

A Comparative Study of Sampling Methods and Algorithms for Imbalanced Time Series Classification

Guohua Liang and Chengqi Zhang

The Centre for Quantum Computation & Intelligent Systems, FEIT,
University of Technology Sydney NSW 2007 Australia

{Guohua.Liang, Chengqi.Zhang}@uts.edu.au

<http://www.qcis.uts.edu.au/index.html>

Abstract. Mining time series data and imbalanced data are two of ten challenging problems in data mining research. Imbalanced time series classification (ITSC) involves these two challenging problems, which take place in many real world applications. In the existing research, the structure-preserving over-sampling (SOP) method has been proposed for solving the ITSC problems. It is claimed by its authors to achieve better performance than other over-sampling and state-of-the-art methods in time series classification (TSC). However, it is unclear whether an under-sampling method with various learning algorithms is more effective than over-sampling methods, e.g., SPO for ITSC, because research has shown that under-sampling methods are more effective and efficient than over-sampling methods. We propose a comparative study between an under-sampling method with various learning algorithms and over-sampling methods, e.g. SPO. Statistical tests, the Friedman test and post-hoc test are applied to determine whether there is a statistically significant difference between methods. The experimental results demonstrate that the under-sampling technique with KNN is the most effective method and can achieve results that are superior to the existing complicated SPO method for ITSC.

Keywords: Imbalanced Time Series Classification, Supervised Learning Algorithms, Under-sampling, Over-sampling

1 Introduction

The problems of mining time series data and imbalanced data are two of ten challenging problems in data mining research [1], which have captured the interest and attention of the data mining and machine learning communities for almost two decades. Imbalanced time series classification (ITSC) involving these two problems can be widely observed in many real-world applications in various domains.

ITSC refers to training examples of time series classification (TSC) which are unevenly distributed with unequal cost among classes [2]. ITSC involves many real world applications, such as ECG beats classification [3, 4]. The various challenges, i.e., high dimensionality, large scale, uneven distribution with different

costs for mis-classification errors between different classes, and the nature of numerical attributes considered as a whole instead of individual numerical attributes, such as the sequence of attributes carrying information with a special connection between them [5], mean that many supervised learning algorithms are not effective for ITSC [2]. Many techniques have been proposed for TSC, such as one-nearest-neighbor (1NN) with Dynamic Time Warping (DTW) [6] because the distance measure has proven “exceptionally difficult to beat” [7], but the drawback is that the computational cost is too high [7, 8].

Research into imbalanced data has attracted growing attention in several communities, particularly two major workshops on Learning from Imbalanced Data Sets, AAAI’00 [9] and ICML’03 [10], and a special issue in ACM SIGKDD Explorations’04 [11]. Imbalanced class distribution refers to the numbers of training instances and/or the costs of mis-classification errors are unevenly distributed among different classes [12]. Most traditional supervised learning algorithms have drawbacks with regard to highly imbalanced class distribution. In addition, the overall accuracy/misclassification error rate is an ineffective evaluation metric for imbalanced class distribution data, because it cannot represent the accuracy of minority class, which is the users interested class [2, 12–16]. Many researchers have attempted a number of ways to improve the performance of the prediction model for the imbalanced class distribution problem, at data level, algorithm level, in cost-sensitive learning and ensemble learning. Re-sampling techniques have been the most commonly used techniques to solve imbalanced classification problems at data level, from the simple random under-sampling and over-sampling methods to advanced sampling techniques such as SMOTE [17], SMOTEBoost [18], and Borderline SMOTE [19]. The main advantage of under-sampling methods is that they significantly reduce the computational cost of training a classification model, because only a proportion of majority class instances are selected to train a classification model. Previous works comparing under-sampling and over-sampling methods with Decision Tree learner C4.5 indicate that under-sampling is more effective than over-sampling [20–22].

In the existing research, a structure-preserving over-sampling (SPO) method with support vector machines (SVM) has been proposed for solving the ITSC problem. Its authors claim that it achieves better performance than other over-sampling methods and state-of-the-art methods in TSC [23]. However, the authors have not compared it with under-sampling methods for ITSC; their claim is based on a comparison of the average values of two evaluation metrics, F -value and Geometric mean (G -mean), without statistical analysis to support their conclusion. In addition, evaluating the performance of multiple methods over multiple data-sets to draw valid conclusions is a challenging issue in analyzing experimental results in data mining research.

