Ensemble Member Selection Using Multi-Objective Optimization

Tuve Löfström, Ulf Johansson and Henrik Boström

Abstract— **Both theory and a wealth of empirical studies have established that ensembles are more accurate than single predictive models. Unfortunately, the problem of how to maximize ensemble accuracy is, especially for classification, far from solved. In essence, the key problem is to find a suitable criterion, typically based on training or selection set performance, highly correlated with ensemble accuracy on novel data. Several studies have, however, shown that it is difficult to come up with a single measure, such as ensemble or base classifier selection set accuracy, or some measure based on diversity, that is a good general predictor for ensemble test accuracy. This paper presents a novel technique that for each learning task searches for the most effective combination of given atomic measures, by means of a genetic algorithm. Ensembles built from either neural networks or random forests were empirically evaluated on 30 UCI datasets. The experimental results show that when using the generated combined optimization criteria to rank candidate ensembles, a higher test set accuracy for the top ranked ensemble was achieved, compared to using ensemble accuracy on selection data alone. Furthermore, when creating ensembles from a pool of neural networks, the use of the generated combined criteria was shown to generally outperform the use of estimated ensemble accuracy as the single optimization criterion.**

I. Introduction

An *ensemble* is a composite model, aggregating multiple *base models* into one predictive model. An ensemble prediction, consequently, is a function of all included base models. The main motivation for using ensembles is the fact that combining several models using averaging will eliminate uncorrelated base classifier errors, see e.g., [1]. This reasoning, however, requires the base classifiers to commit their errors on different instances – clearly there is no point in combining identical models. Informally, the key term *diversity* is therefore used to denote the extent to which the base classifiers commit their errors on different instances.

The vital finding that ensemble error depends not only on the average accuracy of the base models, but also on their diversity was formally derived in [2]. From this, the overall goal when designing ensembles seems to be fairly simple, i.e., somehow combine models that are highly accurate but diverse. Base classifier accuracy and diversity are, however, highly correlated, so maximizing diversity will most likely reduce the average base classifier accuracy. Moreover, diversity is not uniquely defined for classification, further complicating the matter. As a matter of fact, numerous different diversity measures have been suggested, often in combination with quite technical and specialized ensemble creation algorithms. So, although there is strong consensus that ensemble models will outperform even the most accurate single models, there is no widely accepted solution to the problem of how to maximize ensemble accuracy.

When selecting base models for an ensemble, one would typically do this based on some performance measure obtained on training and/or selection data. Frequently used performance measures are of course base classifier selection set accuracy and ensemble selection set accuracy, but also numerous diversity measures. Several studies show that these selection criteria are, however, often not well correlated with ensemble accuracy on novel data, see e.g., [3][4][5]. Clearly, in order to be useful for maximizing ensemble accuracy, a selection criterion well correlated with ensemble test accuracy is required. Specifically, ranking ensembles using a suggested criterion on training or selection data should agree fairly well with ranking ensembles on test set accuracy.

With this in mind, the overall purpose of this paper is to investigate whether better performance can be obtained by combining several atomic performance measures. It should be noted that we do not strive for a single universal criterion, applicable to all learning tasks, but instead suggest a method for optimizing the optimization criterion for each dataset. More specifically, the suggested technique uses Multi-Objective Genetic Algorithms (MOGAs) [6] in order to find an optimal combination of the available performance measures.

II. Background and related work

When Krogh and Vedelsby in [2] derived the result that the ensemble error depends not only on the average error of the base models, but also on their diversity[1], this was very encouraging for the use of ensembles, since it in fact proved that an ensemble will always have higher accuracy than the average accuracy obtained by the base models. In the derived equation, the ensemble error, E, is expressed as the difference between the average error of the base models $\overline{E}$, and the ensemble diversity $\overline{A}$, measured as the weighted average of the squared differences in the predictions of the base models and the ensemble.

$$E = \overline{E} - \overline{A} \tag{1}$$

For regression using averaging, this is equivalent to:

T. Löfström is with the School of Business and Informatics, University of Borås, SE-501 90 Borås, Sweden. (phone: +46(0)33 – 4354236. Email: tuve.lofstrom@hb.se

