2008 IEEE International Conference on Data Mining Workshops

A Comparative Study of Data Sampling and Cost Sensitive Learning

Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, Amri Napolitano
Florida Atlantic University, Boca Raton, Florida, USA
(chrisseiffert@gmail.com; taghi@cse.fau.edu; jvanhulse@gmail.com; amrifau@gmail.com)

Abstract

Two common challenges data mining and machine learning practitioners face in many application domains are unequal classification costs and class imbalance. Most traditional data mining techniques attempt to maximize overall accuracy rather than minimize cost. When data is imbalanced, such techniques result in models that highly favor the overrepresented class, the class which typically carries a lower cost of misclassification. Two techniques that have been used to address both of these issues are cost sensitive learning and data sampling. In this work, we investigate the performance of two cost sensitive learning techniques and four data sampling techniques for minimizing classification costs when data is imbalanced. We present a comprehensive suite of experiments, utilizing 15 datasets with 10 cost ratios, which have been carefully designed to ensure conclusive, significant and reliable results.

1 Introduction

Making cost sensitive decisions is a problem common to many application domains. A cost sensitive model attempts to minimize the costs (or maximize profits) associated with its decisions, rather than simply achieving high overall accuracy. When misclassifying examples of one class is much more expensive than misclassifying examples of the other class(es), then a model which focuses more on correctly classifying examples of the expensive class may be preferable to one that treats all classes equally. Examples of application domains where such a situation occurs are fraud detection, medical diagnosis and software quality engineering.

A problem that goes hand-in-hand with unequal misclassification costs is class imbalance. Data is said to be imbalanced (with respect to class) if one class occurs much more (or less) frequently than the other class(es). When data is highly skewed, many traditional machine learning algorithms will favor the overrepresented (majority) class in order to maximize overall accuracy while misclassifying a disproportionately large number of minority class examples. Typically the minority class carries a higher cost of misclassification, making such a strategy inappropriate.

In this study, we empirically investigate and compare two strategies for dealing with the problems of unequal misclassification costs and class imbalance: cost sensitive learning and data sampling. Cost sensitive learning can be achieved by providing the learner with information regarding example weights or by changing the decision threshold used by the learner when making classification decisions. Data sampling alters the class distribution of the training data, forcing the learner to focus more on correctly classifying examples of the more expensive (positive) class.

The objective of this work is to present highly reliable [9] and conclusive results related to cost sensitive decision-making based on imbalanced training data. Dai [4] cites three key components of reliability in knowledge discovery: *stability*, *equivalence*, and *consistency*. Stability in experimentation refers to repeating a test using the same population and obtaining the same results. Two or more different patterns constructed based on the same content are equivalent if the outcomes are the same. Finally, consistency implies that the learned knowledge should not contain contradictions. We strongly believe that our experimentation satisfies all of the requirements for reliable empirical work. We evaluate four data sampling techniques and two cost sensitive learning methods by training models using two common machine learning algorithms. Fifteen datasets from various application domains are used to train these models. We employ a high degree of repetition, performing 20 independent runs of 5-fold cross validation, increasing our confidence in the reliability of our experiments. In addition, statistical analysis is presented for all results, allowing for the significance of our findings to be validated. In total, over 200,000 models are trained and evaluated for this study. To our knowledge, no previous work in cost sensitive learning has provided such a comprehensive empirical investigation focused on obtaining highly reliable results.

978-0-7695-3503-6/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDMW.2008.119

2 Related Work

Elkan [7] provides the foundations for the cost sensitive learning algorithms used in this study. Topics including cost matrices, selecting the optimal decision threshold and using data sampling to force error-based learners to make cost sensitive decisions are discussed. Related details can be found in Section 3.3.

A preliminary study comparing the performance of cost sensitive learning and data sampling was presented by Weiss et al. [15]. Their work compared the cost sensitive learning capability of C5.0, a commercial tool, to that of random undersampling and random oversampling. While their results were not conclusive, it was determined that oversampling tends to perform better than undersampling, and that cost sensitive learning, while not often the best, is generally more consistent than the sampling techniques. Our work expands upon this work, evaluating twice as many techniques (two cost sensitive and four data sampling), using two commonly used learners (C4.5 and RIPPER, implemented in Weka [16], an open source data mining suite) and providing statistical analysis of the results. Our work is also different in that we draw definitive conclusions about each technique's performance by evaluating their average performance across 15 datasets. Drummond and Holte [6] demonstrate that models built using oversampled data tend to be insensitive to varying costs, and that models built using undersampled data perform better.