Our previous work proposed an under-sampling technique integrated with SVM; we observed that our previous proposed method is more efficient than other more complicated approaches, such as SPO with SVM for ITSC [2]. However, it is unclear whether the under-sampling method with various supervised learning algorithms is more effective than over-sampling methods, SPO, and the under-sampling technique integrated with SVM for ITSC.

These issues have motivated us to conduct a comparative evaluation of the performance of over-sampling methods (e.g., the complex SPO [23]) and under-sampling with different supervised learning algorithms for ITSC. Statistical tests are adopted to validate our conclusions; moreover, it provides a correction of the claim for the SPO method by using statistical analysis. As the overall accuracy/mis-classification error rate is ineffective for imbalanced classification, this work adopts two evaluation metrics: F -value and G -mean. The experimental results demonstrate that the under-sampling technique with KNN is the most effective method and can achieve results that are superior to the existing complicated SPO method for ITSC.

The paper is organized as follows. Section 2 presents an outline of the designed framework. Section 3 presents the evaluation metrics. Sections 4 and 5 provide the experimental setting and experimental analysis. Section 6 concludes this work.

2 Designed Framework

Fig. 1 presents the designed framework. The comparative study is evaluated as follows:

1. For each imbalanced binary class data-set, the random under-sampling method is used to alter the levels of class distribution:
 - Firstly, all the positive examples are distributed equally to a set of five subsets.
 - Secondly, negative examples are randomly allocated to each subset according to the ratio of positive and negative examples (positive:negative) as follows: (30%:70%), (40%:60%), (50%:50%), (60%:40%), and (70%:30%).

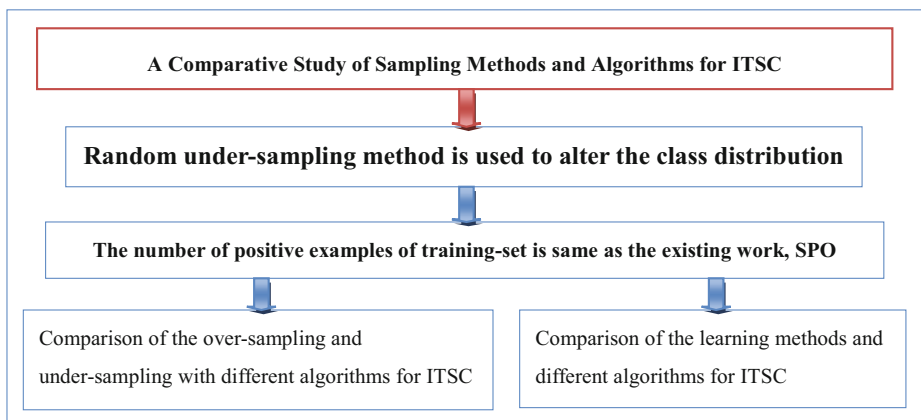


Fig. 1. Designed framework

As a result, each original imbalanced data-set is altered to become five sub-sets with the same number of positive examples and five different ratios of negative examples.

2. The number of positive and negative examples in the altered data-sets in Table 2, which are the same as those in the data-sets of the existing work [23], are used for the evaluation of the performance of each learning algorithm on each subset, based on two evaluation metrics, $F-value$ and $G-mean$. As a result, each learning algorithm has the outputs of five pairs of $F-value$ and $G-mean$ on each original data-set. Only the output pair with the maximum $G-mean$ is selected for reporting.
3. A comparison is made of the performance of the over-sampling methods, SPO, and under-sampling method with five learning algorithms.
4. A comparison is made of the performance of the state-of-the-art learning methods, SPO, and under-sampling with five learning algorithms.
5. Statistical tests, the Friedman test and post-hoc Nemenyi test are applied to determine whether there is a statistically significant difference between methods.

3 Evaluation Metrics

The estimated misclassification error rate/overall accuracy is commonly used as an evaluation metric to assess the performance of a learning algorithm, but it is an ineffective evaluation metric for the imbalanced classification task. This is especially true for ITSC, because it cannot present a true prediction for the minority class, which normally has a higher misclassification error cost than the majority class. Therefore, we have adopted two evaluation metrics for this study as follows: $F-value$ and $G-mean$.

