Creating ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set

D. Anyfantis
Dept. of Mathematics, University of Patras, Greece
E-mail: dany@math.upatras.gr

M. Karagiannopoulos
Dept. of Mathematics, University of Patras, Greece
E-mail: mariosk@math.upatras.gr

Sotiris Kotsiantis
Dept. of Computer Science and Technology, University of Peloponnese, Greece
E-mail: sotos@math.upatras.gr

Panayotis Pintelas
Dept. of Computer Science and Technology, University of Peloponnese, Greece.
E-mail: pintelas@math.upatras.gr

***Abstract*—A classifier induced from an imbalanced data set has, characteristically, a low error rate for the majority class and an undesirable error rate for the minority class. This paper firstly provides a systematic study on the various methodologies that have tried to handle this problem. Finally, it presents an experimental study of these methodologies with a proposed method that creates ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set. Our method seems to permit improved identification of difficult small classes in predictive analysis, while keeping the classification ability of the majority class in an acceptable level. abstract.**

***Index Terms*—machine learning, data mining, classification algorithms**

I. Introduction

Inductive classifiers are normally designed to minimize errors over the training examples. Learning algorithms, because of the fact that the cost of performing well on the over-represented class outweighs the cost of poor accuracy on the smaller class, can ignore classes containing few examples. In addition, the difficulty to distinguish between the rare cases (i.e., true exceptions) and noise is also in charge for poor performance on the minority class [14].

For a number of application domains, a massive disproportion in the number of cases belonging to each class is common. For example, in detection of fraud in telephone calls and credit card transactions, the number of legitimate transactions is much higher than the number of fraudulent transactions [8]. Moreover, in direct marketing [18], it is frequent to have a small response rate (about 1%) for most marketing campaigns. Other examples of domains with intrinsic imbalance can be found in the literature such as rare medical diagnoses [25] and oil spills in satellite images [16]. Thus, learning with skewed class distributions is a vital issue in supervised learning.

The machine learning community has mostly addressed the issue of class imbalance in two ways. One is to give distinct costs to training instances [7]. The other is to re-sample the original dataset, either by oversampling the minority class and/or under-sampling the majority class [15], [12]. Although many methods for coping with imbalanced data sets have been proposed, still remain open questions.

To handle the problem, we proposed method that creates ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set. We separate the instances of the majority class in a number of subsets. The number of subsets is equal to the value of the number of instances of the majority class divided by the number of instances of the minority class. In this way, there are a similar number of instances of the majority and minority class in each subset. A similar classifier is then built for each subset. The decisions of the classifiers are then combined with voting for the final decision. The effectiveness of our approach is evaluated over eight imbalanced datasets using the C4.5 [21], Naive Bayes [6] and 5NN [1] as classifiers and the geometric mean of accuracies as performance measure [16].

Section 2 reviews the attempts for handling imbalanced data sets, while section 3 presents the details of our approach. Section 4 presents experimental results comparing our approach to other approaches. Finally, section 5 discusses the results and suggests directions for future work.

II. Review of existing techniques for handling imbalanced data sets

A simple method that can be used to imbalanced data sets is to reweigh training examples according to the total cost assigned to each class [5]. The idea is to change the class distributions in the training set towards the most costly class. Suppose that the examples of the positive class are four times more than the instances of the negative class. If the number of negative instances are artificially increased by a factor of four, then the learning system, aiming to reduce the number of classification errors, will come up with a classifier that is skewed towards the prevention of error in the negative class, since any such errors are penalised four times more.

Japkowicz [11] discussed the effect of imbalance in a dataset. She mainly evaluated two strategies: under-sampling and resampling. Two resampling methods were considered. Random resampling consisted of resampling the smaller class at random until it consisted of as many samples as the majority class and "focused resampling" consisted of resampling only those minority instances that occurred on the boundary between the minority and majority classes. Random under-sampling was also measured, which involved under-sampling the majority class samples at random until their numbers matched the number of minority class samples; focused under-sampling involved under-sampling the majority class samples lying further away. She noted that both the sampling approaches were helpful, and she also observed that using the sophisticated sampling techniques did not give any clear advantage in the domain considered.

