SELECTIVE VOTING

S. B. Kotsiantis
Educational Software Development Laboratory
Department of Mathematics
University of Patras
Greece
sotos@math.upatras.gr

P. E. Pintelas
Educational Software Development Laboratory
Department of Mathematics
University of Patras
Greece
pintelas@math.upatras.gr

Abstract. **The combination of classifiers is interesting and attractive for several reasons. The most important reason might be that the combination of redundant and complementary classifiers promises to increase robustness, accuracy, and generality of the overall classification. The proposed method combines the advantages of classifier fusion and dynamic selection, which are the two main categories of traditional combining algorithms. The algorithms that are initially used for building the ensemble are tested in a small subset of the training set and if they have statistically worse accuracy than the most accurate algorithm, they do not participate to the final decision of the ensemble. Our experiment for several UCI datasets shows that the proposed combining method outperforms other combining methods we tried as well as any base classifier.**

I. INTRODUCTION

The goal of supervised learning is to learn how to classify objects or situations by analyzing a set of instances whose classes are known. Classes are mutually exclusive labels such as medical diagnoses, qualitative economic projections, image categories, or failure modes. Instances are typically represented as feature-value vectors that give the numerical or nominal values of a fixed collection of properties. Learning input consists of a set of such vectors, each belonging to a known class, and the output consists of a mapping from feature values to classes. This mapping should accurately classify both the given instances and other unseen instances.

Given that no one classification method is the best in all tasks, a variety of approaches have evolved to prevent poor performance due to mismatch of capabilities [17]. One approach to overcome this problem is to determine when a method may be appropriate for a given problem by selecting the best classifier according to cross validation (BestCV) [12]. A second, more popular approach is to combine the capabilities of two or more classification methods.

The success of the techniques that combine classifiers comes from their ability to reduce the bias error as well as the variance error. The bias error is caused when there is a mismatch between the structure of the problem and the structure (representation) reasoned upon by a classifier. If each of the base classifiers uses a different representation, then the chance of finding a representation that matches the problem increases.

At present, most ensemble algorithms utilize all the trained learners to make up an ensemble. This paper shows that it may be better to build selective ensembles, that is, ensembles containing some instead of all of the trained classifiers. The algorithms that are initially used for building the ensemble are tested in a small subset of the training set and if they have statistically worse accuracy than the most accurate algorithm, they do not participate to the final decision of the ensemble. Our experiment for several UCI datasets shows that the proposed combining method (Selective Voting) outperforms other well known ensembles we tried as well as any base classifier.

In the next section, we discuss the current ensemble approaches. In Section 3 we describe the proposed method and investigate its advantages and limitations. In Section 4, we evaluate the proposed method on UCI datasets by comparing it with base classifiers and other well known ensembles. Finally, section 5 concludes the paper and suggests further directions in current research.

II. ENSEMBLES OF CLASSIFIERS USING DIFFERENT LEARNING ALGORITHMS

The use of multiple classifiers has gained momentum in the recent years and researchers have continuously argued for the benefits of using multiple classifiers to solve complex recognition problems. The main motivation for combining classifiers is based upon the assumption that different classifiers using different data representations, different concepts and modeling techniques are likely to arrive at classification results with different patterns of generalisation. Visible evidence of such multimodal diversity among classifiers takes the form of different occurrences of classification errors for different classifiers reported over a set of input instances. As most combination functions benefit from disagreement to errors of individual classifiers, the greater this disagreement, the lower the impact of individual errors on the final decision, and effectively the lower the combined classification error.

The simplest approach for building ensembles is to use a variety of learning algorithms on all of the training data and combine their predictions according to a voting scheme. Among the combination methods, majority vote is the simplest to implement, since it requires no prior training [7].

Stacked generalization [18], or stacking, is another approach. Stacking combines multiple classifiers to induce a higher-level classifier with improved performance. A learning algorithm is used to determine how the outputs of the base classifiers should be combined. The original dataset constitutes the level zero data. All the base classifiers run at this level. The level one data are the outputs of the base classifiers. Another learning process occurs using as input the level one data and as output the final prediction.

The authors of [16] use base-level classifiers whose predictions are probability distributions over the set of class values, rather than single class values. The meta-level attributes are thus the probabilities of each of the class values returned by each of the base-level classifiers. The authors argue that this allows using not only the predictions, but also the confidence of the base-level classifiers. Multi-response linear regression (MLR) was used for meta-level learning. The authors of [5] used model tree induction instead of MLR keeping everything else the same for better results. Recently, other authors modified the method so as to use only the class probabilities associated with the true class [14] and the accuracy seems to be improved (StackingC).