Table 1 presents a confusion matrix, which is used to evaluate the performance of machine learning algorithms; the columns represent the predicted class, and the rows represent the actual class. In the confusion matrix, True Positives (TP) and True Negatives (TN) denote the number of examples correctly classified as positive and negative, respectively. False Positives (FP) and False Negatives (FN) denote the number of misclassified negative examples and positive examples, respectively.

True Positive Rate (TPR) and True Negative Rate (TNR) refer to the proportion of the positive samples and negative samples that have been correctly classified as a positive class and negative class, respectively.

Table 1. Confusion matrix for a binary classification problem

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

F - value is integrated from both recall and precision into a single number, and represents a harmonic mean between recall and precision [24]. As a result, it is closed to one of the smaller number of two (recall and precision). This measure is concerned with the performance of the positive class, so a high F -value indicates that high results in both precision and recall are achieved. On the other hand, the G - mean measure considers the performance of a learning algorithm between two classes to monitor TPR and TNR . The formulas of the evaluation metrics are as follows:

$$TPR = \frac{TP}{TP + FN} \quad (1)$$

$$TNR = \frac{TN}{TN + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$F - value = \frac{2recall * precision}{recall + precision} \quad (5)$$

$$G - mean = \sqrt{TPR * TNR} \quad (6)$$

4 Experimental Setup

This section includes data-set characteristics and the selection of the five learning algorithms. Java platform is used to implement the under-sampling technique to alter the ratio between the positive samples and negative samples. SPSS software is used for the calculation of the Friedman test.

4.1 Data-Sets

Table 2 displays a summary of the characteristics of the five time series data-sets from the public UCR time series repository [25], which were used as the benchmark data-sets of SPO [23]. The first column indicates the index and name of each data-set; the second column presents the data information of the original data-sets, and the altered data-sets (with the number of positive, negative examples, and the ratio between the positive and negative classes), which has the same setting as the existing work [23]; and the last column indicates the class information of the original and the altered data-sets. We also alter 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 a positive class, and the remaining samples are considered as a negative class. For FaceAll and S-Leaf data-sets, the first class is considered as the positive class with 112 and 75 samples, respectively.

Table 2. Time series data-sets

Data-sets		Data Information					Class Information	
Index	Name	TS	Instances		N^-	Ratio P^+/N^-	Previous Class	Altered class
		Length	$(P^+ \& N^-)$	P^+				
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

4.2 Selection of Learning Algorithms

Five learning algorithms are selected from WEKA [26] for this study: Sequential Minimal Optimization (SMO) of SVM, Decision Tree (J48), Random Tree (RTree), KNN, and Multi-layer Proceptron (MLP).

5 Experimental Results Analysis

This section contains two subsections: 5.1 comparison of the performance of over-sampling methods, SPO, and the under-sampling method with different learning algorithms on ITSC; and 5.2 comparison of the performance of other learning methods, SPO, and under-sampling methods with different algorithms for ITSC.

Demar [27] suggests that it is inappropriate to validate the conclusions by using the averaged results over multiple data-sets when the performances of multi-methods are compared. The main reason is that the averages are susceptible to outliers. For example, the KNN algorithm with the under-sampling method has an excellent $F-value$ performance (0.999 on one data-set, wafer) to compensate for other bad $F-value$ performances. It is preferable for the classifiers to perform well on as many problems as possible, which is why it is inappropriate to draw conclusions by averaging the results over multiple data-sets. Previous authors [23] have based their conclusions on the averaged results of the $F-value$ and $G-mean$; thus, their conclusions cannot be validated. This work therefore utilizes statistical tests, the Friedman test and post-hoc Nemenyi test to compare the performance of the multiple learning methods on multiple data-sets, as suggested by [27].

5.1 Comparison of the Performance of Over-Sampling Methods and Different Learning Algorithms with the Under-Sampling Method

Table 3 presents a comparison of the performance of over-sampling methods and five learning algorithms with the under-sampling method based on the $F-value$ and $G-mean$ metrics. The experimental results indicate that KNN with the under-sampling method achieves better performance with $F-value$ than all

Table 3. Comparison of the performance of over-sampling and under-sampling methods with different learning algorithms based on the evaluation metrics $F - value$ and $G - mean$

