. 875–886.

Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. (2002), 'SMOTE: Synthetic minority over-sampling technique', *Journal of Artificial Intelligence Research* **16**(1), 321–357.

Chawla, N., Japkowicz, N. & Kotcz, A. (2004), 'Editorial: Special issue on learning from imbalanced data sets', *ACM SIGKDD Explorations Newsletter* **6**, 1–6.

Chawla, N., Lazarevic, A., Hall, L. & Bowyer, K. (2003), SMOTEBoost: Improving prediction of the minority class in boosting, *in* 'Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases', pp. 107–119.

Chen, X. & Wasikowski, M. (2008), FAST: A ROC-based feature selection metric for small samples and imbalanced data classification problems, *in* 'Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 124–132.

Cherkauer, K. (1996), Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks, *in* 'Proceedings of the Working Notes of the AAAI Workshop on Integrating Multiple Learned Models', pp. 15–21.

Cieslak, D. (2010), Finding problems in, proposing solutions to, and performing analysis on imbalanced data, PhD thesis, University of Notre Dame.

Cieslak, D. & Chawla, N. (2008), 'Learning decision trees for unbalanced data', *Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computing Science* **5211**, 241–256.

Cochran, W. (1977), *Sampling Techniques*, 3rd edn, Wiley, New York.

Collobert, R., Bengio, S. & Bengio, Y. (2002), 'A parallel mixture of SVMs for very large scale problems', *Neural Computation* **14**(5), 1105–1114.

De Castro Dutra, I., Page, D., Santos Costa, V. & Shavlik, J. (2003), An empirical evaluation of bagging in inductive logic programming, *in* 'Proceedings of the 12th International Conference on Inductive Logic Programming', pp. 48–65.

Demšar, J. (2006), 'Statistical comparisons of classifiers over multiple data sets', *Journal of Machine Learning Research* **7**, 1–30.

Dietterich, T. (2000*a*), 'Ensemble methods in machine learning', *Proceedings of the First International Workshop on Multiple Classifier Systems* pp. 1–15.

Dietterich, T. (2000*b*), 'An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization', *Machine Learning* **40**(2), 139–157.

Dietterich, T. (2003), Ensemble learning, *in* M. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', The MIT Press, pp. 405–408.

Dietterich, T. & Bakiri, G. (1995), 'Solving multiclass learning problems via error-correcting output codes', *Journal of Artificial Intelligence Research* **2**(263), 286.

Domingos, P. (1999), Metacost: A general method for making classifiers cost-sensitive, *in* 'Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 155–164.

Domingos, P. (2000), A unified bias-variance decomposition for zero-one and squared loss, *in* 'Proceedings of the 17th National Conference on Artificial Intelligence', pp. 564–569.

Drummond, C., Holte, R. et al. (2003), C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling, *in* 'Workshop on Learning from Imbalanced Datasets II', Vol. 11.

Efron, B. & Tibshirani, R. (1993), *An Introduction to the Bootstrap*, Vol. 57, Chapman and Hall, New York.

Estabrooks, A., Jo, T. & Japkowicz, N. (2004), 'A multiple resampling method for learning from imbalanced data sets', *Computational Intelligence* **20**, 18–36.

Fawcett, T. (2004), 'ROC Graphs: Notes and practical considerations for researchers', *Machine Learning* **31**, 1–38.

Fawcett, T. (2006), 'An introduction to ROC analysis', *Pattern Recognition Letters* **27**, 861–874.

Frank, E. & Pfahringer, B. (2006), 'Improving on bagging with input smearing', *Advances in Knowledge Discovery and Data Mining* pp. 97–106.

Freund, Y. & Schapire, R. E. (1996), Experiments with a new boosting algorithm, *in* 'Proceedings of the 13th International Conference on Machine Learning', pp. 148–156.

Friedman, J. (1997), 'On bias, variance, 0/1 loss, and the curse-of-dimensionality', *Data Mining and Knowledge Discovery* **1**(1), 55–77.

Friedman, J. & Hall, P. (2007), 'On bagging and nonlinear estimation', *Journal of Statistical Planning and Inference* **137**(3), 669–683.

Goebel, M. (2004), Ensemble learning by data resampling, PhD thesis, ResearchSpace@ Auckland.

Guo, H. & Viktor, H. L. (2004), 'Learning from imbalanced data sets with boosting and data generation: The DataBoost-IM approach', *ACM SIGKDD Explorations Newsletter* **6**(1), 30–39.