The authors of [13] propose a method for combining classifiers called grading that learns a meta-level classifier for each base-level classifier. The metalevel classifier predicts whether the base-level classifier is to be trusted (i.e., whether its prediction will be correct). The base-level attributes are used also as meta-level attributes, while the meta-level class values are + (correct) and − (incorrect). Only the base-level classifiers that are predicted to be correct are taken and their predictions combined by summing up the probability distributions predicted.

III. PROPOSED ALGORITHM

It is a well-known fact that the selection of an optimal set of classifiers is an important part of multiple classifier systems and the independence of classifier outputs is generally considered to be an advantage for obtaining better multiple classifier systems. In terms of classifier combination, the voting methods demand no prerequisites from the classifiers.

The proposed method relies on the idea of selection in ensemble creation [15] and combines the advantages of classifier fusion and dynamic selection, which are the two main categories of traditional combining algorithms.

The presented methodology is six steps strategy:

- The dataset is sampled at random about 20% of the initial set
- The new dataset is divided at random into three equal parts
- Two of three parts are used for training of algorithms and the remaining data is the testing set
- The result of three tests are averaged
- The algorithms that have statistically worse accuracy (according to t-test with $p<0.05$) than the most accurate are not used by the ensemble
- The remaining algorithms then executes on the full training set to produce the prediction model with simple voting. A detailed study of the working of the majority voting scheme has been presented in [8].

In other words, the classification process includes two phases: (1) *learning phase*, and (2) *application phase*. During the learning phase, a set of base classifiers is generated and each base classifier in the ensemble (classifiers $h1 \ldots h_n$) is trained. For the ensemble classification the corresponding classifications of the base classifiers are combined with selective voting $h^* = F(h1, h2, \ldots, hn)$ to produce the final classification of the ensemble. At the application phase, a new instance (**x**, ?) is given with the unknown value y to be classified by the ensemble. As a result, the class value y^* is then predicted as $y^* = h^*(\mathbf{x})$.

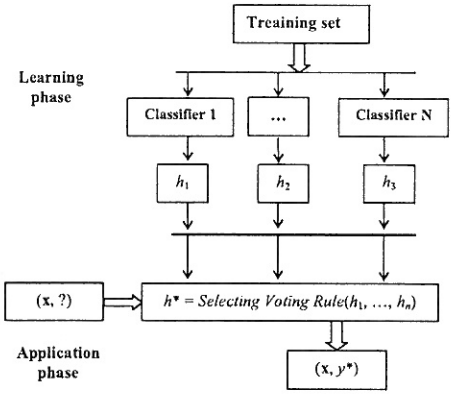

Figure 1. The proposed ensemble

It must also be mentioned that the proposed ensemble can easily be parallelized using a learning algorithm per machine. Parallel and distributed computing is of most importance for ML practitioners because 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).

IV. EXPERIMENTS

We have experimented with 27 datasets from the UCI repository [5]. These datasets cover many different types of problems having discrete, continuous, and symbolic variables. Some datasets have missing values, and some have a mixture of all the above. The datasets are listed in Table 1.

TABLE 1. The used datasets

Datasets	Instances	Categorical features	Numerical features	Classes
autos	205	10	15	6
badges	294	4	7	2
breast-cancer	286	9	0	2
breast-w	699	0	9	2
Credit-a	690	9	6	2
Credit-g	1000	13	7	2
Diabetes	768	0	8	2
glass	214	0	9	6
haberman	306	0	3	2
heart-c	303	7	6	5
heart-h	294	7	6	5
heart-statlog	270	0	13	2
hepatitis	155	13	6	2
ionosphere	351	34	0	2
iris	150	0	4	3
Labor	57	8	8	2
lymphotherapy	148	15	3	4
monk1	124	6	0	2
monk2	169	6	0	2
monk3	122	6	0	2
mushroom	8124	22	0	2
sonar	208	0	60	2
soybean	683	35	0	19
vehicle	846	0	18	4
vote	435	16	0	2
vowel	990	3	10	11
wine	178	0	13	3

In order to calculate the classifiers' accuracy, the whole training set was divided into ten mutually exclusive and equal-sized subsets and for each subset the classifier was trained on the union of all of the other subsets. Then, cross validation was run 10 times for each algorithm and the median value of the 10-cross validations was calculated. It must be mentioned that we used the free available source code for most of the algorithms by [17] for our experiments.

