Evaluating Ensembles on QSAR Classification

Ulf Johansson[1], Tuve Löfström[1] and Ulf Norinder[2]

***Abstract*—Novel, often quite technical algorithms, for ensembling artificial neural networks are constantly suggested. Naturally, when presenting a novel algorithm, the authors, at least implicitly, claim that their algorithm, in some aspect, represents the state-of-the-art. Obviously, the most important criterion is predictive performance, normally measured using either accuracy or area under the ROC-curve (AUC). This paper presents a study where the predictive performance of two widely acknowledged ensemble techniques; GASEN and NegBagg, is compared to more straightforward alternatives like bagging. The somewhat surprising result of the experimentation using, in total, 32 publicly available data sets from the medical domain, was that both GASEN and NegBagg were clearly outperformed by several of the straightforward techniques. One particularly striking result was that *not* applying the GASEN technique; i.e., ensembling all available networks instead of using the subset suggested by GASEN, turned out to produce more accurate ensembles.**

***Index Terms*— Classification, Data Mining, Ensembles, QSAR.**

I. Introduction

THIS paper, which presents work in progress from the INFUSIS project, uses a number of publicly available data sets from the medical domain to evaluate a selection of popular classifiers, particularly several variants of neural network ensembles. The overall purpose is to compare the predictive performances of several well-known ensemble techniques, specifically investigating whether more sophisticated schemes really outperform straightforward choices, when applied to real-world data sets. In addition, the study also looks into the effects of using feature reduction, when performing predictive modeling in this specific domain.

II. Background

Ensemble learning refers to a large collection of methods that learn a target function by training a number of individual learners and combining their predictions. The ensemble prediction, when applied to a novel instance, is consequently a function of all included base models.

As expected, the overall purpose of using ensembles is to increase predictive performance, and there are at least two, intuitive, explanations to why ensemble learning should produce more accurate models.

First of all, there is no such thing as a universal "best" learning technique. Since different algorithms have different inductive biases, they are simply more or less suitable for specific problems. In addition, most learners are not general function approximators; i.e., they are unable to represent some target functions. Often, however, such target functions could be approximated by averaging several models.

Second: if several models are combined by averaging, uncorrelated errors of individual classifiers can be eliminated; see e.g. [1]. This, however, requires that the base classifiers commit their errors on different instances; i.e., ensemble *diversity* has an effect on ensemble accuracy. The problem of how to maximize ensemble accuracy is, unfortunately, far from solved, though. In particular, the relationship between ensemble diversity and accuracy is not completely understood, making it hard to efficiently utilize diversity for ensemble creation.

Krogh and Vedelsby in [2] derived the result that ensemble error depends not only on the average accuracy of the base models, but also on their diversity (ambiguity). More formally, the ensemble error, E, is:

$$E = \bar{E} - \bar{A} \qquad (1)$$

where $\bar{E}$ is the average error of the base models and $\bar{A}$ is the ensemble diversity, measured as the weighted average of the squared differences in the predictions of the base models and the ensemble. Since diversity is always positive, this decomposition proves that the ensemble will always have higher accuracy than the average accuracy obtained by the individual classifiers. The two terms are, however, normally highly correlated, making it necessary to balance them instead of just maximizing the diversity term.

Brown et al. in [3] introduced a taxonomy of methods for creating diversity. The main distinction made is between explicit methods, where some metric of diversity is directly optimized, and implicit methods, where the method is supposed to produce diversity without actually targeting it.

For artificial neural network (ANN) ensembles, which are the focus of this study, the most obvious method to introduce implicit diversity is to randomize the starting weights. Starting from randomized weights is, of course, a standard procedure for most ANN training. 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 number of hidden layers and the number of units in each layer can be varied.

Many methods strive for diversity by splitting the training

[1] CSL@BS, School of Business and Informatics, University of Borås, Sweden. Email: {ulf.johansson, tuve.lofstrom}@hb.se
[2] AstraZeneca R&D, Södertälje, Sweden. Email: ulf.norinder@astrazeneca.com

This work was supported by the INFUSIS project (www.his.se/infusis) at the University of Skövde, Sweden, in partnership with the Swedish Knowledge Foundation under grant 2008/0502.

data in order to train each base classifier using slightly different training sets. Here, we will use mainly standard *bagging* [4], a procedure where the available data is divided by instances.