Han, H., Wang, W. & Mao, B. (2005), 'Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning', *Advances in Intelligent Computing* pp. 878–887.

Hansen, L. & Salamon, P. (1990), 'Neural network ensembles', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**(10), 993–1001.

Hashem, S. (1997), 'Optimal linear combinations of neural networks', *Neural Networks* **10**(4), 599–614.

He, G., Han, H. & Wang, W. (2005), An over-sampling expert system for learning from imbalanced data sets, *in* 'Proceedings of the International Conference on Neural Networks and Brain', Vol. 1, pp. 537–541.

He, H., Bai, Y., Garcia, E. & Li, S. (2008), ADASYN: Adaptive synthetic sampling approach for imbalanced learning, *in* 'IEEE International Joint Conference on Neural Networks, IJCNN 2008', IEEE, pp. 1322–1328.

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

Hidasi, B. & Gáspár-Papanek, C. (2011), 'ShiftTree: An interpretable model-based approach for time series classification', *Machine Learning and Knowledge Discovery in Databases* pp. 48–64.

Hido, S., Kashima, H. & Takahashi, Y. (2009), 'Roughly balanced bagging for imbalanced data', *Statistical Analysis and Data Mining* **2**(5-6), 412–426.

Ho, T. (1998), 'The random subspace method for constructing decision forests', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(8), 832–844.

Hothorn, T. & Lausen, B. (2003), 'Double-bagging: Combining classifiers by bootstrap aggregation', *Pattern Recognition* **36**(6), 1303–1309.

Hothorn, T., Lausen, B., Benner, A. & Radespiel-Tröger, M. (2004), 'Bagging survival trees', *Statistics in Medicine* **23**(1), 77–91.

Hu, H., Li, J., Plank, A. W., Wang, H. & Daggard, G. (2006), A comparative study of classification methods for microarray data analysis, *in* 'Proceedings of the 4th Australian Data Mining Conference, AusDM 2006', pp. 33–37.

Japkowicz, N. (2000), The class imbalance problem: Significance and strategies, *in* 'Proceedings of the International Conference on Artificial Intelligence, ICAI 2000', Vol. 1, pp. 111–117.

Kang, P. & Cho, S. (2006), EUS SVMs: Ensemble of under-sampled SVMs for data imbalance problems, *in* 'Proceedings of the International Conference on Neural Information Processing', pp. 837–846.

Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L. & Ratanamahatana, C. A. (2011), '*UCR* repository of time series classification/clustering homepage: `http://www.cs.ucr.edu/~eamonn/time_series_data/` last accessed: 23 January 2013'.

Kim, H., Pang, S., Je, H., Kim, D. & Bang, S. (2002), Pattern classification using support vector machine ensemble, *in* 'Proceedings of the 16th International Conference on Pattern Recognition', Vol. 2, pp. 160–163.

Kim, K. (2003), 'Financial time series forecasting using support vector machines', *Neurocomputing* **55**(1-2), 307–319.

Kim, M.-J. & Kang, D.-K. (2010), 'Ensemble with neural networks for bankruptcy prediction', *Expert Systems with Applications* **37**(4), 3373 – 3379.

Kittler, J. (1998), 'Combining classifiers: A theoretical framework', *Pattern Analysis & Applications* **1**(1), 18–27.

Kohavi, R. (1995), A study of cross-validation and bootstrap for accuracy estimation and model selection, *in* 'Proceedings of the International Joint Conference on Artificial Intelligence', Vol. 14, pp. 1137–1145.

Kohavi, R. & Wolpert, D. (1996), Bias plus variance decomposition for zero-one loss functions, *in* 'Proceedings of the 13th International Conference in Machine Learninng', pp. 275–283.

Koknar-Tezel, S. & Latecki, L. (2009), Improving SVM classification on imbalanced data sets in distance spaces, *in* 'Proceedings of the 9th IEEE International Conference on Data Mining, ICDM 2009', IEEE, pp. 259–267.

Kong, E. & Dietterich, T. (1995), Error-correcting output coding corrects bias and variance, *in* 'Proceedings of the 12th International Conference on Machine Learning', pp. 313–321.

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

Krogh, A. & Vedelsby, J. (1995), 'Neural network ensembles, cross validation, and active learning', *Advances in Neural Information Processing Systems* pp. 231–238.

Kubat, M., Holte, R. & Matwin, S. (1998), 'Machine learning for the detection of oil spills in satellite radar images', *Machine Learning* **30**(2), 195–215.