The most usually used algorithm of each learning technique was selected as the representative algorithm for each technique. The C4.5 algorithm [11] was the representative of the decision trees in our study and the RIPPER [3] was the representative of the rule-based learners. The most well-known learning algorithm to estimate the values of the weights of a neural network - the Back Propagation (BP) algorithm [9] - was the representative of the perceptron-based algorithms. The SMO algorithm was the representative of the Support Vector Machines [10]. In our study, we also used the 3-NN algorithm as a representative of kNN [1]. The most usually used Naive Bayes algorithm [4] was the representative of the BNs in our study. Finally, we compare the selective voting using as base learners all the previous described algorithms with each algorithm alone.

In Table 2, we represent with "v" that the proposed ensemble looses from the specific algorithm. That is, the specific algorithm performed statistically better than the proposed according to t-test with $p<0.05$. Furthermore, "*" indicates that proposed ensemble performed statistically better than the specific classifier according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). In the last row of the table one can also see the aggregated results in the form (a/b/c). In this notation "a" means that the proposed ensemble is significantly less accurate than the compared algorithm in a out of 27 datasets, "c" means that the proposed algorithm is significantly more accurate than the compared algorithm in c out of 27 datasets, while in the remaining cases (b), there is no significant statistical difference.

TABLE 2. Comparing Local boosting DS with instance based classifiers and other ensembles that use DS as base learner

Datasets	Selective Voting	C4.5	3NN	NB	BP	SMO	RIPPER
autos	81.77	81.77	67.23 *	57.41 *	48.84 *	56.55 *	74.05 *
badges	100	100	100	99.66 *	100	100	100
breast-cancer	72.67	74.28	73.13	72.7	71.18	69.52	71.65
breast-w	96.71	95.01 *	96.61	96.07	95.97	96.75	95.72
credit-a	86.2	85.57	84.94	77.86 *	86.07	84.88	85.33
credit-g	75.3	71.25 *	72.21 *	75.16	72.75	75.15	71.86 *
diabetes	76.59	74.49	73.86	75.75	76.56	76.8	75.22
Glass	72.52	67.63	70.02	49.45 *	51.99 *	57.37 *	66.94 *
haberman	73.13	71.05	69.77	75.06	74.64	73.37	72.43
heart-c	84.45	76.94 *	81.82	83.34	81.39 *	83.89	79.05 *
heart-h	83.79	80.22	82.33	83.95	81.37	82.81	79.26 *
heart-statlog	84.19	78.15 *	79.11 *	83.59	82.11	83.89	78.7 *
hepatitis	84.65	79.22	80.9	83.81	81.3	85.7	77.43 *
ionosphere	90.69	89.74	86.02 *	82.17 *	85.84 *	88.07 *	89.3
Iris	95.6	94.73	95.2	95.53	96.27	96.2	93.93
Labor	93.4	78.6 *	87.83	93.57	88.57	92.97	83.37
lymphography	83.72	75.84	81.74	83.13	80.24 *	86.41	77.36
monk1	92.48	80.61 *	78.97 *	73.38 *	86.69 *	79.58 *	86.74 *
monk2	75.7	57.75 *	54.74 *	56.83 *	75.7	58.52 *	55.91 *
monk3	93.45	92.95	86.72 *	93.45	88.69	93.45	83.99 *
Mushroom	100	100	100	95.76 *	99.97	100	100
sonar	83.76	73.61 *	83.76	67.71 *	78.67 *	77.88 *	75.45 *
soybean	93.71	91.78 *	91.2 *	92.94	33.43 *	93.1	91.93 *
Vehicle	74.18	72.28 *	70.21 *	44.68 *	50.07 *	74.08	68.29 *
Vote	96.16	96.57	93.08 *	90.02 *	94.6	95.77	95.7
vowel	96.99	80.2 *	96.99 *	62.9 *	30.09 *	70.61 *	71.17 *
wine	98.47	93.2 *	95.85	97.46	83.08 *	98.76	92.48
W/D/L		*0/15/12*	*0/16/11*	*0/15/12*	*0/16/11*	*0/20/7*	*0/13/14*
Average accuracy	*86.68*	*81.98*	*82.75*	*79.38*	*76.89*	*82.67*	*81.23*