More specifically, when using bagging to introduce implicit diversity in ANN ensembles, each ANN is trained using individual training sets. Every training set (called bootstrap) has the same size as the original training set, and is created by repeated sampling (using a uniform distribution) of training instances. Since sampling is done with replacement, some instances may appear several times, while others are omitted. On average, a bootstrap sample contains approximately 63% of the original training instances.

A. Related work

Several approaches create ensembles by somehow using genetic algorithms to search for optimal ensembles. Specifically, Zhou et al. have suggested an algorithm named GASEN; see e.g., [5] [6]. GASEN is really a post-processing technique, since several ANNs are trained before a genetic algorithm is used to select an optimal subset of the available networks. The optimization is performed on individual ANNs, and each ANN is coded (in the chromosome) as a real number indicating the goodness of including that ANN. The optimization criterion (the fitness) boils down to accuracy on a holdout (validation) set. The number of ANNs in the ensemble can vary, since all ANNs with strength values higher than a specific threshold (which is a pre-set parameter) are included in the ensemble.

NegBagg [7] is a recently proposed cooperative ensemble learning algorithm for designing ANN ensembles. The algorithm incrementally trains different individual ANNs in an ensemble using the negative correlation learning algorithm [8]. Bagging is used in NegBagg to create different training sets for different ANNs in the ensemble. The idea behind using negative correlation learning in conjunction with the bagging algorithm is to facilitate interaction and cooperation among ANNs during their training. NegBagg also uses a constructive approach to automatically determine the number of hidden neurons for ANNs.

III. METHOD

In this study, we wanted to compare (suggested) state-of-the-art techniques to more straightforward choices. Naturally, NegBagg (NB) and GASEN (GAS) represent the sophisticated algorithms, claimed to have superior performance, at least on benchmark problems. For the comparison, several, readily available techniques implemented in the Weka data mining workbench were used [9]. More specifically, we decided to evaluate the following five techniques in Weka:

- Multilayer perceptron neural network (MLP)
- Radial basis function neural network (RBF)
- Support vector machine, trained using Platt's sequential minimal optimization algorithm (SVM)
- 15 bagged MLPs (Bag-M_W)
- 15 bagged RBF networks (Bag-RBF)

For simplicity, all parameter settings in Weka were left at the default values.

In addition, we included two variants of MLP ensembles, where additional implicit diversity is introduced by varying the architectures of the base classifiers. As a matter of fact, in a previous study [10], we found varying the architectures to be more successful than using bootstraps. The two techniques included in the study were:

- 15 bagged MLPs, all trained on bootstraps and having slightly different architectures (Bag-M_B)
- 15 averaged MLPs, each trained on all available data and having slightly different architectures (Avg-M_A)

For Bag-M_B and Avg-M_A, eight of the 15 ANNs had one hidden layer, and the other seven had two hidden layers. The exact number of units in each hidden layer was slightly randomized, but was based on the number of inputs and classes in the current data set. For ANNs with one hidden layer, the number of hidden units is determined from (2) below.

$$h = \lfloor 2 \cdot rand \cdot \sqrt{v \cdot c} \rfloor \quad (2)$$

where v is the number of input variables and c is the number of classes. rand is a random number in the interval [0, 1]. For ANNs with two hidden layers, the number of units in the first hidden layer is h_1 determined from (3) below and the second hidden layer is h_2 from (4) below. v is again the number of input variables and c is the number of classes.

$$h_1 = \left\lfloor \frac{\sqrt{v \cdot c}}{2} + 4 \cdot rand \cdot \frac{\sqrt{v \cdot c}}{c} \right\rfloor \quad (3)$$

$$h_2 = \left\lfloor c + rand \cdot \frac{\sqrt{v \cdot c}}{c} \right\rfloor \quad (4)$$

When training individual ANNs, 75% of the available data was used for the actual training and the remaining 25% was used as an early stopping validation set. Finally, it should be noted that, in the experimentation, GASEN started from the 15 ANNs produced by BAG-M_B, i.e., GASEN ensembles include a subset of the ANNs used by BAG-M_B.