U. Johansson is with the School of Business and Informatics, University of Borås, Sweden. Email: ulf.johansson@hb.se

H. Boström is with the School of Humanities and Informatics, University of Skövde, Sweden. Email: henrik.bostrom@his.se

[1] Krogh and Vedelsby used the term *ambiguity* instead of diversity. In this paper, the more common term diversity is, however, used.

$$E = (\hat{Y}_{ens} - Y)^2 = \frac{1}{M}\sum_i (\hat{Y}_i - Y)^2 - \frac{1}{M}\sum_i (\hat{Y}_i - \hat{Y}_{ens})^2 \quad (2)$$

where the first term is the (possibly weighted) average of the individual classifiers and the second is the amount of variability among ensemble members, i.e., the diversity term. Since diversity is always positive, this decomposition proves that the ensemble error E, will always be higher than $\bar{E}$, i.e., the average error of the individual models. As mentioned above, the two terms are, however, normally highly correlated, making it necessary to balance them instead of just maximizing diversity.

Furthermore, for classification, where a zero-one loss function is used, it is, not even possible to decompose the ensemble error into error rates of the individual classifiers and a diversity term. Instead, algorithms for ensemble creation typically use heuristic expressions trying to approximate the unknown diversity term. Naturally, the goal is again to find a diversity measure highly correlated with majority vote accuracy.

For a survey of approaches where diversity in some way has been utilized to build ensembles, see [7]. One typical example is when Giacinto and Roli, in [8], form a pair-wise diversity matrix, using the *double fault* measure and the *Q statistic* to select classifiers that are as diverse as possible. For the actual ensemble creation, they searched through the set of pairs of classifiers, adding one base classifier at a time, until the desired number of ensemble members is reached. In a similar fashion, Margineantu and Dietterich, in [9], also search out the pairs of classifiers with lowest *kappa-value* (highest diversity) from a set of classifiers produced by AdaBoost. Giacinto and Roli in [10], also applied a hierarchical clustering approach were ensembles are clustered based on pair-wise diversity. In the suggested approach, the final ensemble was formed by picking a classifier from each cluster, and step-wise joining the two least diverse classifiers until all classifiers belong to the same cluster. The ensemble used in the end is the ensemble with highest accuracy on a selection set. Banfield et al., finally, in [11], considered only the uncertain data points, in order to exclude base classifiers failing on a larger proportion of these instances. It should be noted that all these approaches select ensembles based on diversity between pairs of classifiers, rather than on ensemble diversity.

When Kuncheva and Whitaker studied ten statistics measuring diversity among binary classifier outputs, i.e., correct or incorrect vote for the class label, all diversity measures evaluated showed low or very low correlation with test set accuracy; see [3]. These results were also confirmed in another study, where an empirical evaluation was performed using the same ten diversity measures together with artificial neural network (ANN) ensembles on 11 UCI datasets; see [4]. In [5], it was reported that although techniques like resampling and varied architectures do produce more diverse base classifiers, this does not lead to more accurate ensembles. As a matter of fact, in that study, only ensemble training accuracy and base classifier training accuracy showed positive correlations with ensemble test accuracy. For diversity measures evaluated, the opposite was true, i.e., ensembles with low diversity were generally more accurate. In [12], it was investigated whether it is possible to somehow use results on training or selection data to estimate ensemble performance on novel data. With the specific setup evaluated, i.e., using ensembles built from a pool of independently trained ANNs, all with the same number of base classifiers, and targeting diversity only implicitly, the answer was a resounding no. Despite the fact that the measures evaluated included all the most frequently used, i.e., ensemble training and selection set accuracy, base classifier training and selection set accuracy, ensemble training and selection set AUC and diversity measured as *double fault* or *difficulty*, the results clearly showed that there was in general nothing to gain, in performance on novel data, by choosing an ensemble based on any of these measures. The empirical study showed not only that correlations between available training or selection measures and test set performance are very low, but also that there is no indication that ensembles with better performance on training or selection data would keep this edge on test data.

III. Method

The purpose of this study was to evaluate combinations of measures that could be used to estimate ensemble accuracy on independent test data. More specifically, we investigated using altogether four measures, either separately or somehow combined.