Kubat, M. & Matwin, S. (1997), Addressing the curse of imbalanced training sets: One-sided selection, *in* 'Proceedings of the International Conference on Machine Learning', pp. 179–186.

Laurikkala, J. (2001), 'Improving identification of difficult small classes by balancing class distribution', *Artificial Intelligence in Medicine* pp. 63–66.

Leung, K. & Parker, D. (2003), Empirical comparisons of various voting methods in bagging, *in* 'Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', pp. 595–600.

Li, C. (2007), Classifying imbalanced data using a bagging ensemble variation (BEV), *in* 'Proceedings of the 45th Annual Southeast Regional Conference', pp. 203–208.

Liang, G. (2012), An investigation of sensitiveity on bagging predictors: An empirical approach, *in* 'Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI 2012', pp. 2439–2440.

Liang, G. (2013), A effective method for imbalanced time series classification: Hybrid-sampling, *in* 'Proceedings of the 26th Australasian Joint Conference on Artificial Intelligence, AI 2013', Springer, pp. 374–385.

Liang, G. & Cohn, A. G. (2013), An effective approach for imbalanced classification: Unevenly balanced bagging, *in* 'Proceedings of the 27th AAAI Conference on Artificial Intelligence, AAAI 2013', pp. 1633–1634.

Liang, G. & Zhang, C. (2011*a*), An empirical evaluation of bagging with different algorithms on imbalanced data, *in* 'Proceedings of the 7th International Conference on Advanced Data Mining and Applications, ADMA 2011', pp. 339–352.

Liang, G. & Zhang, C. (2011*b*), Empirical study of bagging predictors on medical data, *in* 'Proceedings of the 9th Australasian Data Mining Conference, AusDM 2011', Vol. 121 of *CRPIT*, pp. 31–40.

Liang, G. & Zhang, C. (2012*a*), A comparative study of sampling methods and algorithms for imbalanced time series classification, *in* 'Proceedings of the 25th Australasian Joint Conference on Artificial Intelligence, AI 2012', Springer, pp. 637–648.

Liang, G. & Zhang, C. (2012*b*), An efficient and simple under-sampling technique for imbalanced time series classification, *in* 'Proceedings of the 21st ACM International Conference on Information and Knowledge Management, CIKM 2012', ACM, pp. 2339–2342.

Liang, G., Zhu, X. & Zhang, C. (2011*a*), An empirical study of bagging predictors for different learning algorithms, *in* 'Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI 2011', pp. 1802–1803.

Liang, G., Zhu, X. & Zhang, C. (2011*b*), An empirical study of bagging predictors for imbalanced data with different levels of class distribution, *in* 'Proceedings of the 24th Australasian Joint Conference on Artificial Intelligence, AI 2011', Springer, pp. 213–222.

Liang, G., Zhu, X. & Zhang, C. (2014), 'The effect of varying levels of class distribution on bagging with different algorithms: An empirical study', *International Journal of Machine Learning and Cybernetics* **5**(1), 63–71.

Ling, C., Huang, J. & Zhang, H. (2003), AUC: a better measure than accuracy in comparing learning algorithms, *in* 'Proceedings of the 16th Canadian Society for Computational Studies of Intelligence Conference on Advances in Artificial Intelligence', pp. 329–341.

Ling, C. & Li, C. (1998), Data mining for direct marketing: Problems and solutions, *in* 'Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining', pp. 73–79.

Liu, W. & Chawla, S. (2011), Class confidence weighted KNN algorithms for imbalanced data sets, *in* 'Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD 2011', pp. 345–356.

Liu, W., Chawla, S., Cieslak, D. & Chawla, N. (2010), A robust decision tree algorithm for imbalanced data sets, *in* 'Proceedings of the 10th SIAM International Conference on Data Mining', pp. 766–777.

Liu, X., Wu, J. & Zhou, Z. (2009), 'Exploratory undersampling for class-imbalance learning', *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **39**(2), 539–550.

Lopes, L., Scalabrin, E. & Fernandes, P. (2008), 'An empirical study of combined classifiers for knowledge discovery on medical data bases', *Advanced Web and Network Technologies, and Applications* pp. 110–121.

Maloof, M. (2003), Learning when data sets are imbalanced and when costs are unequal and unknown, *in* 'ICML 2003 Workshop on Learning from Imbalanced Data Sets II', Vol. 2.

Mazurowski, M., Habas, P., Zurada, J., Lo, J., Baker, J. & Tourassi, G. (2008), 'Training neural network classifiers for medical decision making:

The effects of imbalanced datasets on classification performance', *Neural Networks* **21**(2-3), 427–436.