A. Data sets

In this study, eight different Quantitative Structure-Activity Relationship (QSAR) data sets from the study reported by Bruce et al. in [11] are used. The same data sets were originally used by Sutherland et al. in [12]. The data sets are related to the following biological targets:

- A set of 114 angiotensin converting enzyme (ACE) inhibitors. Activities of these compounds spread over a wide range, with pIC50 values ranging from 2.1 to 9.9.
- A set of 111 acetylcholinesterase (AchE) inhibitors with pIC50 values ranging from 4.3 to 9.5.
- A set of 163 ligands for the benzodiazepine receptor (BZR) with activity (pIC50) values ranging from 5.5 to 8.9.
- A set of 322 cyclooxygenase-2 (COX2) inhibitors having pIC50 values that range from 4.0 to 9.0.

- A set of 397 dihydrofolate reductase inhibitors (DHFR) with pIC50 values for rat liver enzyme ranging from 3.3 to 9.8.
- A set of 66 inhibitors of glycogen phosphorylase b (GPB) with pKi values ranging from 1.3 to 6.8
- A set of 76 thermolysin inhibitors (THER) having pKi values ranging from 0.5 to 10.2
- A set of 88 thrombin inhibitors (THR) with pKi values ranging from 4.4 to 8.5

The continuous numerical values for activity (pIC50 for the first five data sets and pKi for the last three) in the study by Sutherland et al. were transformed by Bruce et al. into two categorical classes (active and inactive). Since each data set showed a uniform distribution of activity values, the transformation, used the median activity value as a threshold between the two classes to create a 50/50 split of active/inactive observations.

It should be noted that Bruce et al. used two separate feature sets, *2.5D descriptors* (2.5D) and *linear fragment descriptors* (Frags.) to characterize the chemical structures in the data sets. Following Bruce et al., we too evaluate both these feature sets here.

An initial investigation of the data sets showed that several input attributes were strongly correlated. Because of this, we decided to investigate whether feature reduction could be beneficial for the predictive performance. To keep things uncomplicated, the standard procedure for feature selection in Weka (called CfsSubSetEval) was used. Simply put, CfsSubSetEval, when using standard settings, performs a best first search, keeping only attributes relatively strongly correlated with the target variable. In addition, attributes strongly correlated with other attributes, already in the reduced feature set, are discarded. Here, the feature selection resulted in a significant reduction of features for all data sets. Table I below summarizes the resulting 32 data sets.

TABLE I
CHARACTERISTICS OF DATA SETS USED

Data set	Instances	Attributes			
		2.5D	2.5D_red	Frags	Frags_red
ACE	114	56	12	1024	11
AchE	111	63	9	774	16
BZR	163	75	16	832	20
COX2	322	74	12	660	24
DHFR	397	70	10	951	14
GPB	66	70	2	692	14
THER	76	64	10	575	21
THR	88	66	7	527	7

A. Experiments

This study contains two main experiments. In the first experiment, the data sets with 2.5D feature sets were used. In the second experiment, the Fragment feature sets were used.

In the experiments, both accuracy and area under the ROC curve (AUC) were used for evaluation. While accuracy is based only on the final classification, AUC measures the ability to rank instances according to how likely they are to belong to the positive class; see e.g., [13]. AUC can be interpreted as the probability of ranking a true positive instance ahead of a false positive; see [14]. For actual experimentation, standard 10-fold cross-validation was used. The reported accuracies are therefore averaged over the ten folds. Following the standard procedure for AUC evaluation, only one ROC curve based on all test set instances from each fold is produced per data set.

Some initial experiments showed that the unreduced fragment feature sets turned out to have too many attributes for Weka's MLPs; running even a single MLP on just one fold would take several hours. Because of this, MLP and BAG-M_W could not be evaluated on these feature sets.