The four measures were: *ensemble accuracy* (EA), *base classifier accuracy* (BA), and the diversity measures *Kohavi-Wolpert* (KW) and *difficulty* (DI). It should be noted that EA is the accuracy obtained by the ensemble, while BA refers to the average accuracy obtained by the base classifiers, on a specific dataset.

Let $l(z_j)$ denote the number of classifiers correctly recognizing the instance z_j, L be the number of base classifiers and N the number of instances. Then, the *Kohavi-Wolpert variance* [13], is defined as:

$$KW = \frac{1}{NL^2}\sum_{j=1}^{N} l(z_j)(L - l(z_j)) \quad (3)$$

The *difficulty measure* was introduced in [14] by Hansen and Salomon. Let X be a random variable taking values in $\{0/L, 1/L, \ldots, L/L\}$. X is defined as the proportion of classifiers correctly classifying an instance x drawn randomly from the dataset. To estimate X, all L classifiers are evaluated on the dataset. The difficulty θ is then defined as the variance of X.

In this study, only the above two diversity measures are

included since they have been demonstrated to perform better than other measures, see e.g., [4]. In principle, however, any diversity measure could be included.

A. Ensemble Settings

Two types of base classifier were considered in this study: ANNs and decision trees (DTs). 45 base classifiers of each type were initially generated for each training set.

For ANN ensembles, some basic implicit diversity is introduced just by randomizing the starting weights. Several methods strive for increasing diversity further by splitting available data in order to train each base model (here ANN), on a slightly different training set. Such resampling techniques can divide the available data either by *features* or by *instances*. For ANN ensembles, it is also possible to use ANNs with different architectures in the ensemble. If the base classifiers are standard, fully-connected, feed-forward ANNs, the architectures can be varied by using different number of hidden layers, and of course, different number of units in each layer. According to Brown et al., random initialization of weights is generally ineffective for producing diverse ANNs; see [15]. The reason is that ANNs often converge to the same, or very similar optima, in spite of starting in different parts of the space. In addition, Brown et al. also state that manipulation of ANN architectures most often turns out to accomplish little. Regarding resampling, finally, Brown et al. say that the view is that it is more effective to divide training data by feature than by instance.

In this study, three sets of ANNs, each consisting of 15 ANNs, were generated. In the first set, the ANNs did not have a hidden layer, thus resulting in weaker models. The ANNs in the second set had one hidden layer, where the number of units, h, was based on dataset characteristics, but also slightly randomized for each ANN; see (4).

$$h = \left\lfloor 2 \cdot rand \cdot \sqrt{(v \cdot c)} \right\rfloor \tag{4}$$

Here, v is the number of input variables and c is the number of classes. *rand* is a random number in the interval [0, 1]. This set represents a standard setup for ANN training.

In the third set, each ANN had two hidden layers, where h_1 in (5) determined the number of units in the first hidden layer, and h_2 in (6) determined the number of units in the second layer. Again, v denotes the number of input variables and c the number of classes. Naturally, the purpose of using varied architectures was to produce a fairly diverse set of ANN base classifiers.

$$h_1 = \left\lfloor \sqrt{(v \cdot c)} / 2 + 4 \cdot rand \cdot \sqrt{(v \cdot c)} / c \right\rfloor \tag{5}$$

$$h_2 = \left\lfloor rand \cdot (\sqrt{(v \cdot c)} / c) + c \right\rfloor \tag{6}$$

In the experiments, 4-fold cross-validation was employed. For each fold, two thirds of the available training data was used for generating the base classifiers and one third was used for selection. All ANNs were trained without early stopping validation, leading to slightly over-fitted models. In order to introduce some further implicit diversity, each ANN used only 80 % of the available features, drawn randomly. Majority voting was used to determine ensemble classifications.