In the last raw of the Table 2 one can see the aggregated results. The presented ensemble is significantly more accurate than single C4.5 and NB in 12 out of the 27 datasets, while it has significantly higher error rate in none dataset. What is more, the proposed ensemble is significantly more accurate than RIPPER in 14 out of the 27 datasets, whilst it has significantly higher error rate in none dataset. Likewise, the proposed ensemble is significantly more accurate than BP and 3NN in 11 out of the 27 datasets, whilst it has significantly higher error rate in none dataset. In addition, the presented ensemble is significantly more accurate than SMO in 7 out of the 29 datasets, whilst it has significantly higher error rate in none dataset.

To sum up, the performance of the presented ensemble is more accurate than any base algorithm. The average relative accuracy improvement of the proposed methodology is from 5% to 13% in relation to simple algorithms.

Subsequently, we compare the proposed ensemble methodology with other well known ensembles using the same base learners:

- Voting [7]
- Grading [13]
- Stacking with MLR [16]
- Stacking with model trees [5]
- The methodology of selecting the best classifier of the according to 3-cross validation (BestCV) [12].
- StackingC [14]

In the last raw of the Table 3 one can see the aggregated results.

TABLE 3. Comparing Local boosting OneR with instance based classifiers and other ensembles that use OneR as base learner

Datasets	Selective voting	voting	grading	Stacking with MLR	Stacking with model trees	StackingC	BestCV
autos	81.77	81.62	79.08	50.33 *	77.2	82.27	79
badges	100	100	100	100	100	100	100
breast-cancer	72.67	71.13	72.73	72.17	72.56	72.63	71.65
breast-w	96.71	96.71	96.84	96.72	96.67	96.7	96.45
credit-a	86.2	86.14	85.75	85.8	85.41	85.78	84.77
credit-g	75.3	75.63	75.55	75.48	75.18	75.71	74.61
diabetes	76.59	77.17	76.78	76.95	75.98	76.89	76.8
Glass	72.52	71.98	71.56	50.79 *	69.25	71.13	68.21
haberman	73.13	72.58	73.57	72.78	72.09	73.04	73.43
heart-c	84.45	82.7	83.2	76.4	81.86	83.11	82.51
heart-h	83.79	83.48	83.48	66.03 *	81.81	84.2	83.04
heart-statlog	84.19	84.04	83.41	84.33	84.3	84.19	83.59
hepatitis	84.65	84.07	83.24	82.97	82.71	82.92	82.85
ionosphere	90.69	91.94	91.46	91.57	91.19	91.74	89.06
Iris	95.6	95.8	95.33	94.93	95.53	95.13	94.93
Labor	93.4	94.7	94.13	91.6	92.07	92.13	91.2
lymphography	83.72	84.56	85.15	82.03	82.03	82.97	84.24
monk1	92.48	86.33*	85.51*	86.31*	86.77*	86.47*	83.88*
monk2	75.7	60.7 *	66.1 *	79.67	74.29	79.75	73.41
monk3	93.45	93.28	93.45	93.45	92.87	93.45	93.45
Mushroom	100	100	100	100	100	100	100
sonar	83.76	81.66	82.9	85.48	84.1	85.97	82.23
soybean	93.71	93.65	94.17	91.5	93.34	93.51	92.84
Vehicle	74.18	74.43	75.17	75.84	74.97	76.05	73.85
Vote	96.16	95.77	95.86	96.06	95.31	96.11	95.84
vowel	96.99	92.92 *	93.66 *	93.12	96.33	97.18	96.99
wine	98.47	97.97	98.42	97.41	97.3	97.74	97.69
W/D/L		0/24/3	0/24/3	0/24/3	0/26/1	0/26/1	0/26/1
Average accuracy	86.68	85.59	85.80	83.32	85.60	86.55	85.43

The presented ensemble is significantly more accurate than simple voting, grading and stacking with MLR in 3 out of the 27 datasets, while it has significantly higher error rate in none dataset. What is more, the proposed ensemble is significantly more accurate than stacking with model trees, BestCV and StackingC in 1 out of the 27 datasets, respectively, whilst it has significantly higher error rate in none dataset.

To sum up, the performance of the presented ensemble is more accurate than the other well-known ensembles, using less time for training, too. The selective voting can achieve an increase in classification about 1.5% compared to simple voting. The average relative accuracy improvement of the proposed methodology is from 1% to 4% in relation to the remaining methods.