Other techniques have also been proposed for cost sensitive learning, but evaluating these techniques is left as future work. Domingos proposed MetaCost [5], which employs bagging to determine optimal class labels for training data, relabeling examples accordingly. Several cost sensitive boosting techniques have also been proposed [8, 13, 14]. These techniques modify the AdaBoost algorithm using cost information provided by the user. Margineautu [11] proposes a technique for making decision tree learners cost sensitive using confidence and class probability estimates. While many of these techniques show promise, their evaluation is left as future work.

3 Experiments

3.1 Datasets

Table 1 provides the details of the 15 datasets used in our experiments. This table provides the number of examples in each dataset (Size), as well as the number and percentage of examples belonging to the positive class (#Pos and %Pos). Of these 15 datasets, 7 are real-world software quality datasets. KC1 is available through the NASA Data Metrics Program[1]. SP1 through SP4 are proprietary datasets from a large telecommunications company and the two CCCS datasets are derived from a Department of Defense software project. The mammography dataset was generously provided by Dr. Nitesh Chawla [2]. The remaining seven datasets represent a wide variety of application domains and were obtained through the UCI repository[2]. This collection of datasets includes many different sizes and levels of imbalance. Since this study focuses only on binary classification, some datasets were transformed by selecting a single class as the positive class (while the remaining classes made up the negative class). In the case of datasets where the dependent variable was continuous, a domain appropriate threshold was selected.

Dataset	Size	#Pos	%Pos
SP3	3541	47	1.33
SP4	3978	92	2.31
MAMMOGRAPHY	11183	260	2.32
NURSERY	12960	328	2.53
SP2	3981	189	4.75
SP1	3649	229	6.28
GLASS	214	17	7.94
ECOLI	336	35	10.42
SEGMENT	2310	330	14.29
KC1	2107	325	15.42
CCCS04	282	55	19.50
CONTRACEPTIVE	1473	333	22.61
HABERMAN	306	81	26.47
CCCS02	282	83	29.43
PIMA	768	268	34.90

Table 1. Dataset Characteristics

3.2 Data Sampling

Data sampling modifies the distribution of training data by either removing examples of the majority class (undersampling) or adding examples to the minority class (oversampling). The simplest sampling techniques are random undersampling and random oversampling. Random undersampling (RUS) selects examples (at random) to be deleted from the majority class. Random oversampling (ROS) randomly duplicates examples of the minority class. Undersampling can lead to loss of information (due to deleting examples) while random oversampling can lead to overfitting [6].

Chawla et al. proposed a more "intelligent" method of oversampling, the Synthetic Minority Oversampling Technique, or SMOTE [2]. Rather than randomly duplicating minority class examples, SMOTE creates new examples by extrapolating between existing minority class instances. Borderline-SMOTE, proposed by Han et al. [10], attempts to improve upon SMOTE by only creating new minority class examples based on those that lie near the decision boundary. In this paper, SMOTE and borderline-SMOTE are abbreviated SM and BSM, respectively.

[1]http://mdp.ivv.nasa.gov

[2]http://archive.ics.uci.edu/ml/

	actual class 0	actual class 1
predicted class 0	C(0,0)	C(0,1)
predicted class 1	C(1,0)	C(1,1)

Figure 1. Two class cost matrix

3.3 Cost Sensitive Learning

Every classification decision, whether it is correct or incorrect, is associated with a cost. These costs are described in a *cost matrix*, an example of which is shown in Figure 1. The diagonal (where the predicted class = the actual class) provides the costs associated with correct classifications while the values not found on this diagonal provide the costs associated with the different types of miclassifications. For simplicity, cost matrices are typically used with no cost for correct classifications. Therefore in a binary classification problem we are concerned with only two costs, C(0,1) and C(1,0), or the *cost ratio*, C(0,1)/C(1,0).