Targeting diversity is inherent in *random forest* models [16], even if no diversity measure is explicitly maximized. A single tree in a random forest is very similar to a standard decision tree like C4.5 [17] or CART [18]. The basic idea is, however, to directly create an accurate decision tree ensemble, by introducing randomness in both the instance selection and in the feature selection. The random forests considered in this study consisted of 45 unpruned[2] trees, where each tree was generated from a bootstrap replicate of the training set [19], and at each node in the tree generation, only a random subset of the available features was considered for partitioning the examples. The size of the subset was in this study set to the square root of the number of available features, as suggested in [16]. The set of instances used for estimating class probabilities, i.e., the estimation examples, consisted of the entire set of training instances. In the actual experimentation, the random forests were trained using the Rule Discovery System [20].

B. Experiments

The empirical study was divided into two experiments. The purpose of the first experiment was to search for combinations of measures, able to outperform ensemble accuracy as selection criteria. Naturally, combining accuracy measures with diversity measures fits very well into the original Krogh-Vedelsby idea, i.e., that ensembles should consist of accurate models that disagree in their predictions. The purpose of the second experiment was to evaluate the found complex performance measures. More specifically, these complex criteria were used to guide another GA search, but now among all possible ensembles, consisting of base classifiers from the pre-trained pools. So, in summary, in the first experiment, promising ensembles are selected from a fixed number of available ensembles, while in the second experiment, available base classifiers are freely combined into ensembles.

In the first experiment, 5000 random ensembles, where the number of base classifiers was normally distributed between 2 and 45, were used. The base classifiers were drawn at random from the 45 pre-trained models without replacement. For the actual search, MOGA was used to optimize the combination of measures. More specifically, the MOGA function '*gamultobj*' in Matlab was used [21].

Each individual in the MOGA population was represented as a vector, consisting of four float values. Each value corresponds to a weight for a specific performance measure (i.e., EA, BA, KW and DI). In Experiment 1, the

[2] Pruning has been observed to have a detrimental effect on forests of classification trees [17].

complex optimization criterion used for selection was the weighted sum of the four measures. For the MOGA, the following two objectives (fitness functions) were used:

- Maximizing correlation between the complex optimization criterion, and the ensemble accuracy on the selection set, measured on the 5000 ensembles.
- Maximizing average ensemble accuracy on the selection set for the top 5% ensembles, when all 5000 were ranked using the complex optimization criterion.

The purpose of the first objective was to achieve a solution that could rank the 5000 ensembles as well as possible, while the purpose of the second objective was to make the solution able to pinpoint the most accurate ensembles.

When using MOGA, the result is a set of solutions, residing in the Pareto-front. If $A = \{a_1,\ldots,a_m\}$ is a set of alternatives (here complex optimization criteria) characterized by a set of objectives $C = \{C_1,\ldots,C_M\}$ (here correlation and top 5% average ensemble accuracy), then the Pareto-optimal set $A^* \subseteq A$ contains all non-dominated alternatives. An alternative a_i is non-dominated iff there is no other alternative $a_j \in A, j \neq i$, where a_j is better than a_i on all criteria.

In the experiments, three specific solutions generated by the MOGA (and consequently residing in the Pareto front) were evaluated. More specifically, first the two individual solutions with best performance on each single objective were selected. These two solutions correspond to the two edges of the Pareto front, and are very similar to solutions achieved by optimizing each objective separately. The only difference to single objective search is that if there are ties, the solution with best performance on the second objective is guaranteed to be selected, which would not be the case in single objective search. The third solution selected was the one closest to the median of all the solutions in the Pareto front along both objectives. The Euclidian distance was used to find the ensemble closest to the median. This solution was selected, with the obvious motivation that it represents one of the solutions with highest correlation with selection set accuracy and at the same time one of the solutions leading to high ensemble accuracy among the top ranked ensembles.

For the actual evaluation of the solutions (complex optimization criteria) found, the test set accuracy of the highest ranked ensemble, from among the 5000, is reported. Naturally, when ranking the ensembles, the weighted sum of measures, as represented by the MOGA solution, is used. If there are several top ranked solutions, the average test accuracy of all ensembles selected, are reported.

The default settings for Matlab's GA toolbox were used when running the MOGA, except for the settings described in Table I.