Kubat and Matwin [15] also selectively under-sampled the majority class while keeping the original population of the minority class with satisfied results. Batista et al. [2] used a more sophisticated under-sampling technique in order to reduce the amount of potentially useful data. The majority class instances are classified as "safe", "borderline" and "noise" instances. Borderline and noisy cases are detected using Tomek links, and are removed from the data set. Only safe majority class instances and all minority class instances are used for training the learning system. A Tomek link [24] can be defined as follows: given two instances x and y belonging to different classes, and be d(x, y) the distance between x and y. A (x, y) pair is called a Tomek link if there is not a case z, such that d(x, z) < d(x, y) or d(y, z) < d(y, x).

Another approach is that of Ling and Li [18]. They combined over-sampling of the minority class with under-sampling of the majority class. However, the over-sampling and under-sampling combination did not provide significant improvement. Chawla et al. [4] recommend an over-sampling approach in which the minority class is over-sampled by creating ``synthetic'' instances rather than by over-sampling with replacement with better results.

Changing the class distribution is not the only technique to improve classifier performance when learning from imbalanced data sets. A different approach to incorporating costs in decision-making is to define fixed and unequal misclassification costs between classes. Cost model takes the form of a cost matrix, where the cost of classifying a sample from a true class j to class i corresponds to the matrix entry λ_{ij}. This matrix is usually expressed in terms of average misclassification costs for the problem. The diagonal elements are usually set to zero, meaning correct classification has no cost. We define conditional risk for making a decision α_i as:

$$R(a_i \mid x) = \sum_i \lambda_{ij} P(v_j \mid x)$$

The equation states that the risk of choosing class i is defined by fixed misclassification costs and the uncertainty of our knowledge about the true class of x expressed by the posterior probabilities. The goal in cost-sensitive classification is to minimize the cost of misclassification, which can be realized by choosing the class (v_j) with the minimum conditional risk.

An alternative to balancing the classes is to develop a learning algorithm that is intrinsically insensitive to class distribution in the training set. An example of this kind of algorithm is the SHRINK algorithm [16] that finds only rules that best summarize the positive instances (of the small class), but makes use of the information from the negative instances. MetaCost [7] is another method for making a classifier cost-sensitive. The procedure begins to learn an internal cost-sensitive model by applying a cost-sensitive procedure, which employs a base learning algorithm. Then, MetaCost procedure estimates class probabilities using bagging and then re-labels the training instances with their minimum expected cost classes, and finally relearns a model using the modified training set.

III. Proposed Technique

Under-sampling is a class-imbalance learning method which uses only a subset of major class examples and thus is very efficient. The main deficiency is that many major class examples are ignored. We propose an algorithm to overcome the deficiency. The proposed method samples several subsets from the major class, trains a learner using as training set each of these subsets with all the instances of the minority class, and combines the outputs of those learners. In detail, we separate the instances of the majority class in a number of subsets. The number of subsets is equal to the value of the number of instances of the majority class divided by the number of instances of the minority class. In this way, there are a similar number of instances of the majority and minority class in each subset. A similar classifier is then built for each subset. The decisions of the classifiers are then combined with voting for the final decision. The a-posteriori probabilities generated by the individual classifiers are correspondingly denoted p1(i), p2(i) for each output class i. Next, the class represented by the maximum sum value of the a-posteriori probabilities is taken as the voting hypothesis (h*). The predictive class is computed by the rule:

$$predictive\_Class = \arg\max_{i,j} \sum_{i=1,\, j=1}^{i=number\_of\_classes,\, j=number\_of\_subsets} p_j(i)$$

Broadly speaking, techniques for scaling data mining algorithms can be divided into five basic categories: 1) manipulating the data so that it fits into memory, 2) using specialized data structures to manage out of memory data, 3) distributing the computation so that it exploits several processors, 4) precomputing intermediate quantities of interest, and 5) reducing the amount of data mined.

One of the easiest ways to speed up algorithms on large data sets is to use more than one processor. The success depends

upon how easy it is to break up the problem into sub-problems which can be assigned to the different processors. The easiest technique is data parallelism. With the proposed technique, essentially the same classifier is applied to different partitions of the data. If we look at the total number of records that are available per processor at the entire level, the balance among the processors is perfect. Taking advantage of a parallel or a distributed execution a ML system may: i) increase its speed; ii) increase the range of applications where it can be used (because it can process more data, for example).