Metrics	Data-set	Results from Previous Research [23]						Results from This Work					
	Name	Over-sampling Methods						Under-sampling					
		REP	SMO	BoS	ADA	DB	SPO	SVM	J48	RTree	KNN	MLP	
$F - value$	Adiac	0.375	0.783	0.783	0.783	0.136	0.963	0.967	0.883	0.903	0.918	0.947	
	S-Leaf	0.761	0.764	0.764	0.759	0.796	0.796	0.841	0.820	0.849	0.836	0.786	
	Wafer	0.962	0.968	0.968	0.967	0.977	0.982	0.891	0.929	0.956	0.999	0.933	
	FaceAll	0.935	0.935	0.935	0.935	0.890	0.936	0.957	0.876	0.863	0.909	0.919	
	Yoga	0.710	0.729	0.721	0.727	0.689	0.702	0.744	0.771	0.811	0.807	0.780	
	AverageValue	0.740	0.836	0.834	0.834	0.698	0.876	0.880	0.856	0.876	0.894	0.873	
	AverageRank	8.3	6.3	6.7	7.1	7.9	4.3	4	6.8	5.2	3.6	5.8	
$G - mean$	Adiac	0.480	0.831	0.831	0.831	0.748	0.999	0.957	0.910	0.920	0.958	0.975	
	S-Leaf	0.800	0.861	0.861	0.849	0.898	0.898	0.902	0.809	0.812	0.887	0.856	
	Wafer	0.965	0.969	0.970	0.970	0.980	0.984	0.903	0.907	0.956	0.998	0.937	
	FaceAll	0.950	0.950	0.950	0.950	0.948	0.957	0.966	0.870	0.860	0.929	0.925	
	Yoga	0.741	0.756	0.750	0.755	0.724	0.735	0.630	0.807	0.803	0.808	0.774	
	AverageValue	0.787	0.783	0.872	0.871	0.860	0.915	0.872	0.861	0.870	0.916	0.893	
	AverageRank	8.3	5.8	5.9	6.2	6.5	3.3	5.6	7.6	7.2	3.4	6.2	

over-sampling methods and other learning algorithms with the under-sampling method on average value and average rank of $F - value$; while KNN achieves 0.894 and 3.6, respectively, which is better than all over-sampling methods and all learning algorithms with the under-sampling method.

On average value and average rank of the $G - mean$ metric, however, the SPO over-sampling method achieves 0.915 and 3.3, respectively, which is the best among all the over-sampling and under-sampling methods on average rank of the $G - mean$ metric, whereas KNN with the under-sampling method achieves 0.916 and 3.4, respectively, which is the best among all algorithms with the under-sampling method and all over-sampling methods on average of the $G - mean$ metric. The results highlighted in red indicate the correction of the previous work [23].

Figs 2 and 3 present a comparison of over-sampling and under-sampling methods with the Nemenyi test, where the x -axis indicates the ranking order of the sampling methods; 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 not significantly different at a 95% confidence interval are indicated when the vertical bars overlap. The results indicate that there is no statistically significant difference between the over-sampling method and the under-sampling method, based on both $F - value$ and $G - mean$ metrics, even though KNN and SPO show better average rank in $F - value$ and $G - mean$, respectively.

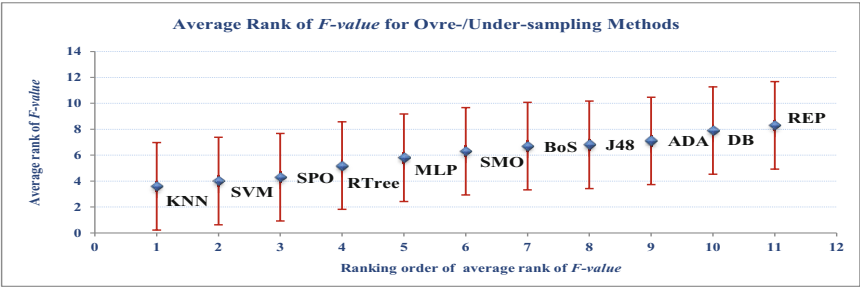


Fig. 2. Comparison of average rank of the $F - value$ with the Nemenyi test for the over-sampling and under-sampling methods, where the x -axis indicates the ranking order of all the sampling methods and learning algorithms, the y -axis indicates the average rank of the $F - value$, and the vertical bars indicate the “Critical Difference”