IV. RESULTS

Table II below shows the accuracy results, using non-reduced 2.5D descriptor sets. Comparing overall averages and mean ranks, it is obvious that especially NegBagg exhibits remarkably poor performance. Although GASEN has the third best mean rank, it should be noted that it is still higher than Bag-M_B; i.e., GASEN post-processing was actually, more often than not, detrimental. Using this descriptor set, the highest average accuracy, and best mean rank, was obtained by bagging MLPs in Weka.

TABLE II
EXPERIMENT 1: 2.5D DESCRIPTORS, NO REDUCTION. ACCURACY

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	84.4	82.6	91.4	88.0	86.1	87.7	86.8	89.5	87.7
AchE	72.9	71.2	72.1	72.8	73.8	70.0	70.9	69.1	63.6
BZR	75.4	79.8	77.3	77.9	81.0	77.9	72.9	77.8	71.2
COX2	73.9	65.2	76.4	76.1	69.0	76.9	74.2	75.1	74.2
DHFR	80.4	70.8	79.9	83.6	73.1	81.3	82.3	80.8	81.1
GPB	77.1	70.0	68.3	74.3	71.4	76.9	74.1	81.7	77.4
THER	75.0	64.3	66.4	77.9	68.6	73.4	72.1	73.4	65.9
THR	73.9	75.1	66.0	72.6	75.0	69.9	74.7	70.1	66.8
Mean	**76.6**	**72.4**	**74.7**	**77.9**	**74.7**	**76.8**	**76.0**	**77.2**	**73.5**
Mean rank	**4.88**	**6.50**	**5.63**	**3.00**	**5.00**	**4.00**	**5.25**	**4.25**	**6.13**

The picture in Table III, which shows AUC results for the same descriptor set, is quite similar. Here, however, the difference between bagging MLPs in Weka, and the second best technique is quite large. Somewhat surprising, using single MLPs was actually quite a strong option.

TABLE III
EXPERIMENT 1: 2.5D DESCRIPTORS, NO REDUCTION. AUC

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	0.95	0.90	0.91	0.95	0.94	0.95	0.96	0.95	0.94
AchE	0.76	0.74	0.72	0.78	0.76	0.75	0.77	0.74	0.74
BZR	0.83	0.88	0.77	0.85	0.86	0.82	0.82	0.83	0.80
COX2	0.81	0.73	0.76	0.84	0.77	0.83	0.84	0.81	0.81
DHFR	0.89	0.76	0.80	0.91	0.81	0.91	0.90	0.90	0.89
GPB	0.80	0.74	0.69	0.81	0.77	0.80	0.81	0.82	0.77
THER	0.84	0.70	0.68	0.87	0.75	0.82	0.80	0.81	0.78
THR	0.83	0.76	0.67	0.85	0.81	0.80	0.82	0.74	0.65
Mean	**0.84**	**0.78**	**0.75**	**0.86**	**0.81**	**0.84**	**0.84**	**0.82**	**0.80**
Mean rank	**3.75**	**7.13**	**8.50**	**1.75**	**5.38**	**3.88**	**3.38**	**4.25**	**6.63**

When using reduced 2.5D feature sets, bagging MLPs in Weka suddenly obtained a very poor mean rank, comparing accuracies; see Table IV below. Bagging MLPs with different architectures and trained on bootstraps was here the best

choice. Again, especially NegBagg, but also GASEN, were clearly outperformed by the more straightforward techniques.

TABLE IV
EXPERIMENT 1: 2.5D DESCRIPTORS, FEATURE REDUCTION. ACCURACY

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	88.7	88.9	90.5	88.9	92.4	90.0	88.5	91.1	92.1
AchE	69.4	72.1	69.2	69.3	78.4	72.7	70.9	70.9	59.1
BZR	75.5	75.4	75.5	74.8	76.7	76.6	75.4	76.0	74.1
COX2	76.7	75.2	73.9	77.9	75.8	74.5	73.2	74.5	72.6
DHFR	82.6	74.3	78.6	85.4	78.3	82.9	84.1	81.8	78.0
GPB	77.4	78.8	76.0	74.5	77.4	76.9	75.5	76.9	71.2
THER	68.4	65.9	72.7	67.3	65.9	70.0	69.6	73.8	72.5
THR	71.5	71.7	73.8	70.3	72.8	74.9	77.2	71.5	72.4
Mean	**76.3**	**75.3**	**76.3**	**76.1**	**77.2**	**77.3**	**76.8**	**77.1**	**74.0**
Mean rank	**4.88**	**5.38**	**5.00**	**6.00**	**3.38**	**3.38**	**5.38**	**4.00**	**6.75**