Moreover, our approach is schematically represented in Figure 1.

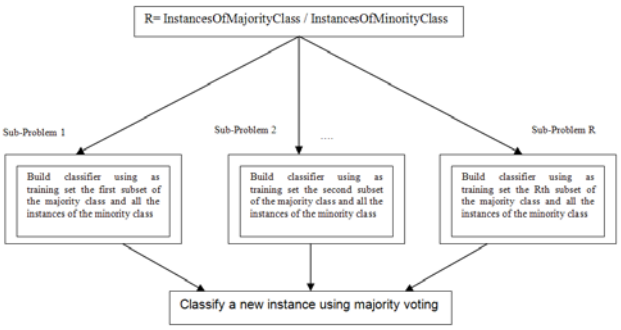

Fig 1. Representation of our agent-based approach

In pseudo-code, the proposed technique - Distributing Imbalances Dataset (DID) - is presented in Fig. 2. A key feature of our method is that it does not require any modification of the underlying learning algorithm. In the following section, we empirically evaluate the performance of our approach with the other well known techniques using a decision tree, an instance base learner and a Bayesian model as base classifiers.

```
RatioA = InstancesOfMajorityClass / InstancesOfMinorityClass;
Instances[] splitted = SplitInstances(RatioA);
methods = new Classifier[splitted.length];

for (int i = 0; i < splitted.length; i++) {
/*ADDING MINORITY INSTANCES DATASET (lessData) TO
SUBSETS OF MAJORITY INSTANCES*/
  Instances mergedData = MergeInstances(splitted[i], lessData);
  method.buildClassifier(mergedData);
  methods[i] = method;
}

Vote vote = new Vote();
vote.setClassifiers(methods); //set the combining classifiers
Instances TestCV= tmpData.testCV; // obtaining test cross validation  set
eval.evaluateModel(vote, TestCV);
```

Fig 2. Creating ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set

A similar approach (but not distributed) was proposed by [17]. During classification, each of the N classifiers generates one class label as its vote, and the majority vote from the N classifiers is used as the classification of the new data instead of the maximum sum value of the a-posteriori probabilities that is used by our method. Liu et al. [19] also propose two similar algorithms: EasyEnsemble and BalanceCascade. EasyEnsemble samples several subsets from the major class with the use of boosting technique, trains a learner using each of them, and combines the outputs of those learners. BalanceCascadeis similar to EasyEnsemble except that it removes correctly classified major class examples of trained learners from further consideration.

IV. Experiments

For the aim of our study the most well-known decision tree algorithm - C4.5 [21] – was used. One of the latest researches that compare decision trees and other learning algorithms is made by [23] and shows that the mean error rates of most algorithms are similar and that their differences are statistically insignificant. But, unlike error rates, there are huge differences between the training times of the algorithms. C4.5 has one of the best combinations of error rate and speed. Decision tree classifiers, regularly, employ post-pruning techniques that evaluate the performance of decision trees as they are pruned using a validation set. Any node can be removed and assigned the most common class of the training examples that are sorted to the node in question. As a result, if a class is rare, decision tree algorithms often prune the tree down to a single node that classifies all instances as members of the common class leading to poor accuracy on the examples of minority class.

The extreme skewness in class distribution is problematic for Naïve Bayes [6]. The prior probability of the majority class overshadows the differences in the attribute conditional probability terms. Instance-based learning algorithms belong to the category of lazy-learning algorithms [25], as they delay the induction until classification is performed. One of the most straightforward instance-based learning algorithms is the nearest neighbour algorithm [1]. In our study, we made use of the commonly used 5-NN algorithm. In imbalanced data sets as the number of the instances of the majority class grows, so does the likelihood that the nearest neighbour of any instance will belong to the majority class. This leads to the problem that many instances of the minority class will be misclassified.

In Table 1, there is a brief description of the data sets that we used for our experiments. Except for the “eap” data set, all were drawn from the UC Irvine Repository [3]. Eap data is from Hellenic Open University and was used in order to determine whether a student is about to drop-out or not [14]. The data sets from UC Irvine Repository are from domains of: image recognition (ionosphere), medical diagnosis (breast-cancer, diabetes, haberman, hepatitis, sick) and commodity trading (credit-g).