TABLE I
GA SETTINGS EXPERIMENT 1

Parameter	Value
Population Size	200
Generations	100

In Experiment 2, GA was again used, now for searching for ensembles maximizing a complex optimization criterion. Here, the three sets of weights found in the first experiment were used, i.e., the search was standard GA looking for an ensemble maximizing the specific complex optimization criterion used. As comparison, results from employing GA to optimize EA only are included. In this experiment, no restriction on the size of the ensembles was enforced, except that an ensemble should consist of at least two base classifiers. The second experiment used the same two sets of base classifiers from which ensembles were constructed in the first experiment.

In Experiment 2, each individual in the GA population is a bitstring of length 45, where a '1' in any location indicates that the base classifier with the corresponding index should be included in the ensemble. For experimentation, the default settings for Matlab's GA toolbox were used, except for the settings described in Table II.

TABLE II
GA SETTINGS EXPERIMENT 2

Parameter	Value
Population type	Bitstring
Tolerance Function	30
Population Size	100
Generations	500

C. Datasets

For this study, 30 datasets from the UCI Repository [22] was used. For a summary of the characteristics of the data sets, see Table III. *No*. is a numbering used in the result tables instead of abbreviations. *Inst.* is the total number of instances in the data set. *Cl.* is the number of output classes in the data set. *Var*. is the number of input variables.

TABLE III
CHARACTERISTICS OF DATA SETS USED

Data set	No.	Inst.	Var.	Cl.	Data set	No.	Inst.	Var.	Cl.
bcancer	1	286	9	2	iono	16	351	34	2
breast (wbc)	2	699	9	2	labor	17	57	16	2
bupa	3	345	6	2	led7	18	3200	7	10
cleve	4	303	13	2	pima (diabetes)	19	768	8	2
cmc	5	1473	9	3	sick	20	2800	29	2
crabs	6	200	6	2	sonar	21	208	60	2
crx	7	690	15	2	soybean	22	683	35	19
ecoli	8	336	8	8	spambase	23	4601	57	2
german	9	1000	20	2	tae	24	151	5	3
glass	10	214	9	6	tictactoe	25	958	9	2
heart	11	270	13	2	waveform	26	5000	21	3
hepatitis	12	155	19	2	vehicle	27	846	18	4
horse	13	368	22	2	wine	28	178	13	3
hypo (thyroid)	14	3163	25	2	votes	29	435	16	2
image	15	2310	19	7	zoo	30	101	16	7

IV. RESULTS

The results from the first experiment are presented in Table IV. The tabulated values are 4-fold test set accuracies *for the*

highest ranked ensemble using the different atomic measures or combinations of measures. In the first four columns, the results when using ANNs as base classifiers are tabulated, while the results using random forests are shown in the last four columns. The column Ensemble Accuracy (EA) tabulates the results when using only ensemble accuracy as selection criteria. The columns Med, Corr and Top show results for using the complex selection criteria, found by the MOGA. The row Mean shows the average accuracy over all datasets. The row Rank, finally, contains the average ranks among the results from each set of base classifiers.