Comparing AUCs in Table V below, neither GASEN nor NegBagg were among the better techniques. Although the differences in average AUC are small, the mean ranks still indicate that most straightforward ensemble techniques generally will have higher AUCs than the sophisticated techniques.

TABLE V
EXPERIMENT 1: 2.5D DESCRIPTORS, FEATURE REDUCTION. AUC

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	0.94	0.94	0.90	0.95	0.96	0.97	0.96	0.96	0.97
AchE	0.73	0.83	0.69	0.80	0.85	0.78	0.77	0.78	0.71
BZR	0.81	0.85	0.75	0.83	0.86	0.82	0.80	0.82	0.82
COX2	0.83	0.82	0.74	0.85	0.84	0.84	0.83	0.84	0.81
DHFR	0.90	0.85	0.79	0.92	0.88	0.91	0.91	0.90	0.86
GPB	0.82	0.82	0.77	0.80	0.81	0.76	0.77	0.77	0.77
THER	0.74	0.74	0.73	0.78	0.76	0.75	0.73	0.77	0.77
THR	0.72	0.79	0.74	0.78	0.79	0.82	0.82	0.78	0.79
Mean	**0.81**	**0.83**	**0.76**	**0.84**	**0.84**	**0.83**	**0.82**	**0.83**	**0.81**
Mean rank	**5.75**	**4.50**	**8.25**	**3.25**	**3.13**	**3.88**	**5.50**	**4.25**	**5.38**

The most important result of Experiment 1 is, of course, the fact that the simple techniques clearly outperformed NegBagg and GASEN, with respect to both accuracy and AUC. Naturally, the difference in performance between two techniques, on a single data set, is often very small. Still, the overall picture is, however, quite clear. Specifically, Bag-M_B obtained a lower mean rank than GASEN and NegBagg on all four comparisons; i.e., both accuracy and AUC using either all features or the reduced feature set.

Another interesting result was the fact that using this descriptor set, the feature reduction turned out to be, at least, moderately successful. More often than not, the reduction led to improved accuracy, while the picture was more mixed regarding AUCs. The numbers in Table VI below indicate wins, draws and losses, for using the reduced feature set, compared to the original.

TABLE VI
EXPERIMENT 1: 2.5D DESCRIPTORS, EFFECT OF FEATURE REDUCTION

Technique	Accuracy			AUC		
	win	draw	loss	win	draw	loss
MLP	5	0	3	3	0	5
RBF	6	0	2	7	0	1
SVM	3	0	5	3	0	5
Bag-M_W	4	0	4	3	1	4
Bag-RBF	5	0	3	6	1	1
Bag-M_B	4	1	3	4	0	4
Avg-M_A	5	1	2	1	0	7
Gas	5	0	3	4	0	4
NB	4	0	4	3	0	5
Summary	**41**	**2**	**29**	**34**	**2**	**36**

Turning to Experiment 2, i.e., using the Fragments descriptor set, Table VII below shows the accuracies obtained by the different techniques, when using all features. Again, GASEN and NegBagg have lower accuracies, compared to Avg-M_A and Bag-M_B. Looking at Table VIII, the same holds for AUCs as well.