TABLE I
DESCRIPTION OF THE DATA SETS

Datasets	Instances	Categorical Features	Numerical Features	Missing Values	Classes
breast-cancer	286	9	0	85	2
credit-g	1000	13	7	300	2

Diabetes	768	0	8	268	2
Haberman	306	0	3	81	2
Hepatitis	155	13	6	32	2
Ionosphere	351	34	0	126	2
Eap	344	11	0	122	2
Sick	3772	22	7	231	2

A classifier's performance of two class problems can be separately calculated for its performance over the positive instances (denoted as α+) and over the negative instances (denoted as α-). The true positive rate (α+) or sensitivity is the fraction of positive instances predicted correctly by the model. Similarly, the true negative rate (α-) or specificity is the fraction of negative instances predicted correctly by the classifier.

Kubat et al. [16] propose the geometric mean of the accuracies: for imbalanced data sets. The basic idea behind this measure is to maximize the accuracy on both classes. Moreover, ROC curves (Receiving Operator Characteristic) provide a visual representation of the trade off between true positives (α+) and false positives (α-). These are plots of the percentage of correctly classified positive instances α+ with respect to the percentage of incorrectly classified negative instances α- [20]. If the model is perfect, then its area under the ROC curve would equal to 1. If the model corresponds to random guessing, then its area under ROC curve would be equal to 0.5. Anything less than 0.5 would be worse than random guessing.

The most popular method for plotting a ROC curve is threshold variation [25]: given a set of test instances and a classifier, the numeric output for each test instance is computed, and the instances are ordered according to the corresponding numeric prediction. Then, for each instance, a (1-α+, α+) point is obtained, that is, considering that instances before it are classified as positive and instances after it are classified as negative. Subsequent (1-α+, α+) points are linked. The method for plotting a ROC curve is closely related to a method for making algorithms cost-sensitive, that we call Threshold method (Witten and Frank, 2005). This method uses a threshold so as to maximize the given performance measure in the curve. For the examined cost models, the relationship between false negative and false positive costs was chosen to be the inverse of the assumed prior to compensate for the imbalanced priors.

Classification ability of the learning methods in our experiments was measured with geometric mean of the accuracies. In the following Tables, win (v) indicates that the specific method along with the learning algorithm performed statistically better than the single classifier according to t-test with $p<0.05$. Loss (*) indicates that the specific method along with the learning algorithm performed statistically worse than the single classifier according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results. In Table 2, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data set using Naive Bayes (NB) as base classifier. Five well-known algorithms were used for the comparison: Threshold method [25], Reweighing and Cost Sensitive method [5], Adaboost cost sensitive method [22] and Metacost algorithm (Domingos, 1999). We also present the accuracy of the simple Bayes algorithm as borderline. It must be mentioned that we used the free available source code for these methods by [25] for our experiments. In the Table 2 except for geometric mean we also present the true-positive rate, and true-negative rate. It must be mentioned that positive class is the majority class for our experiments. In the last row of the Table 2, the mean value of the geometric means is also calculated in all data sets. In general, all the tested techniques give better results than the single Naive Bayes. It must be mentioned that for Naïve Bayes classifier, modifying the decision boundary (Cost Sensitive method) is equivalent to reweighing training instances so as the relationship between false negative and false positive costs to be the inverse of the imbalanced priors. All the tested techniques give similar results however the proposed technique has the advantage that can be easily parallelized and scaled up in large datasets.

In Table 3, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data sets using C4.5 as base classifier. The same five well-known techniques for handling imbalanced data sets were also used for this comparison. The proposed technique, the reweighing method and Metacost algorithm give similar results however the proposed technique has the advantage that can be easily parallelized and scaled up in large datasets.

Similarly to our results, on several experiments performed in (Provost and Fawcett, 2001), decision tree classifiers generated from balanced distributions obtained results that were, frequently, better than those obtained from the naturally occurring distributions. In Table 4, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data sets using 5NN as base classifier. The same five well-known techniques for handling imbalanced data sets were also used for this comparison. The proposed technique, the reweighing method and Cost Sensitive method give similar results however the proposed technique has the advantage that can be easily parallelized and scaled up in large datasets.