Mena, L. & Gonzalez, J. (2006), Machine learning for imbalanced datasets: Application in medical diagnostic, *in* 'Proceedings of the 19th International FLAIRS Conference', pp. 574–579.

Merz, C. & Murphy, P. (2006), 'UCI Repository of Machine Learning Databases `http://archive.ics.uci.edu/ml/` last accessed: 23 January 2013'.

Meyer, I. (2007), 'A practical guide to the art of RNA gene prediction', *Briefings in Bioinformatics* **8**(6), 396–414.

Molinara, M., Ricamato, M. & Tortorella, F. (2007), Facing imbalanced classes through aggregation of classifiers, *in* 'Proceedings of the 14th International Conference on Image Analysis and Processing, ICIAP 2007', pp. 43–48.

Ng, W. & Dash, M. (2006), An evaluation of progressive sampling for imbalanced data sets, *in* 'Proceedings of the 6th IEEE International Conference on Data Mining Workshops, ICDM Workshops 2006', IEEE, pp. 657–661.

Opitz, D. & Maclin, R. (1999), 'Popular ensemble methods: An empirical study', *Journal of Artificial Intelligence Research* **11**(1), 169–198.

Opitz, D. & Shavlik, J. (1996*a*), 'Actively searching for an effective neural network ensemble', *Connection Science* **8**(3-4), 337–354.

Opitz, D. & Shavlik, J. (1996*b*), 'Generating accurate and diverse members of a neural-network ensemble', *Advances in Neural Information Processing Systems* pp. 535–541.

Phua, C., Alahakoon, D. & Lee, V. (2004), 'Minority report in fraud detection: Classification of skewed data', *ACM SIGKDD Explorations Newsletter* **6**(1), 50–59.

Provost, F. & Fawcett, T. (1997), Analysis and visualization of classifier performance with nonuniform class and cost distributions, *in* 'Proceedings of AAAI 97 Workshop on AI Approaches to Fraud Detection & Risk Management', pp. 57–63.

Provost, F. & Fawcett, T. (2001), 'Robust classification for imprecise environments', *Machine Learning* **42**(3), 203–231.

Provost, F., Fawcett, T. & Kohavi, R. (1998), The case against accuracy estimation for comparing induction algorithms, *in* 'Proceedings of the 15th International Conference on Machine Learning', Vol. 445-453.

Qin, Z. (2005), ROC analysis for predictions made by probabilistic classifiers, *in* 'Proceedings of the 4th International Conference on Machine Learning and Cybernetics, ICMLC 2005', Vol. 5, pp. 3119–3124.

Quinlan, J. (1986), 'Induction of decision trees', *Machine Learning* **1**(1), 81–106.

Quinlan, J. (1996), Bagging, boosting, and C4.5, *in* 'Proceedings of the National Conference on Artificial Intelligence', pp. 725–730.

Rao, R., Krishnan, S. & Niculescu, R. (2006), 'Data mining for improved cardiac care', *ACM SIGKDD Explorations Newsletter* **8**, 3–10.

Schapire, R. (1990), 'The strength of weak learnability', *Machine Learning* **5**(2), 197–227.

Schapire, R. (1997), Using output codes to boost multiclass learning problems, *in* 'Proceedings of the 14th International Conference on Machine Learning', pp. 313–321.

Slaby, A. (2007), ROC analysis with Matlab, *in* 'Proceedings of the 29th International Conference on Information Technology Interfaces, ITI 2007', pp. 191–196.

Su, C. & Hsiao, Y. (2007), 'An evaluation of the robustness of MTS for imbalanced data', *IEEE Transactions on Knowledge and Data Engineering,* **19**, 1321–1332.

Su, X., Khoshgoftarr, T. & Zhu, X. (2008), VoB predictors: Voting on bagging classifications, *in* 'Proceedings of the 19th International Conference on Pattern Recognition, ICPR 2008', pp. 1–4.

Sun, Y., Kamel, M., Wong, A. & Wang, Y. (2007), 'Cost-sensitive boosting for classification of imbalanced data', *Pattern Recognition* **40**, 3358–3378.

Tu, M., Shin, D. & Shin, D. (2009*a*), A comparative study of medical data classification methods based on decision tree and bagging algorithms, *in* 'Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, DASC 2009', pp. 183–187.

Tu, M., Shin, D. & Shin, D. (2009*b*), Effective diagnosis of heart disease through bagging approach, *in* 'Proceedings of the 2nd International Conference on Biomedical Engineering and Informatics, BMEI 2009', IEEE, pp. 1–4.