TABLE VII
EXPERIMENT 2: FRAGMENT DESCRIPTORS, NO REDUCTION. ACCURACY

Set	RBF	SVM	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	76.5	78.4	79.1	80.7	81.6	78.1	80.8
AchE	66.6	68.5	71.1	71.8	70.0	64.6	70.0
BZR	69.4	74.9	69.5	76.7	78.5	74.2	77.9
COX2	64.6	72.7	67.4	69.2	70.1	68.5	66.9
DHFR	73.6	84.1	75.8	85.6	85.6	85.9	84.9
GPB	73.6	78.1	73.6	73.1	68.1	68.8	75.7
THER	78.9	74.8	77.7	78.6	78.8	81.6	76.3
THR	62.9	72.9	55.8	65.6	68.9	67.9	65.6
Mean	**70.8**	**75.6**	**71.2**	**75.1**	**75.2**	**73.7**	**74.7**
Mean rank	**5.63**	**3.63**	**4.75**	**3.25**	**2.63**	**4.13**	**3.63**

TABLE VIII
EXPERIMENT 2: FRAGMENT DESCRIPTORS, NO REDUCTION. AUC

Set	RBF	SVM	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	0.76	0.78	0.85	0.92	0.91	0.91	0.89
AchE	0.69	0.68	0.74	0.75	0.75	0.68	0.77
BZR	0.71	0.75	0.72	0.82	0.84	0.81	0.84
COX2	0.67	0.73	0.74	0.78	0.79	0.76	0.76
DHFR	0.75	0.84	0.83	0.93	0.92	0.92	0.93
GPB	0.82	0.78	0.87	0.88	0.84	0.87	0.83
THER	0.80	0.75	0.86	0.88	0.88	0.86	0.84
THR	0.64	0.72	0.59	0.74	0.72	0.74	0.72
Mean	**0.73**	**0.75**	**0.78**	**0.84**	**0.83**	**0.82**	**0.82**
Mean rank	**6.38**	**5.75**	**4.75**	**1.88**	**2.38**	**3.63**	**3.25**

When using the reduced Fragments feature set, GASEN and NegBagg were actually the two worst techniques regarding accuracy, see Table IX below. This is particularly interesting, since this is the feature set producing the best results overall. Here, the best option is to bag RBF networks, which is a very uncommon ensemble technique. It should be noted, however, that even single RBFs perform remarkably well, explaining most of the success for the RBF ensemble. Comparing GASEN to Bag-M_B, it is obvious that the post-processing performed by GASEN (i.e. selecting a subset of the available networks) most often will reduce the accuracy instead of increasing it.

TABLE IX
EXPERIMENT 2: FRAGMENT DESCRIPTORS, FEATURE REDUCTION. ACCURACY

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	85.3	87.8	87.9	86.0	87.8	88.7	88.7	87.8	88.7
AchE	78.2	80.9	72.0	79.1	80.9	79.1	78.2	74.6	76.4
BZR	82.2	83.5	79.2	82.2	83.5	80.4	79.8	79.2	80.4
COX2	71.8	70.2	71.1	72.1	71.5	68.2	67.6	67.3	68.2
DHFR	86.9	86.7	87.2	86.9	86.7	86.4	86.1	86.4	84.1
GPB	74.8	81.0	72.6	74.8	81.0	76.2	73.6	74.8	75.0
THER	75.0	85.2	80.2	74.8	85.2	76.3	78.9	75.2	73.8
THR	74.3	68.3	71.7	75.3	71.7	77.1	78.2	76.0	69.2
Mean	**78.6**	**80.4**	**77.7**	**78.9**	**81.0**	**79.0**	**78.9**	**77.6**	**77.0**
Mean rank	**4.75**	**3.50**	**5.50**	**4.25**	**2.75**	**3.88**	**5.25**	**6.38**	**6.50**

Considering, finally, the AUCs in Table X below, the most evident difference is that NegBagg here exhibits slightly better performance, outperforming at least BAG-M_B and GASEN. Still, bagging MLPs or RBFs, as well as single MLPs, in Weka and AVG-M_A, all obtain a lower mean rank than NegBagg.