TABLE IV
ACCURACIES FOR EXPERIMENT 1

	ANN base classifiers				DT base classifiers			
No	**EA**	**Med**	**Corr**	**Top**	**EA**	**Med**	**Corr**	**Top**
1	.683	.718	.718	.722	.715	.722	.722	.711
2	.965	.964	.964	.966	.962	.968	.968	.967
3	.708	.698	.680	.706	.707	.701	.695	.718
4	.800	.820	.777	.790	.798	.807	.810	.810
5	.533	.546	.545	.537	.518	.518	.520	.511
6	.871	.910	.875	.915	.712	.690	.690	.735
7	.841	.852	.852	.853	.870	.866	.860	.874
8	.823	.827	.827	.821	.844	.827	.827	.833
9	.724	.770	.765	.756	.734	.748	.744	.748
10	.694	.726	.731	.698	.710	.703	.708	.708
11	.818	.799	.806	.806	.817	.832	.825	.817
12	.813	.817	.804	.810	.819	.849	.836	.789
13	.814	.804	.823	.810	.816	.823	.826	.818
14	.981	.981	.981	.980	.987	.989	.989	.988
15	.943	.945	.945	.941	.940	.937	.937	.936
16	.892	.889	.880	.886	.930	.937	.937	.934
17	.884	.873	.886	.868	.919	.911	.893	.929
18	.734	.737	.736	.712	.735	.737	.736	.738
19	.734	.762	.758	.766	.749	.758	.763	.764
20	.969	.968	.971	.968	.980	.981	.981	.980
21	.743	.736	.744	.740	.775	.760	.760	.760
22	.914	.922	.922	.918	.902	.903	.906	.903
23	.915	.922	.922	.922	.908	.922	.922	.924
24	.495	.453	.453	.473	.525	.493	.493	.527
25	.876	.888	.888	.851	.874	.889	.889	.896
26	.840	.868	.868	.867	.841	.842	.844	.844
27	.822	.831	.814	.829	.746	.750	.749	.751
28	.966	.971	.967	.965	.965	.977	.977	.977
29	.954	.954	.946	.956	.957	.961	.963	.944
30	.934	.930	.930	.930	.922	.920	.920	.910
Mean	**.823**	**.829**	**.826**	**.825**	**.823**	**.824**	**.823**	**.825**
Rank	**2.72**	**2.18**	**2.40**	**2.70**	**2.98**	**2.38**	**2.32**	**2.32**

The statistical test suitable for comparisons between several different algorithms over many datasets is a Friedman test. However, since we are primarily interested in comparing the results obtained when using any of the complex optimization criteria with the results achieved when using only ensemble accuracy, the tabulated wins, draws and losses can still provide valuable insights on each of these three results compared to the ensemble accuracy and will be presented below.

The Friedman test did not detect any significant differences for either ANNs or random forests. For random forests, however, all the complex optimization criteria were clearly better than ensemble accuracy. Considering ANNs, at least the median solution were clearly better than ensemble accuracy.

The standard, one-tail sign test is suitable to use when comparing two algorithms on many datasets. When using a sign test to compare two algorithms on 30 datasets, 20 wins or more are required for statistical significance at $\alpha = 0.05$, while 19 wins has a p-value of 0.1002. Table V shows the number of wins, draws and losses for the three different complex optimization criteria against ensemble accuracy. Results that are significant when comparing two algorithms at $\alpha = 0.05$ are bold and underlined, while results only underlined are significant at $\alpha = 0.10$.

TABLE V
WINS/DRAWS/LOSSES FOR EXPERIMENT 1

W/D/L	ANN	DT
Med	18/2/10	19/1/10
Corr	19/1/10	**20/0/10**
Top	13/0/17	**19/2/9**

As seen in Table V, the use of a complex optimization criterion for selecting ensembles was quite successful. As a matter of fact, in five of six comparisons, it was clearly better to use an ensemble selected based on the complex criterion, compared to an ensemble selected based on selection set accuracy only.

Table VI below shows the results from Experiment 2.

TABLE VI
ACCURACIES FOR EXPERIMENT 2

	ANN base classifiers				DT base classifiers			
No	**EA**	**Med**	**Corr**	**Top**	**EA**	**Med**	**Corr**	**Top**
1	.690	.722	.722	.739	.725	.718	.708	.715
2	.968	.966	.967	.970	.966	.973	.970	.968
3	.718	.718	.724	.712	.721	.709	.692	.706
4	.792	.817	.823	.820	.797	.797	.797	.797
5	.757	.860	.915	.775	.780	.730	.705	.715
6	.842	.846	.849	.849	.862	.862	.862	.865
7	.754	.757	.767	.764	.745	.747	.743	.745
8	.805	.832	.843	.840	.810	.813	.802	.806
9	.839	.842	.842	.829	.842	.836	.822	.836
10	.813	.815	.821	.817	.823	.837	.829	.829
11	.980	.983	.980	.983	.989	.990	.990	.990
12	.900	.897	.886	.897	.928	.925	.931	.931
13	.881	.884	.893	.893	.839	.893	.893	.929
14	.745	.760	.762	.760	.741	.758	.754	.753
15	.969	.974	.971	.972	.983	.981	.982	.981
16	.745	.727	.738	.732	.755	.779	.779	.784
17	.892	.863	.864	.849	.884	.883	.877	.882
18	.955	.954	.951	.958	.949	.954	.949	.965
19	.689	.712	.726	.712	.736	.731	.731	.708
20	.500	.486	.500	.507	.520	.520	.534	.514
21	.833	.826	.822	.819	.755	.746	.746	.746
22	.964	.966	.977	.972	.966	.977	.977	.983
23	.930	.930	.940	.930	.910	.910	.900	.910
24	.818	.842	.842	.836	.863	.869	.857	.863
25	.921	.922	.918	.914	.901	.910	.904	.910
26	.554	.553	.552	.558	.534	.535	.535	.542
27	.948	.947	.946	.946	.945	.944	.939	.944
28	.736	.737	.736	.738	.737	.738	.736	.738
29	.861	.867	.865	.868	.844	.843	.844	.843
30	.925	.929	.925	.927	.920	.922	.925	.924
Mean	**.824**	**.831**	**.836**	**.829**	**.826**	**.828**	**.824**	**.827**
Rank	**2.95**	**2.53**	**2.20**	**2.32**	**2.43**	**2.27**	**2.93**	**2.37**