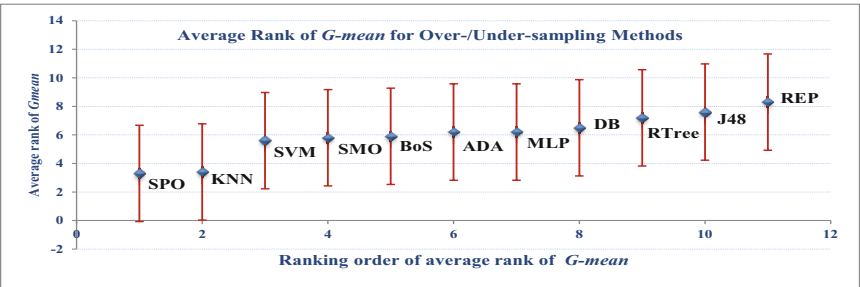


Fig. 3. Comparison of average rank of the $G - mean$ with the Nemenyi test for all the over-sampling and under-sampling methods, where the x -axis indicates the ranking order of all the sampling methods and learning algorithms, the y -axis indicates the average rank of the $G - mean$, and the vertical bars indicate the “Critical Difference”

Comparing SPO, KNN, and the other over-sampling and under-sampling methods, the complex over-sampling method SPO, and simple under-sampling method KNN are not statistically significantly better than any over-sampling methods and learning algorithms with the under-sampling method based on both $F - value$ and $G - mean$. Therefore, the statistical tests demonstrate that there is no statistically significant difference between SPO, KNN and the other over-sampling methods and learning algorithms with under-sampling, even though SPO and KNN have the best average rank in $G - mean$ or $F - value$.

5.2 Comparison of the Performance Learning Methods, Over-Sampling SPO, and Under-Sampling Method with Five Learning Algorithms

Table 4 presents a comparison of the performance of learning methods (results from previous research [23]) and five learning algorithms using the

Table 4. Comparison of the performance of learning methods from previous research [23] and five learning algorithms using the under-sampling method from this work based on evaluation metrics: F – *value* and G – *mean*

Metrics	Data-set	Results from Previous Research [23]					Results from This Work				
	Name	Learning Methods					Under-sampling				
		Easy	Bal.	1NN	1NN_DW	SPO	SVM	J48	RTree	KNN	MLP
F – <i>value</i>	Adiac	0.534	0.348	0.800	0.917	0.963	0.967	0.883	0.903	0.918	0.947
	S-Leaf	0.521	0.578	0.716	0.429	0.796	0.841	0.820	0.849	0.836	0.786
	Wafer	0.795	0.954	0.949	0.857	0.982	0.891	0.929	0.956	0.999	0.933
	FaceAll	0.741	0.625	0.802	0.959	0.936	0.957	0.876	0.863	0.909	0.919
	Yoga	0.356	0.689	0.652	0.710	0.702	0.744	0.771	0.811	0.807	0.780
	AverageValue	0.589	0.639	0.784	0.774	0.876	0.880	0.856	0.876	0.894	0.873
	AverageRank	9.4	8	7.4	6.2	3.8	3.6	5.6	3.6	3	4.4
G – <i>mean</i>	Adiac	0.782	0.897	0.875	0.920	0.999	0.957	0.910	0.920	0.958	0.975
	S-Leaf	0.721	0.898	0.798	0.572	0.898	0.902	0.809	0.812	0.887	0.856
	Wafer	0.817	0.970	0.953	0.870	0.984	0.903	0.907	0.956	0.998	0.937
	FaceAll	0.792	0.918	0.983	0.985	0.957	0.966	0.870	0.860	0.929	0.925
	Yoga	0.464	0.688	0.695	0.741	0.735	0.630	0.807	0.803	0.808	0.774
	AverageValue	0.713	0.874	0.861	0.818	0.915	0.872	0.861	0.870	0.916	0.893
	AverageRank	9.8	5.5	6.2	6.1	2.7	6.1	6.2	5.5	2.4	4.5

under-sampling method (results from this work) based on F – *value* and G – *mean* evaluation metrics. The experimental results indicate that KNN using the under-sampling method achieves better performance of F – *value* and G – *mean* than the remaining learning methods and learning algorithms using the under-sampling method on average value and average rank of F – *value* and G – *mean*. For F – *value* and G – *mean* metrics, 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 best among all the remaining learning methods and learning algorithms using the under-sampling method. The results highlighted in red indicate the correction of the previous work [23].