V. Conclusion

The problem of imbalanced data sets arises frequently. It is a problem in medical diagnosis, robotics, industrial production processes, communication network troubleshooting, machinery diagnosis, automated testing of electronic equipment, and many other areas. In this work, we survey some methods proposed by the Machine Learning community to solve the problem, we discuss some limitations of these methods and we propose that creating ensembles of classifiers by distributing an imbalance dataset to reach balance in each resulting training set could be a more effective solution to problem.

TABLE II
ACCURACY ON MAJORITY CLASS (A+), ACCURACY ON MINORITY CLASS (A-) AND GEOMETRIC MEAN (G) WITH NB AS BASE CLASSIFIER

Data sets		DIDNB	ReWNB	ThresNB	CostNB	AdabcosNB	MetacostNB	NB
breast-cancer	g	0.64	0.66	0.63	0.66	0.63	0.65	0.6*
	α+	0.74	0.74	0.62 *	0.74	0.72	0.79 v	0.85v
	α-	0.55	0.58 v	0.65 v	0.58 v	0.56	0.54	0.43*
credit-g	g	0.7	0.72	0.71	0.72	0.71	0.66 *	0.65*
	α+	0.77	0.75	0.69 *	0.75	0.75	0.77	0.86v
	α-	0.64	0.69 v	0.74 v	0.69 v	0.67 v	0.57 *	0.49*
diabetes	g	0.73	0.73	0.72	0.73	0.73	0.70 *	0.71
	α+	0.76	0.78	0.65 *	0.78	0.77	0.75	0.84v
	α-	0.70	0.68	0.8	0.68	0.69	0.66 *	0.6 *
haberman	g	0.56	0.56	0.59 v	0.56	0.56	0.57	0.44*
	α+	0.85	0.89 v	0.64 *	0.89 v	0.88 v	0.87	0.94v
	α-	0.38	0.35 *	0.55 v	0.35 *	0.36	0.38	0.21*
hepatitis	g	0.8	0.8	0.76 *	0.8	0.78	0.81	0.78
	α+	0.85	0.83	0.87	0.83	0.86	0.79 *	0.87
	α-	0.75	0.78 v	0.67 *	0.78 v	0.71 *	0.84 v	0.7*
ionosphere	g	0.83	0.82	0.88 v	0.82	0.91 v	0.77*	0.83
	α+	0.78	0.78	0.93 v	0.78	0.93 v	0.68 *	0.8
	α-	0.87	0.87	0.81 *	0.87	0.9 v	0.88	0.86
eap	g	0.84	0.85	0.83	0.85	0.83	0.85	0.84
	α+	0.86	0.87	0.86	0.87	0.85	0.88	0.9 v
	α-	0.82	0.83	0.81	0.83	0.82	0.83	0.78*
sick	g	0.86	0.86	0.76*	0.86	0.87	0.8*	0.86
	α+	0.83	0.82	0.98 v	0.82	0.88 v	0.73 *	0.94v
	α-	0.89	0.9	0.59 *	0.9	0.86 *	0.87	0.78*
MEAN	g	0.75	0.75	0.74	0.75	0.75	0.73	0.71

TABLE III
ACCURACY ON MAJORITY CLASS (A+), ACCURACY ON MINORITY CLASS (A-) AND GEOMETRIC MEAN (G) WITH C4.5 AS BASE CLASSIFIER