When the results were evaluated using a Friedman test, no significant differences could be detected between the complex optimization criteria and ensemble accuracy for ANNs, even though both the correlated and the top solution

were clearly better than ensemble accuracy. No significant differences could be detected for random forests either, but here, the median solution was clearly better than the correlated solution.

Table VII below shows the number of wins, draws and losses for the three different complex optimization criteria, against the GA using only EA as objective.

TABLE VII
WINS/DRAWS/LOSSES FOR EXPERIMENT 2

W/D/L	ANN	DT
Med	19/0/11	15/1/14
Corr	18/1/11	12/0/18
Top	**21/0/9**	14/3/13

The results in Table VII are a bit mixed. When using ANNs as base classifiers, it is always clearly better to use a complex optimization criterion. For random forests, however, there is very little difference between using the complex criterion and just ensemble accuracy.

Table VIII below, finally, compares the results obtained in the two different experiments.

TABLE VIII
COMPARISON EXPERIMENT 1 AND EXPERIMENT 2

W/D/L	ANN Exp 1			
Exp 2	**EA**	**Med**	**Corr**	**Top**
EA	**20/0/10**	17/0/13	14/0/16	19/0/11
Med	**22/0/8**	16/0/14	19/0/11	**20/0/10**
Corr	**25/0/5**	19/0/11	**21/0/9**	**21/0/9**
Top	**24/0/6**	19/1/10	**21/0/9**	**22/0/8**
	DT Exp 1			
EA	19/0/11	13/0/17	15/0/15	13/0/17
Med	**22/0/8**	15/0/15	16/0/14	14/0/16
Corr	**20/0/10**	15/0/15	14/0/16	15/0/15
Top	**22/0/8**	19/0/11	19/0/11	14/0/16

As seen in Table VIII, it is obvious that the unlimited search among all possible ensembles, using the 45 base classifiers is, at least for ANNs, generally a stronger approach than selecting from a smaller subset (here 5000) of ensembles. This is despite the fact that these 5000 ensemble were actually the ones used for finding the optimization criteria. It is also worth noting that for all fitness functions, the use of a complex optimization criterion to search for the best ensemble (in Experiment 2) was better than what was achieved using ensemble accuracy in Experiment 1.

V. CONCLUSIONS

We have in this paper suggested a novel technique for constructing nsembles, where a genetic algorithm is used for combining several atomic measures into a complex optimization criterion. It should be noted that this optimization criterion is itself optimized for each dataset. The selected combined criterion can then be used for evaluating a set of candidate ensembles or for searching for a set of ensemble base classifiers. When selecting a specific ensemble from a number of existing ensembles, experimental results presented in this work, clearly show that it is generally beneficial to use an optimized combined criterion, instead of ensemble accuracy as the single criterion. When using the complex criterion as fitness function in a GA search aimed at selecting base classifiers from a pool to form an ensemble, the results are not as clear-cut. For ANNs base classifiers, the experimental results give strong evidence for that it is better to use the combined optimization criterion, but for random forests, there is only marginal difference between using the combined criterion and ensemble accuracy alone.