The machine learning community currently makes little use of the notion of selection; their aim is usually that of identifying an algorithm that creates effective ensembles, applying it, and presenting the results. However, it is quite likely that similar approaches to the proposed (based on testing and selection methodology) will outperform such methods.

A. Reducing computational cost

One generally wants (a) each classifier in a committee to be formed using as much data as possible, and (b) the size of the committee to be as large as possible. Practical considerations typically (a) limit the amount of data that can be used in training a single classifier, and (b) limit the useful size of a classifier committee. If the original dataset is too large to handle conveniently, then creating an ensemble will of course present even greater problems.

Reducing large datasets into more compact representative subsets while retaining essentially the same extractable knowledge could speed up learning and reduce storage requirements. Sampling is well accepted by the statistics community, who observe that "a powerful computationally intense procedure operating on a sub-sample of the data may in fact provide superior accuracy than a less sophisticated one using the entire data base" [6]. In practice, as the amount of data grows, the rate of increase in accuracy slows, forming the familiar learning curve. Whether sampling will be effective depends on how dramatically the rate of increase slows.

Interestingly, our results indicate that the proposed ensemble in smaller partitions can yield good results for learning curve. Given a training set of size n, we simple draw t random instances (a percentage of n instances) from the dataset without replacement and these t instances are then learned. In the last rows of the Table 4 one can see the aggregated results using different subsets of the training sets (20%, 33% and 50% of the instances of the whole sets) with the presented ensemble.

The proposed ensemble using the full training sets is significantly more accurate than the same ensemble using 20% of the training instances in 11 out of the 27 datasets. In addition, the presented ensemble using the full training set is significantly more accurate than the same ensemble using 33% of the training set in 8 out of the 27 datasets. Whereas, the presented ensemble using the full training set is significantly more accurate than the same ensemble using 50% of the training in only 3 out of the 27 datasets.

Even in the case of random partitioning, where any individual classifier created on a subset often performs significantly worse than a single classifier learned on the entire dataset, the selective voting on disjoint subsets can improve performance.

TABLE 4. Sampling

	Selective Voting (100%)	Selective Voting (20%)	Selective Voting (33%)	Selective Voting (50%)
autos	81.77	46.89 *	54.19 *	61.92 *
badges	100	100	100	100
breast-cancer	72.67	70.15	71.17	71.95
breast-w	96.71	96.04	96.35	96.82
credit-a	86.2	84.78	85.16	85.77
credit-g	75.3	71.78 *	73.24	74.47
diabetes	76.59	75.35	75.93	76.37
Glass	72.52	59.81 *	64.38 *	68.66
haberman	73.13	74.45	73.88	73.59
heart-c	84.45	81.43	83.15	83.45
heart-h	83.79	82.01	82.42	82.15
heart-statlog	84.19	81.33	81.7	83
hepatitis	84.65	80.36	81.8	82.4
ionosphere	90.69	86.87	88.69	89.15
Iris	95.6	93.27	94.6	94.67
Labor	93.4	78.4 *	85.23	87.97
lymphography	83.72	78.32	79.91	82.34
monk1	92.48	70.13 *	78.85 *	84.17
monk2	75.7	50.83 *	53.26 *	57.06 *
monk3	93.45	83.3 *	89.96	92.21
Mushroom	100	99.9	99.98	100
sonar	83.76	68.15 *	71.3 *	77.54
soybean	93.71	82.64 *	88.37 *	91.79
Vehicle	74.18	66.56 *	69.86 *	71.84
Vote	96.16	95.22	95.45	95.51
vowel	96.99	29.44 *	46.15 *	64.69 *
wine	98.47	93.43	96.13	97.46
W/D/L		0/16/11	0/19/8	0/24/3
Average accuracy	86.68	77.07	80.04	82.48

It is clear that search for the best re-sampling rate of the dataset needs to take into consideration not only the properties (the meta-characterizations) of the particular dataset but also our preferences with regard to some performance criteria such as time, accuracy and model size.

V. CONCLUSION

Combined classifiers can show better performance than the best single classifier used in isolation. This is because different classifier can potentially offer complementary information about the pattern and group decisions can take the advantage of the benefit of combining multiple classifiers in making final decision.