Given such a cost matrix (or cost ratio), Elkan [7] provides the required formulas for modifying the decision threshold or resampling data to construct a cost sensitive model. For binary classification, learners typically use a default decision threshold of 0.5. That is, if the predicted probability of an example belonging to class *i* is greater than 50%, then that example is classfied as belonging to class *i*. When misclassification costs are not equal, the default decision threshold may not be optimal. Based on the user supplied cost ratio, the optimal decision threshold, $p*$, can be calculated as:

$$p* = \frac{C(1,0) - C(0,0)}{C(1,0) - C(0,0) + C(0,1) - C(1,1)} \quad (1)$$

The formula for $n*$, the factor by which the number of majority class examples should be multiplied to achieve the same effect using undersampling, is provided as:

$$n* = \frac{p*}{(1-p*)} \frac{(1-p_o)}{p_o} \quad (2)$$

where p_o is the decision threshold being used. Using the default decision threshold of 0.5 reduces this equation to $p*/(1-p*)$. For oversampling, the number of minority class examples is divided by $n*$ (Since $n* < 1$, this increases the number of negative class examples).

Another technique for making cost sensitive models is example weighting. This functionality is provided by the "CostSensitiveClassifier" meta-learner in the Weka data mining tool [16]. Using this meta-learner, the Weka implementations of C4.5 and RIPPER can place greater emphasis on examples with higher weights when constructing a model. In this paper, changing the decision threshold and example weighting are denoted Thresh and Weight, respectively.

3.4 Performance metric and ANOVA Analysis

To ensure that larger datasets do not dominate the results, the average cost of misclassifying an example, rather than total cost, is utilized. For each dataset, the per-example cost (PEC) is calculated as:

$$\text{PEC} = \frac{\#fpos \times C(p,n) + \#fneg \times C(n,p)}{\#tpos + \#tneg + \#fpos + \#fneg} \quad (3)$$

where $\#tpos$ and $\#tneg$ are the number of examples correctly classified as positive and negative, $\#fpos$ and $\#fneg$ are the number of examples incorrectly classified as positive and negative and $C(p,n)$ and $C(n,p)$ are the costs associated with false positives (fpos) and false negatives (fneg).

All results presented in this study are tested for statistical significance at the $\alpha = 5\%$ level using one-factor analysis of variance (ANOVA) [1]. A one-factor ANOVA model, which can be used to test the hypothesis that the classification performances for each level of the main factor(s) are equal against the alternative hypothesis that at least one is different, can be represented as:

$$\psi_{jn} = \mu + \theta_j + \epsilon_{jn}$$

where ψ_{jn} is the response (PEC) for the n^{th} observation of the j^{th} level of θ, μ is the overall mean performance, θ_j is the mean performance of level j for the factor θ, and ϵ_{jn} is the random error.

The main factor θ is the cost sensitive or data sampling technique used. That is, we are testing to see if the average performance of the seven levels (NONE, Thresh, Weight, RUS, ROS, SM and BSM) of θ are equal. If the alternative hypothesis (that at least one level of θ is different) is accepted, numerous procedures can be used to determine which are different. This involves a pair-wise comparison of the mean performance of each technique, with the null hypothesis that the means are equal. In this work, we use Tukey's Honestly Significant Difference (HSD) test [1] to identify which levels of θ are significantly different.

3.5 Design Summary

Models were built using both the C4.5 decision tree learner [12] and RIPPER [3], a rule based learner. These learners were selected due to their frequent use in related literature as well as their wide availability to practitioners in the field. Experiments were performed using 5-fold cross validation. That is, the datasets were split into five partitions, one of which was used as test data while models were built using the remaining four partitions. This process was repeated five times so that each partitions acts as test data once. Twenty independent runs of this process were performed to eliminate any biasing that may occur due to the random partitioning process and to ensure the statistical significance of our results. In addition to using four data sampling techniques (Section 3.2) and the two cost sensitive learning techniques (Section 3.3), models were constructed without using data sampling, thresholding or weighting, denoted as NONE throughout this paper, as a baseline for comparison. Ten cost ratios between 1 and 20 were used to construct models and evaluate their performance. In total, 210,000 model performances were evaluated to produce the results presented in this work.

4 Empirical Results

In this section, we present the results of our experiments. We evaluate the performance of two cost sensitive learning techniques and four data sampling techniques with two commonly used learners. The results are presented as an average of the performance across all 15 datasets. The PEC is calculated for each fold of cross validation and the average of $15 \times 20 \times 5 = 1500$ (15 datasets and 20 runs of 5-fold cross validations) PECs are presented in this study. The results using the C4.5 and RIPPER learners are presented in Sections 4.1 and 4.2.