VI. DISCUSSION

The approach presented here show promising results for both ensembles of ANNs and DTs. The experimental results demonstrate that it is clearly better to use the novel approach when selecting from a set of existing ensembles. They also show that a complex optimization criterion is better for an unlimited search among all possible ensembles when using ANNs as base classifiers. For DTs, however, the novel approach does not give any competitive advantage over using just ensemble selection set accuracy as optimization criterion, as seen in Experiment 2. The reason for this difference in applicability to different types of base classifiers, as well as future enhancements of the overall approach, is the focus of future work.

ACKNOWLEDGMENT

This work was supported by the Information Fusion Research Program (University of Skövde, Sweden) in partnership with the Swedish Knowledge Foundation under grant 2003/0104 (URL: http://www.infofusion.se).

REFERENCES

[1] T. G. Dietterich, Machine learning research: four current directions, *The AI Magazine*, 18: 97-136, 1997.

[2] A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems*, *Volume 2*, pp. 650-659, San Mateo, CA, Morgan Kaufmann, 1995.

[3] L. I. Kuncheva and C. J. Whitaker, Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, *Machine Learning*, (51):181-207, 2003.

[4] U. Johansson, T. Löfström and L. Niklasson, The Importance of Diversity in Neural Network Ensembles - An Empirical Investigation, *The International Joint Conference on Neural Networks*, IEEE Press, Orlando, FL, pp. 661-666, 2007.

[5] U. Johansson, T. Löfström and H. Boström, The Problem with Ranking Ensembles Based on Training or Validation Performance, *The International Joint Conference on Neural Networks*, IEEE Press, Hong Kong, China, pp. 3221-3227, 2008.

[6] C. M. Fonseca and P. J. Fleming, An Overview of Evolutionary Algorithms in Multiobjective Optimization, *Evolutionary computation*, Vol. 3, No. 1: 1-16, 1995.

[7] L.I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.

[8] G. Giacinto and F. Roli, Design of Effective Neural Network Ensembles for Image Classification Purposes, *Image and Vision Computing*, vol. 19: 699-707, 2001.

[9] D. D. Margineantu and T. G. Dietterich, Pruning Adaptive Boosting, *14th International Conference on Machine Learning*, Morgan Kaufmann Publishers, Nashville, TN, pp. 211-218, 1997.

[10] G. Giacinto and F. Roli, An Approach to the Automatic Design of Multiple Classifier Systems, *Pattern Recognition Letters*, vol. 22: 25-33, 2001.

[11] R. E. Banfield, L. O. Hall, K. W. Bowyer and W. P. Kegelmeyer, A New Ensemble Diversity Measure Applied to Thinning Ensembles, *International Workshop on Multiple Classifier Systems*, Surrey, UK, pp. 306 - 316, 2003.

[12] U. Johansson, T. Löfström and L. Niklasson,, Evaluating Standard Techniques for Implicit Diversity, *Advances in Knowledge Discovery and Data Mining - 12th Pacific-Asia Conference, PAKDD 2008*, Springer Verlag, LNAI 5012: 613-622, 2008.

[13] R. Kohavi, and D. H. Wolpert, Bias plus variance decomposition for zero-one loss functions. *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996

[14] L. Hansen and P. Salomon, Neural Network Ensembles, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12:933-1001, 1990,.

[15] G. Brown, J. Wyatt, R. Harris and X. Yao, Diversity Creation Methods: A survey and Categorisation, *Journal of Information Fusion*, 6(1):5-20, 2005.

[16] L. Breiman, Random forests, *Machine Learning*, 45(1):5-32, 2001.

[17] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.

[18] L. Breiman, J. H. Friedman, R. A. Olshen and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, 1984.

[19] L. Breiman, Bagging predictors, *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996

[20] Compumine, Rule Discovery System (RDS) 2.6; http://www.compumine.se

[21] The Mathworks, Genetic Algorithm and Direct Search Toolbox - MATLAB; http://www.mathworks.com/ products/gads/.

[22] A. Asuncion and D.J. Newman, UCI machine learning repository, 2007