TABLE X
EXPERIMENT 2: FRAGMENT DESCRIPTORS, FEATURE REDUCTION. AUC

Set	MLP	RBF	SVM	Bag-M_W	Bag-RBF	Bag-M_B	Avg-M_A	Gas	NB
ACE	0.94	0.92	0.88	0.95	0.94	0.97	0.97	0.96	0.97
AchE	0.85	0.83	0.72	0.83	0.82	0.81	0.81	0.80	0.84
BZR	0.86	0.86	0.79	0.86	0.88	0.87	0.87	0.87	0.89
COX2	0.83	0.81	0.71	0.82	0.82	0.80	0.81	0.79	0.81
DHFR	0.93	0.92	0.87	0.93	0.91	0.92	0.92	0.92	0.92
GPB	0.89	0.91	0.73	0.91	0.91	0.89	0.89	0.89	0.89
THER	0.85	0.94	0.80	0.88	0.92	0.86	0.90	0.83	0.87
THR	0.81	0.69	0.71	0.80	0.81	0.80	0.81	0.78	0.77
Mean	**0.87**	**0.86**	**0.78**	**0.87**	**0.88**	**0.87**	**0.87**	**0.86**	**0.87**
Mean rank	**3.88**	**4.50**	**8.88**	**3.38**	**3.38**	**5.13**	**3.63**	**6.00**	**4.13**

The results from Experiment 2 strengthen the analysis of Experiment 1. Overall, Negbagg showed amazingly poor performance, especially regarding accuracy. As a matter of fact, comparing Bag-M_B to NegBagg in Experiment 2, Bag-M_B obtained higher accuracy on 9 data sets while NegBagg won 3 and there were 4 ties. With NegBagg being an alleged "state-of-the-art" algorithm, this must be considered quite grave.

Analyzing the results for GASEN, the picture is equally gloomy. A pair-wise comparison with Bag-M_B shows that the selection of a subset of the available ANNs only rarely succeeds, increasing the accuracy on only three, and the AUC on only two, of the 16 data sets.

When using the larger Fragments feature set, the feature reduction was very successful, almost always leading to improved performance, for all techniques. Table XI below shows wins, draws and losses, for using the reduced feature set, compared to the original.

TABLE XI
EXPERIMENT 2: FRAGMENT DESCRIPTORS, EFFECT OF FEATURE REDUCTION

Technique	Accuracy			AUC		
	win	draw	loss	win	draw	loss
RBF	8	0	0	8	0	0
SVM	5	0	3	5	0	3
Bag-RBF	8	0	0	8	0	0
Bag-M_B	6	0	2	6	0	2
Avg-M_A	7	0	1	7	0	1
Gas	6	0	2	6	0	2
NB	5	0	3	7	0	1
Summary	**45**	**0**	**11**	**47**	**0**	**9**

Another comparison, potentially interesting for the medical domain is, of course, between the two available feature sets. As seen in Table XII below (showing wins, ties and losses for the 2.5D feature set when compared to Fragments) using 2.5D turns out to be beneficial for both accuracy and AUC.

TABLE XII
COMPARING 2.5D DESCRIPTORS TO FRAGMENT, NO REDUCTION

Technique	Accuracy			AUC		
	win	draw	loss	win	draw	loss
RBF	5	0	3	6	0	2
SVM	4	0	4	4	0	4
Bag-RBF	5	0	3	5	0	3
Bag-M_B	5	0	3	5	0	3
Avg-M_A	5	0	3	4	0	4
Gas	6	0	2	5	0	3
NB	4	0	4	2	0	6
Summary	**34**	**0**	**22**	**31**	**0**	**25**

When comparing the results for reduced feature sets, however, the picture is completely different; now Fragments is a better feature set than 2.5D for producing highly accurate models, see Table XIII below. As a matter of fact, in this study, using reduced Fragments feature sets was the choice leading to the best predictive performance, in general.

TABLE XIII
COMPARING 2.5D DESCRIPTORS TO FRAGMENT, REDUCED FEATURE SET

Technique	Accuracy			AUC		
	win	draw	loss	win	draw	loss
MLP	3	0	5	0	2	6
RBF	3	0	5	3	1	4
SVM	4	0	4	4	0	4
Bag-M_W	2	0	6	1	1	6
Bag-RBF	3	0	5	3	0	5
Bag-M_B	3	0	5	2	0	6
Avg-M_A	2	0	6	2	0	6
Gas	3	0	5	2	0	6
NB	3	0	5	1	0	7
Summary	**26**	**0**	**46**	**18**	**4**	**50**

V. CONCLUSION

In this study, most straightforward techniques evaluated were more than able to match the performance of the alleged state-of-the-art algorithms GASEN and NegBagg.