4.1 C4.5

This section presents the results using the C4.5 decision tree learner. Table 2 shows the PEC achieved by each technique at various cost ratios. Table 3 shows the results of Tukey's Honestly Significant Difference (HSD) test at the $\alpha = 5\%$ level. Finally, Figure 2 shows the improvement each technique achieved over None. The values in this figure are calculated as:

$$\text{Relative Cost} = \frac{AvgCost(t)}{AvgCost(NONE)} \tag{4}$$

where $AvgCost(t)$ is the average PEC achieved by the given technique t. Values below the line labeled NONE indicate a reduction in cost, while values above this line indicate an increase in cost. Several cost ratios that are not presented in Tables 2 and 3 are included in this figure.

Table 2 and Figure 2 show a large amount of variation between the performances of the cost sensitive learning and sampling techniques. Random oversampling (ROS) is the only technique to result in increased cost at cost ratio 2, and at other cost ratios its cost reduction is much less than the other techniques. At the highest cost ratio, it has only reduced the average cost of classifications to about 75% of NONE. The other oversampling techniques perform better than ROS, but are still outperformed by the cost sensitive techniques, especially at higher cost ratios (at low cost ratios, SM and BSM outperform Thresh). The two cost sensitive techniques perform very well, with Weight consistently outperforming Thresh. At the highest cost ratio, Thresh and Weight reduced cost to about 47% and 42% of NONE, respectively. Random undersampling results in a similar performance to Weight at lower cost ratios, but at higher cost ratios it emerges as the superior technique, reducing cost to about 36% of NONE (at cost ratio 20).

	Cost Ratio					
	1	3	5	10	15	20
NONE	0.1091	0.2441	0.3791	0.7166	1.0541	1.3916
BSM	0.1091	0.2259	0.3150	0.5232	0.7265	0.9152
ROS	0.1091	0.2391	0.3404	0.5859	0.8183	1.0470
SM	0.1091	0.2246	0.3116	0.5026	0.6867	0.8448
RUS	0.1091	0.2285	0.3028	0.3945	0.4565	0.5004
Thresh	0.1091	0.2306	0.3179	0.4606	0.5655	0.6534
Weight	0.1091	0.2240	0.2997	0.4335	0.5099	0.5842

Table 2. Average costs achieved using the C4.5 learner

	Cost Ratio					
	1	3	5	10	15	20
NONE	A	A	C	F	E	E
BSM	A	A	AB	D	C	C
ROS	A	A	B	E	D	D
SM	A	A	A	CD	C	C
RUS	A	A	A	A	A	A
Thresh	A	A	AB	BC	B	B
Weight	A	A	A	AB	AB	B

Table 3. Results of HSD test for C4.5

The statistical significance of these results are demonstrated in Table 3. If two techniques are assigned the same letter for a given cost ratio, then their performance was not significantly different at that cost ratio. Techniques that do not have a letter in common resulted in significantly different performances. For example, at cost ratio 3, there was no statistically significant difference in the performance of any of the techniques (they are all assigned the same letter), but at cost ratio 5, all of the techniques significantly outperformed NONE (it is the only one assigned the letter C).

At lower cost ratios, there is relatively little difference in the performance of the various techniques. For example, at cost ratio 3, even though SM reduced the cost to 92% of NONE, this difference is not statistically significant. At higher cost ratios, however, the differences between the techniques is significant. At the highest cost ratio, RUS significantly outperforms the other techniques, being the only technique assigned the letter A. RUS is assigned to group A for all cost ratios. At cost ratio 20, the two cost sensitive techniques, Thresh and Weight, are both assigned to group B, suggesting that their performances were not significantly different from each other, but that they do significantly outperform the oversampling techniques, all of which are assigned to group C or worse. Although Weight consistently