Figs 4 and 5 present a comparison of learning methods (results from previous research) and five learning algorithms using the under-sampling method with the Nemenyi test, where the x -axis indicates the ranking order of the learning methods and learning algorithms; 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 and learning algorithms that are not significantly different at a 95% confidence interval are indicated when the vertical bars overlap. The statistical test results demonstrate that over-sampling SPO and under-sampling KNN are statistically significantly better than the learning method, Easy; however, there is no statistically significant difference between over-sampling SPO, under-sampling KNN, and the remaining learning methods and algorithms based on the average rank of both the F – *value* and G – *mean* metrics.

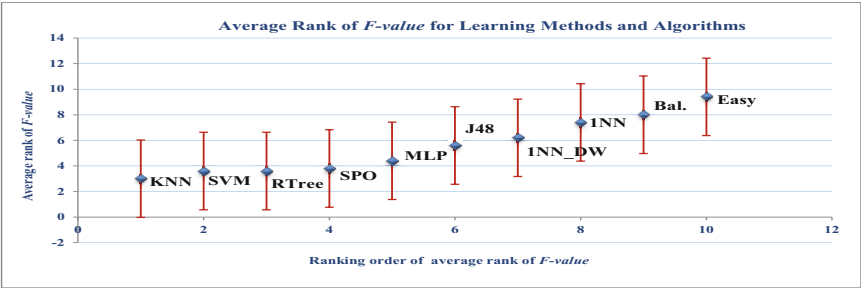


Fig. 4. Comparison of average rank of the F – value metric with the Nemenyi test for the learning methods and five learning algorithms using the under-sampling method, where the x -axis indicates the ranking order of all the learning methods and learning algorithms, the y -axis indicates the average rank of F – value, and the vertical bars indicate the “Critical Difference”

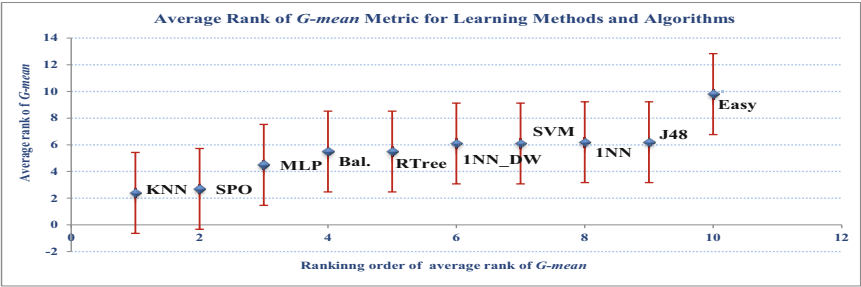


Fig. 5. Comparison of average rank of the G – mean metric with the Nemenyi test for the learning methods and five learning algorithms using the under-sampling method, where the x -axis indicates the ranking order of all the learning methods and learning algorithms, the y -axis indicates the average rank of G – mean, and the vertical bars indicate the “Critical Difference”

6 Conclusion

This study empirically evaluates the performance of complex over-sampling SPO and simple under-sampling with five learning algorithms for ITSC, based on two evaluation metrics, F – value and G – mean. The existing over-sampling techniques generate more synthetic samples to balance training sets to improve prediction models, while the under-sampling method selects a part of negative samples for training. The drawback of the over-sampling method is that it is considered to increase computational cost in training classifiers. The advantages of the under-sampling method is that it is considered to be faster and to have less computational cost than the over-sampling method for training the prediction model. These issues have motivated us to investigate whether different learning algorithms combining a random under-sampling method with different

ratios between positive samples and negative samples in the training set outperform the complex over-sampling SPO. The experimental results indicate that simple under-sampling KNN achieves better results on average for both evaluation metrics F – *value* and G – *mean*, and achieves better results in average rank for the G – *mean* metric. However, when we apply statistical tests to analyze the results, we find that there is no statistically significant difference between the complex over-sampling SPO and the simple under-sampling KNN; the over-sampling SPO and under-sampling KNN are statistically significantly better than the learning method, Easy; and there is no statistically significant difference between the remaining learning methods and algorithms for ITSC. Therefore, the experimental results demonstrate that the simple under-sampling KNN can achieve results that compare favorably with the existing complicated SPO method for ITSC.