Especially NegBagg, which, after all, is a fairly recent algorithm, was constantly outperformed by most of the versions of standard bagging included in the study. Nevertheless, the results for GASEN were even more striking since the experimentation showed that it was actually better, on a large majority of the data sets, to use all available ANNs instead of applying GASEN to select a subset.

VI. Discussion and future work

First of all, it should be noted that the overall purpose of this study was not really to scrutinize GASEN and NegBagg. On the contrary, we believe that many recognized algorithms would suffer similar results if evaluated in the same way. Nevertheless, the results presented here clearly cast some doubts on sophisticated techniques in general, and of course on GASEN and NegBagg in particular. Furthermore, the results from a parallel study, using standard UCI benchmarking data sets, were quite similar, making it quite questionable if all these fancy ensemble techniques really will outperform standard choices like bagging outside the studies reported in the original papers.

Having said that, it would still be interesting to analyze the performances of the sophisticated algorithms further. Especially, looking at the relationships between base classifier accuracies, diversity and ensemble accuracy could potentially explain (or at least shed some light on) the reasons for the discouraging results.

Another interesting study would be to compare slightly different versions of the straightforward techniques. Should all ANN base classifiers use the same instances and features during training, or should this be somewhat randomized to increase diversity? Should different architectures be employed? Should each ANN base classifier be trained with or without an early stopping validation set? How many ANNs should be used? Would it be fruitful to mix MLPs with RBF networks? As seen above, there are a large number of options even when "simply averaging all available ANNs".

Finally, the experimentation indicates that feature reduction may often be beneficial, not only for reducing the training time, but also for predictive performance. Naturally, feature reduction in general has been heavily investigated by researchers. Studies targeting feature reduction for classifiers specifically trained to be part of ensembles are, however, quite rare. With this in mind, it would be interesting to look into methods aimed at somehow producing "optimal" different feature sets for the training of each base classifier. As a first try, we will develop an algorithm based on genetic algorithms for the feature selection. Within that framework, different fitness functions, potentially but not necessarily using actual classifications, will be evaluated.

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] 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.

[4] L. Breiman, Bagging predictors. *Machine Learning*, 24(2), pp. 123-140, 1996.

[5] Z.-H. Zhou, J.-X. Wu, Y. Jiang and S.-F. Chen. Genetic algorithm based selective neural network ensemble, 17*th International Joint Conference of Artificial Intelligence*, Vol. 2:797-802, Seattle, WA, 2001.

[6] Z.-H. Zhou, J.-X. Wu and W. Tang. Ensembling Neural Networks: Many Could Be Better Than All, *Artificial Intelligence*, Vol. 137, No. 1-2:239-263, Elsevier, 2002.

[7] M. M. Islam, X. Yao, S. M. Shahriar Nirjon, M. A. Islam and K. Murase, Bagging and boosting negatively correlated neural networks. *IEEE transactions on systems, man, and cybernetics, Part B: Cybernetics*, 38(3):771-84, 2008.

[8] Y. Liu and X. Yao, Ensemble learning via negative correlation, *Neural Networks*, vol. 12:1399–1404, 1999.

[9] I. H. Witten and E. Frank, Data Mining – *Practical Machine Learning Tools and Techniques*, Elsevier, 2005.

[10] 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.

[11] C. L. Bruce, J. L. Melville, S. D. Pickett and J. D. Hirst, Contemporary QSAR Classifiers Compared, *J. Chem. Inf. Model*, 47:219-227, 2007.

[12] J. J Sutherland, L. A. O'Brien and D. F. A. Weaver, Comparison of Methods for Modeling Quantitative Structure−Activity Relationships. *J. Med. Chem.*, 47:5541−5554, 2004.

[13] T. Fawcett, Using rule sets to maximize roc performance, *15th International Conference on Machine Learning*, pp. 445-453, 2001.

[14] Y. A. Bradley, The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(6):1145-1159, 1997.