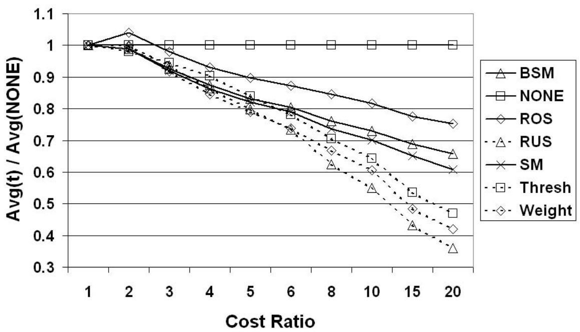

Figure 2. Relative costs using C4.5

	Cost Ratio 1	3	5	10	15	20
NONE	0.1064	0.2406	0.3748	0.7103	1.0458	1.3813
BSM	0.1062	0.2175	0.3072	0.5373	0.7620	1.0019
ROS	0.1062	0.2282	0.3369	0.6009	0.8304	1.0621
SM	0.1062	0.2153	0.3012	0.5077	0.6945	0.8787
RUS	0.1064	0.2217	0.3003	0.3940	0.4485	0.4974
Thresh	0.1064	0.2403	0.3449	0.4844	0.5967	0.6712
Weight	0.1066	0.2187	0.2903	0.3947	0.4441	0.4883

Table 4. Average costs achieved using the RIPPER learner

performed better than Thresh, this difference is not found to be significant at any cost ratio (they always have a letter in common). The difference in performance of SM and BSM is never significant, but both do significantly outperform ROS at higher cost ratios. All of the techniques significantly outperform NONE for cost ratios five and higher.

4.2 RIPPER

In this section, we examine the performance of the various techniques using RIPPER as the base learner. The supporting data for this analysis can be found in Tables 4 and 5 as well as Figure 3.

Using RIPPER, RUS and Weight result in very similar performances, especially at the highest cost ratios. Figure 3 shows the performance of these two techniques to be nearly identical for cost ratios eight and higher. RUS and Weight

	Cost Ratio 1	3	5	10	15	20
NONE	A	B	C	E	F	E
BSM	A	A	A	C	D	D
ROS	A	AB	B	D	E	D
SM	A	A	A	BC	C	C
RUS	A	AB	A	A	A	A
Thresh	A	B	B	B	B	B
Weight	A	A	A	A	A	A

Table 5. Results of HSD test for RIPPER

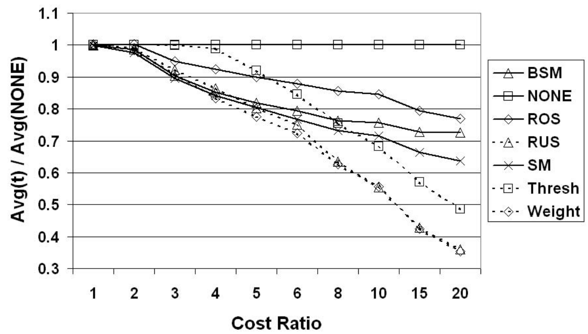

Figure 3. Relative costs using RIPPER

are also the two best techniques, both reducing cost to about 35% (at cost ratio 20) of NONE. The next best technique at higher cost ratios is Thresh. For cost ratios 10 and higher, Thresh results in lower costs than all of the oversampling techniques, but at lower cost ratios, the opposite is true. At lower cost ratios, the performance of Thresh is very similar to that of NONE. For each fold of cross validation, RIPPER produces only two posterior probabilities, rather than a range of probabilities like other learners. Therefore, especially at lower cost ratios, it is likely that changing the decision threshold will not affect the resulting classifications. At lower cost ratios, SM and BSM perform about as well as RUS and Weight, but as the cost ratio is increased, SM and BSM underperform. At the highest cost ratio, they only reduce cost to 63.7% (SM) and 72.5% (BSM) of NONE. Once again, ROS is the weakest technique. Although ROS outperforms Thresh at very low cost ratios, for all cost ratios six and higher, ROS is the worst performing technique.

Table 5 provides the results of statistical significance (HSD) testing. At low cost ratios, BSM and SM perform very well, being in group A for cost ratios five and lower. However, at higher cost ratios these two oversampling techniques are significantly outperformed by both RUS and Weight. RUS and Weight are both in group A for all cost ratios. Thresh is in group B for all cost ratios (except 1), significantly outperforming the oversampling techniques at all higher cost ratios. Once again, random oversampling tends to underperform, being in the second worst group (NONE is the worst) for all cost ratios five and higher.