However, although classifier combination is widely applied in many fields, theoretical analysis of combination schemes can be very difficult. In many cases the performance of a combination method cannot be estimated theoretically and it can be evaluated on experimental basis in specific working conditions (a specific set of classifiers, training data and sessions, etc.). In this case the result depends on the specific conditions of the test and no information can be derived on the performance of the combination method if the working conditions change.

In the course of this paper, a number of different methods for ensemble construction have been outlined and considered. Some comparisons of the relative effectiveness can be found in the literature [2, 5] and whilst there is some progress here, there is still no general agreement about which methods are the most appropriate for which problems. Whilst there is still a lack of clarity about the best methods to adopt for particular applications, the "test and select" methodology advocated here provides a useful approach.

It should be noted that the time complexity of the selection of optimal subset of classifiers increases with respect to the number of base learners used. From this point of view the heuristic rule to test the algorithms in a small subset of the training set decreases the computational complexity. The algorithms that are initially used for building the ensemble are tested in a small subset of the training set and if they have statistically worse accuracy than the most accurate algorithm, they do not participate to the final decision of the ensemble.

Our experiment for several UCI datasets shows that the proposed combining method outperforms other combining methods we tried as well as any individual classifiers. Due to the encouraging results obtained from these experiments, we can expect that the proposed combining method can be successfully applied to the classification task in the real world case with more accuracy than the traditional data mining approaches.

Note that our approach is intended for combining classifiers that are heterogeneous (derived by different learning algorithms, using different model representations) and strong (i.e., each of the base-level classifiers performs relatively well in its own right), rather than homogeneous and weak such as bagging and boosting [2].

Another conclusion is that datasets too large to handle practically in the memory of the typical computer are appropriately handled by simple sampling to form a committee of classifiers. One advantage of this approach is that the partition size can simply be set at whatever amount of the original data can be conveniently handled on the available system.

In a future work, we will study the optimal re-sampling rate that may vary depending on a number of factors including the level of noise in the training set, the size of the training set, etc.

VI. REFERENCES

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

[2] E. Bauer and R.. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, 36(1/2):525–536, 1999.

[3] William W. & Cohen. Fast Effective Rule Induction. In Proc. ICML-95. 1995. 115-123.

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

[5] S. Dzeroski, B. Zenko. Is Combining Classifiers Better than Selecting the Best One. ICML 2002: 123-130, 2002.

[6] J.H. Friedman. Data mining and statistics: What's the connection? Proceedings of the 29th Symposium on the Interface Between Computer Science and Statistics, 1997.

[7] C. Ji & S. Ma. Combinations of weak classifiers. IEEE Transaction on Neural Networks, 8(1):32–42, 1997.

[8] L. Lam and C. Y. Suen. Application of majority voting to pattern recognition: An analysis of its behavior and performance. IEEE Trans. Pattern Analysis and Machine Intelligence,27(5):553–568,1997.

[9] Mitchell, T., Machine Learning, McGraw Hill, 1997.

[10] J. Platt. Using sparseness and analytic QP to speed training of support vector machines. In M. S. Kearns, S. A. Solla, & D. A. Cohn (Eds.), Advances in neural information processing systems 11. MA: MIT Press, 1999.

[11] J.R. Quinlan, C4.5: Programs for machine learning. Morgan Kaufmann, 1993.

[12] C. Schaffer. Selecting a classification method by cross-validation. Machine Learning 13, 135-143, 1993.

[13] K. Seewald and J. Furnkranz. An evaluation of grading classifiers. In Advances in Intelligent Data Analysis: Proceedings of the Fourth International Symposium (IDA-01), pages 221–232, Berlin, 2001. Springer.

[14] A.K. Seewald. How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness, in Sammut C., Hoffmann A. (eds.), Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002), Morgan Kaufmann Publishers, pp.554-561, 2002.

[15] A Sharkey, N. Sharkey, U. Gerecke, and G. Chandroth. The test and select approach to ensemble combination. In J. Kittler and F. Roli, editors, Multiple Classifier Systems. First International Workshop, MCS 2000, Cagliari, Italy, volume 1857 of Lecture Notes in Computer Science, pages 30–44. Springer-Verlag, 2000.

[16] K. Ting, & I. Witten. Issues in Stacked Generalization, Artificial Intelligence Research 10, 271-289, Morgan Kaufmann, 1999.

[17] I. Witten, and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, San Mateo, CA, 2000.

[18] D. Wolpert. "Stacked Generalization". Neural Networks, 5(2):241–260, 1992.