References

1. Yang, Q., Wu, X.: 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making* 5(4), 597–604 (2006)
2. Liang, G., Zhang, C.: An efficient and simple under-sampling technique for imbalanced time series classification. In: *CIKM 2012* (in press, 2012)
3. Acr, N.: 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 (2005)
4. Übeyli, E.: ECG beats classification using multiclass support vector machines with error correcting output codes. *Digital Signal Processing* 17(3), 675–684 (2007)
5. Hidasi, B., Gáspár-Papanek, C.: ShiftTree: An Interpretable Model-Based Approach for Time Series Classification. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) *ECML PKDD 2011, Part II*. LNCS, vol. 6912, pp. 48–64. Springer, Heidelberg (2011)
6. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing* 26(1), 43–49 (1978)
7. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.: Fast time series classification using numerosity reduction. In: *Proceedings of 23rd International Conference in Machine Learning, ICML 2006*, pp. 1033–1040 (2006)
8. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: INSIGHT: Efficient and Effective Instance Selection for Time-Series Classification. In: Huang, J.Z., Cao, L., Srivastava, J. (eds.) *PAKDD 2011, Part II*. LNCS, vol. 6635, pp. 149–160. Springer, Heidelberg (2011)
9. Japkowicz, N., et al.: Learning from imbalanced data sets: A comparison of various strategies. In: *AAAI Workshop on Learning from Imbalanced Data Sets*, vol. 68 (2000)
10. Chawla, N., Japkowicz, N., Kolcz, A.: *Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Data Sets* (2003)
11. Chawla, N., Japkowicz, N., Kotcz, A.: Editorial: Special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter* 6(1), 1–6 (2004)
12. Liang, G., Zhu, X., Zhang, C.: The effect of varying levels of class distribution on bagging with different algorithms: An empirical study. *International Journal of Machine Learning and Cybernetics* (in press, 2012)

13. Liang, G.: An investigation of sensitivity on bagging predictors: An empirical approach. In: 26th AAAI Conference on Artificial Intelligence, pp. 2439–2440 (2012)
14. Liang, G., Zhang, C.: An Empirical Evaluation of Bagging with Different Algorithms on Imbalanced Data. In: Tang, J., King, I., Chen, L., Wang, J. (eds.) ADMA 2011, Part I. LNCS, vol. 7120, pp. 339–352. Springer, Heidelberg (2011)
15. Liang, G., Zhang, C.: Empirical study of bagging predictors on medical data. In: 9th Australian Data Mining Conference, AusDM 2011, pp. 31–40 (2011)
16. Liang, G., Zhu, X., Zhang, C.: An Empirical Study of Bagging Predictors for Imbalanced Data with Different Levels of Class Distribution. In: Wang, D., Reynolds, M. (eds.) AI 2011. LNCS, vol. 7106, pp. 213–222. Springer, Heidelberg (2011)
17. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16(1), 321–357 (2002)
18. Chawla, N., Lazarevic, A., Hall, L., Bowyer, K.: SMOTEBoost: Improving Prediction of the Minority Class in Boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 107–119. Springer, Heidelberg (2003)
19. Han, H., Wang, W., Mao, B.: Borderline-SMOTE: A New Over-Sampling Method in Imbalanced Data Sets Learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
20. Drummond, C., Holte, R., et al.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Proceedings of the ICML 2003 Workshop on Learning from Imbalanced Datasets II (2003)
21. Liu, X., Wu, J., Zhou, Z.: Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 39(2), 539–550 (2009)
22. Ling, C., Li, C.: Data mining for direct marketing: Problems and solutions. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 73–79 (1998)
23. Cao, H., Li, X., Woon, Y., Ng, S.: SPO: Structure preserving oversampling for imbalanced time series classification. In: : Proceedings of the IEEE 11th International Conference on Data Mining, ICDM 2011, pp. 1008–1013 (2011)
24. Tan, P., Steinbach, M., Kumar, V., et al.: Introduction to data mining. Pearson, Addison Wesley (2006)
25. Keogh, E., Zhu, Q., Hu, B., Hao, Y., Xi, X., Wei, L., Ratanamahatana, C.A.: UCR Repository of time series classification/clustering homepage, http://www.cs.ucr.edu/~eamonn/time_series_data/ (2011)
26. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tool and Techniques. Morgan Kaufmann (2005)
27. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)