Data sets		DIDC4.5	ReWC4.5	ThresC4.5	CostC4.5	Adabcos C4.5	Metacost C4.5	C4.5
breast-cancer	g	0.56	0.57	0.45*	0.5 *	0.56	0.55	0.5 *
	α+	0.82	0.72 *	0.8	0.85 v	0.77 *	0.84	0.95 v
	α-	0.38	0.45 v	0.25 *	0.3 *	0.41 v	0.36	0.26 *
credit-g	g	0.66	0.66	0.64	0.61*	0.62*	0.64	0.58 *
	α+	0.69	0.67	0.7	0.82 v	0.81 v	0.76 v	0.85 v
	α-	0.63	0.65	0.58 *	0.46 *	0.47 *	0.54 *	0.4 *
diabetes	g	0.72	0.72	0.7	0.72	0.67*	0.73	0.7
	α+	0.66	0.72 v	0.69 v	0.78 v	0.79 v	0.78 v	0.82 v
	α-	0.78	0.73 *	0.71 *	0.67 *	0.57 *	0.67 *	0.6 *
haberman	g	0.65	0.63	0.56 *	0.58 *	0.57 *	0.62 *	0.52 *
	α+	0.66	0.68	0.61 *	0.66	0.76 v	0.76 v	0.85 v
	α-	0.63	0.58 *	0.51 *	0.51 *	0.43 *	0.52 *	0.32 *
hepatitis	g	0.75	0.73	0.62 *	0.64 *	0.7 *	0.68 *	0.58 *
	α+	0.79	0.62 *	0.78	0.86 v	0.9 v	0.83 v	0.9 v
	α-	0.72	0.85 v	0.49 *	0.48 *	0.55 *	0.56 *	0.37 *
ionosphere	g	0.87	0.89	0.88	0.88	0.9 v	0.9 v	0.88
	α+	0.89	0.94 v	0.95 v	0.94 v	0.94 v	0.98 v	0.94 v
	α-	0.85	0.85	0.81*	0.82 *	0.86	0.82 *	0.82 *
eap	g	0.8	0.81	0.69 *	0.83 v	0.79	0.82	0.83 v
	α+	0.83	0.86 v	0.91 v	0.94 v	0.85	0.89 v	0.94 v
	α-	0.77	0.77	0.53 *	0.74 *	0.74 *	0.76	0.74 *
sick	g	0.96	0.97	0.92 *	0.96	0.95	0.96	0.93 *
	α+	0.95	0.99 v	0.99 v	0.99 v	1 v	0.98 v	0.99 v
	α-	0.97	0.95	0.85 *	0.92 *	0.9 *	0.95	0.87 *
MEAN	g	0.75	0.75	0.68	0.72	0.72	0.74	0.69

TABLE IV
ACCURACY ON MAJORITY CLASS (A+), ACCURACY ON MINORITY CLASS (A-) AND GEOMETRIC MEAN (G) WITH 5NN AS BASE CLASSIFIER

Data sets		DID5NN	ReW5NN	Thres5NN	Cost5NN	Adabcos5NN	Metacost5NN	5NN
breast-cancer	g	0.6	0.62	0.6	0.61	0.61	0.51 *	0.45*
	α+	0.83	0.73 *	0.57 *	0.72 *	0.7 *	0.86 v	0.96v
	α-	0.44	0.52 v	0.63 v	0.52 v	0.53v	0.3 *	0.21*
credit-g	g	0.65	0.66	0.59 *	0.66	0.63	0.63	0.57 *
	α+	0.71	0.69	0.84 v	0.69	0.7	0.73	0.89 v
	α-	0.6	0.63 v	0.42 *	0.63 v	0.56 *	0.55 *	0.37 *
diabetes	g	0.73	0.71	0.69 *	0.71	0.66 *	0.71	0.68 *
	α+	0.71	0.69	0.79 v	0.69	0.71	0.75 v	0.83v
	α-	0.75	0.74	0.61 *	0.74	0.62 *	0.68 *	0.56*
haberman	g	0.54	0.57 v	0.58 v	0.57 v	0.53	0.59 v	0.39*
	α+	0.59	0.68 v	0.52 *	0.68 v	0.68 v	0.66 v	0.9v
	α-	0.49	0.47	0.65 v	0.47	0.41 *	0.52 v	0.17*
hepatitis	g	0.76	0.69 *	0.68 *	0.73 *	0.58 *	0.8 v	0.66 *
	α+	0.85	0.79 *	0.91 v	0.85	0.8	0.84	0.94 v
	α-	0.68	0.6 *	0.51 *	0.62 *	0.42 *	0.76 v	0.46 *
ionosphere	g	0.81	0.83	0.82	0.83	0.83	0.79	0.78*
	α+	0.98	0.97	0.97	0.97	0.95 *	0.98	0.98
	α-	0.67	0.71 v	0.7 v	0.71 v	0.72 v	0.63 *	0.62 *
eap	g	0.8	0.8	0.79	0.8	0.78	0.77 *	0.78
	α+	0.8	0.84 v	0.83	0.84 v	0.79	0.87 v	0.9 v
	α-	0.8	0.76 *	0.75 *	0.76 *	0.77 *	0.69 *	0.68 *
sick	g	0.86	0.84	0.62 *	0.84	0.87	0.79 *	0.61 *
	α+	0.86	0.89 v	0.99 v	0.89 v	0.98 v	0.9 v	0.99 v
	α-	0.86	0.79 *	0.39 *	0.79 *	0.77 *	0.7 *	0.37*
AVERAGE	g	0.72	0.72	0.67	0.72	0.69	0.7	0.62