Tuv, E. (2006), 'Ensemble learning', *Feature Extraction* pp. 187–204.

Valentini, G. & Dietterich, T. (2002), 'Bias & Variance analysis and ensembles of SVM', *Multiple Classifier Systems* pp. 27–38.

Valentini, G. & Dietterich, T. (2003), Low bias bagged support vector machines, *in* 'Proceedings of the 20th International Conference on Machine Learning, ICML 2003', pp. 752–759.

Van Hulse, J., Khoshgoftaar, T. & Napolitano, A. (2007), Experimental perspectives on learning from imbalanced data, *in* 'Proceedings of the 24th International Conference on Machine Learning', pp. 935–942.

Wang, B., Zhou, Y., Qiu, X., Zhang, Q. & Huang, X. (2010), Bagging to find better expansion words, *in* 'Proceedings of the International Conference on Natural Language Processing and Knowledge Engineering, NLP-KE 2010', pp. 1–8.

Webb, G. & Zheng, Z. (2004), 'Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques', *IEEE Transactions on Knowledge and Data Engineering* **16**(8), 980–991.

Wei, L. & Keogh, E. (2006), Semi-supervised time series classification, *in* 'Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and data mining', ACM, pp. 748–753.

Weiss, G. (2004), 'Mining with rarity: A unifying framework', *ACM SIGKDD Explorations Newsletter* **6**, 7–19.

Weiss, G. & Provost, F. (2001), The effect of class distribution on classifier learning: An empirical study, Technical report, Rutgers University.

Weiss, G. & Provost, F. (2003), 'Learning when training data are costly: The effect of class distribution on tree induction', *Journal of Artificial Intelligence Research* **19**(1), 315–354.

West, D., Dellana, S. & Qian, J. (2005), 'Neural network ensemble strategies for financial decision applications', *Computers & Operations Research* **32**(10), 2543–2559.

Witten, I. & Frank, E. (2005), *Data Mining: Practical Machine Learning Tool and Techniques*, Morgan Kaufmann, San Francisco.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P. et al. (2008), 'Top 10 algorithms in data mining', *Knowledge and Information Systems* **14**(1), 1–37.

Xi, X., Keogh, E., Shelton, C., Wei, L. & Ratanamahatana, C. (2006), Fast time series classification using numerosity reduction, *in* 'Proceedings of the 23rd International Conference on Machine Learning, ICML 2006', pp. 1033–1040.

Xu, W., Zuo, M., Zhang, M. & He, R. (2010), Constraint bagging for stock price prediction using neural networks, *in* 'Proceedings of the International Conference on Modelling, Identification and Control, ICMIC 2011', pp. 606–610.

Yang, P., Zhang, Z., Zhou, B. & Zomaya, A. (2011), Sample subset optimization for classifying imbalanced biological data, *in* 'Proceedings of the 15th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining', pp. 333–344.

Yang, Q. & Wu, X. (2006), '10 challenging problems in data mining research', *International Journal of Information Technology & Decision Making* **5**(4), 597–604.

Yen, S., Lee, Y., Lin, C. & Ying, J. (2006), Investigating the effect of sampling methods for imbalanced data distributions, *in* 'Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, SMC 06', Vol. 5, pp. 4163–4168.

Zaman, F. & Hirose, H. (2008), A robust bagging method using median as a combination rule, *in* 'Proceedings of the 8th International Conference

on Computer and Information Technology Workshops, IEEE CIT Workshops 2008', pp. 55–60.

Zaman, F. & Hirose, H. (2009), 'Effect of subsampling rate on subbagging and related ensembles of stable classifiers', *Pattern Recognition and Machine Intelligence* pp. 44–49.

Zavaljevski, N., Stevens, F. & Reifman, J. (2002), 'Support vector machines with selective kernel scaling for protein classification and identification of key amino acid positions', *Bioinformatics* **18**(5), 689–696.

Zhu, X. (2007), Lazy bagging for classifying imbalanced data, *in* 'Proceedings of the 7th IEEE International Conference on Data Mining, ICDM 2007', pp. 763–768.

Zhu, X., Bao, C. & Qiu, W. (2008), Bagging very weak learners with lazy local learning, *in* 'Proceedings of the 19th International Conference on Pattern Recognition, ICPR 2008', pp. 1–4.

Zhu, X. & Yang, Y. (2008), 'A lazy bagging approach to classification', *Pattern Recognition* **41**(10), 2980–2992.