4.3 Summary of Results

In this section, we compared the performance of six different techniques for constructing cost sensitive models. Our results show that, in general, random undersampling (RUS) tends to result in the best performance. Using C4.5, RUS significantly outperforms the other five techniques at higher cost ratios. Using RIPPER, its performance is similar to that of instance weighting (Weight), both of which significantly outperform the other four techniques. The two cost sensitive techniques (Weight and Thresh) performed consistently well, usually outperforming the oversampling techniques. In general, oversampling is a poor choice for constructing cost sensitive models, except at lower cost ratios where there tends to be no significant difference between the six techniques.

5 Conclusion

This work presents a comprehensive study comparing the performance of traditional cost sensitive learning techniques to that of sampling. We compare two variations of cost sensitive learning (example weighting and changing the decision threshold) to four sampling techniques (random undersampling, random oversampling and two variations of SMOTE). Our experiments utilize 15 imbalanced datasets and two commonly used learners. To enhance the reliability of our results, all experiments were performed using 20 independent runs of 5-fold cross validation. In addition, all results are accompanied by statistical analysis, ensuring the significance of our findings. The thorough experimentation employed in this work helps ensure that our results are widely-applicable to data mining researchers and practitioners, and are highly reliable and consistent.

Our results show that random undersampling tends to perform as well as, or better than, the other techniques. Using C4.5, random undersampling significantly outperforms all five of the other techniques at higher cost ratios. For the other learner, RIPPER, random undersampling performs as well as example weighting, both of which significantly outperform the remaining techniques at higher cost ratios. The two traditional cost sensitive learning techniques result in relatively similar performances, with example weighting usually outperforming decision threshold adjustment. In general, our experiments show that oversampling is a poor choice for building cost sensitive models.

This last result is somewhat in contrast to an earlier work [15] which showed that oversampling resulted in the lowest cost of misclassification more often than undersampling, although the authors state no such definitive conclusion. Unlike the previous work, we find that undersampling outperforms oversampling, and present statistical analysis supporting this claim. That is not to say that oversampling will not outperform undersampling for some datasets and/or learners, but in general, we find undersampling to be superior. Undersampling, however, has the drawback that data is lost, and may be inappropriate for small datasets when data is highly skewed and cost ratios are very high.

Future work will investigate other techniques for constructing cost sensitive models, including but not limited to cost sensitive boosting and MetaCost [5]. The performance of these techniques using learners from additional machine learning paradigms will be investigated. As with any empirical study, similar experimentation should be conducted using additional datasets to gain further confidence in the conclusions we have presented.

References

[1] M. L. Berenson, D. M. Levine, and M. Goldstein. *Intermediate Statistical Methods and Applications: A Computer Package Approach*. Prentice-Hall, Inc., 1983.

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

[3] W. W. Cohen. Fast effective rule induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123. Morgan Kaufmann, 1995.

[4] H. Dai. A study on reliability in graph discovery. In *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 775–779, Washington, DC, USA, 2006. IEEE Computer Society.

[5] P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[6] C. Drummond and R. C. Holte. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning*, 2003.

[7] C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 239–246, 2001.

[8] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan. AdaCost: misclassification cost-sensitive boosting. In *Proc. 16th International Conf. on Machine Learning*, pages 97–105. Morgan Kaufmann, San Francisco, CA, 1999.

[9] Y. Feng and Z. Wu. Enhancing reliability throughout knowledge discovery process. In *ICDMW '06: Proceedings of the Sixth IEEE International Conference on Data Mining - Workshops*, pages 754–758, Washington, DC, USA, 2006. IEEE Computer Society.

[10] H. Han, W. Y. Wang, and B. H. Mao. Borderline-SMOTE: A new over-sampling method in imbalanced

data sets learning. In *International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644*, pages 878–887. Springer-Verlag, 2005.

[11] D. D. Margineantu. Class probability estimation and cost-sensitive classification decisions. In *Proceedings of the 13th European Conference on Machine Learning*, pages 270–281, London, UK, 2002. Springer-Verlag.

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

[13] Y. Sun, M. S. Kamel, A. K. C. Wong, and Y. Wang. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378, 2007.

[14] K. M. Ting. A comparative study of cost-sensitive boosting algorithms. In *Proc. 17th International Conf. on Machine Learning*, pages 983–990. Morgan Kaufmann, San Francisco, CA, 2000.

[15] G. Weiss, K. McCarthy, and B. Zabar. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In *Proceedings of the 2007 International Conference on Data Mining*, pages 35–41, Las Vegas, NV, USA, 2007. CSREA Press.

[16] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, California, 2nd edition, 2005.