REFERENCES

[1] Aha, D. (1997), Lazy Learning, Dordrecht: Kluwer Academic Publishers.

[2] Batista G., Carvalho A., Monard M. C. (2000), Applying One-sided Selection to Unbalanced Datasets. In O. Cairo, L. E. Sucar, and F. J. Cantu, editors, Proceedings of the Mexican International Conference on Artificial Intelligence – MICAI 2000, pages 315–325. Springer-Verlag.

[3] Blake, C., Keogh, E. & Merz, C.J. (1998). UCI Repository of machine learning databases. [http:// www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California.

[4] Chawla N., Bowyer K., Hall L., Kegelmeyer W. (2002), SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16, 321 - 357.

[5] Domingos P. (1998), How to get a free lunch: A simple cost model for machine learning applications. Proc. AAAI-98/ICML98, Workshop on the Methodology of Applying Machine Learning, pp1-7.

[6] Domingos P. & Pazzani M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29, 103-130.

[7] Domingos, P. (1999). MetaCost: A General Method for Making Classifiers Cost-Sensitive. Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining, 155-164. ACM Press.

[8] Fawcett T. and Provost F. (1997), Adaptive Fraud Detection. Data Mining and Knowledge Discovery, 1(3):291–316.

[9] Friedman J. H. (1997), On bias, variance, 0/1-loss and curse-of-dimensionality. Data Mining and Knowledge Discovery, 1: 55-77.

[10] Hongyu Guo, Herna L. Viktor: Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. SIGKDD Explorations 6(1): 30-39 (2004)

[11] Japkowicz N. (2000), The class imbalance problem: Significance and strategies. In Proceedings of the International Conference on Artificial Intelligence, Las Vegas.

[12] Japkowicz N. and Stephen, S. (2002), The Class Imbalance Problem: A Systematic Study Intelligent Data Analysis, Volume 6, Number 5.

[13] Kotsiantis, S., Pierrakeas, C., Pintelas, P., Preventing student dropout in distance learning systems using machine learning techniques, Lecture Notes in Artificial Intelligence, KES 2003, Springer-Verlag Vol 2774, pp 267-274, 2003.

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

[15] Kubat, M. and Matwin, S. (1997), 'Addressing the Curse of Imbalanced Data Sets: One Sided Sampling', in the Proceedings of the Fourteenth International Conference on Machine Learning, pp. 179-186.

[16] Kubat, M., Holte, R. and Matwin, S. (1998), 'Machine Learning for the Detection of Oil Spills in Radar Images', Machine Learning, 30:195-215.

[17] Cen Li: Classifying imbalanced data using a bagging ensemble variation (BEV), ACM Southeast Regional Conference 2007: 203-208

[18] 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 (KDD-98) New York, NY. AAAI Press.

[19] Liu, X., Jianxin Wu, Zhi-Hua Zhou: Exploratory Under-Sampling for Class-Imbalance Learning. ICDM 2006: 965-969

[20] Provost, F. and Fawcett, T. (2001). Robust Classification for Imprecise Environments", Machine Learning, 42, 203–231.

[21] Quinlan J.R. (1993), C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco.

[22] Schapire R., Singer Y. and Singhal A. (1998). Boosting and Rochhio applied to text filtering. In SIGIR'98.

[23] Tjen-Sien Lim, Wei-Yin Loh, Yu-Shan Shih (2000), A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New

[24] Tomek, I., Two modifications of CNN, IEEE Transactions on Systems Man and Communications, SMC-6. 769-772, 1976.

[25] Ian H. Witten and Eibe Frank (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.