WCCI 2012 IEEE World Congress on Computational Intelligence
June, 10-15, 2012 - Brisbane, Australia IJCNN

Producing Implicit Diversity in ANN Ensembles

Ulf Johansson
School of Business and IT
University of Borås
Sweden
Email: ulf.johansson@hb.se

Tuve Löfström
School of Business and IT
University of Borås
Sweden
Email: tuve.lofstrom@hb.se

***Abstract*—Combining several ANNs into ensembles normally results in a very accurate and robust predictive models. Many ANN ensemble techniques are, however, quite complicated and often explicitly optimize some diversity metric. Unfortunately, the lack of solid validation of the explicit algorithms, at least for classification, makes the use of diversity measures as part of an optimization function questionable. The merits of implicit methods, most notably bagging, are on the other hand experimentally established and well-known. This paper evaluates a number of straightforward techniques for introducing implicit diversity in ANN ensembles, including a novel technique producing diversity by using ANNs with different and slightly randomized link structures. The experimental results, comparing altogether 54 setups and two different ensemble sizes on 30 UCI data sets, show that all methods succeeded in producing implicit diversity, but that the effect on ensemble accuracy varied. Still, most setups evaluated did result in more accurate ensembles, compared to the baseline setup, especially for the larger ensemble size. As a matter of fact, several setups even obtained significantly higher ensemble accuracy than bagging. The analysis also identified that diversity was, relatively speaking, more important for the larger ensembles. Looking specifically at the methods used to increase the implicit diversity, setups using the technique that utilizes the randomized link structures generally produced the most accurate ensembles.**

I. Introduction

Predictive classification is the task of learning a target function f mapping each instance $\boldsymbol{x}$ to one of the predefined class labels y. The target attribute y (the class label) is a discrete attribute, restricted to values in a predefined set $\{c_1, c_2, \ldots, c_n\}$. When using machine learning techniques for predictive classification, the algorithm uses a set of training instances, each consisting of an input vector $\boldsymbol{x}_i$ and a corresponding target value y_i to learn the function $y = f(\mathbf{x}; \theta)$. During training, the parameter values θ are optimized, based on a score function. When sufficiently trained, the predictive model is able to accurately predict a value $\hat{y}$, when presented with a novel (test) instance $\boldsymbol{x_j}$.

Within machine learning, it is well established that combining several individual classifiers into *ensembles* all but guarantees improved predictive performance, compared to single models, see e.g., [1] and [2]. An ensemble aggregates multiple classifiers (called *base classifiers*) into a composite model, making the ensemble prediction a function of all the included base classifiers.

If the base classifiers $\mathbf{H} = \{h_1, h_2, \ldots, h_m\}$ of an ensemble are trained on the training set, the output (prediction) of a base classifier h_j on instance $\boldsymbol{x}_i$ is $h_j(\boldsymbol{x}_i)$. This prediction is, in the general case, a vector of size n consisting of beliefs (typically probability estimates) associated with each class. Often, however, each base classifier simply votes for one specific class, i.e., just produces a class label.

Given a set of trained base classifiers, together with a corresponding set of weights $\mathbf{W} = \{w_1, w_2, \ldots, w_m\}$, where $w_j \geq 0$ and $\sum_{j=1}^{m} w_j = 1$, the ensemble classifies each instance by choosing the class that receives the largest weighted vote. If all classifier weights are equal, the procedure is referred to as *majority voting* when the base classifiers output class labels, and *averaging* when they output beliefs.

The most intuitive explanation for why ensembles work is that the aggregation of several models, using averaging or majority voting, will eliminate uncorrelated base classifier errors; see e.g., [3]. Consequently, ensemble accuracy will normally be higher than mean base classifier accuracy, but only as long as the base classifiers commit their errors on different instances. Ideally, the base classifiers should make their mistakes independently. Informally, the key term *ensemble diversity* therefore measures and describes how base classifier mistakes are distributed over the instances.

In [4], Brown et al. introduced a taxonomy of methods for creating diversity. The first obvious distinction made is between *explicit* methods, where some diversity metric is directly optimized, and *implicit* methods, where the method is likely to produce diversity without actually targeting it. Several implicit methods produce diversity by supplying each classifier with a slightly different training set. Standard *bagging* [5] obtains diversity by using resampling to create different training sets for each base classifier. More specifically, each training set (called a *bootstrap*) has the same size as the original training set, but since the instances are randomly selected with replacement, a training set will contain multiple copies of some instances while lacking others completely. On average, approximately 63% of the orginal instances are present in each bootstrap. In the *random subspace method* [6], the base classifiers are trained on randomly chosen subspaces of the original attribute space, i.e., the individual training sets are instead sampled in the attribute (feature) space.

In contrast to, for instance, *random forests* [7], most dedicated ensemble techniques utilizing artificial neural networks (ANNs) as base classifiers have a tendency to be quite complicated, often explicitly optimizing some diversity metric. This is despite the fact that solid empirical, as well as theoretical, validation of the explicit algorithms, are absent from the field [8]. Actually, the current situation can be summarized like this: On the one hand, diversity is obviously beneficial for ensembles, but on the other hand, it is very questionable if any suggested diversity measure is actually useful as part of an optimization function.

Based on this, we argue that implicit methods for producing diversity in ANN ensembles should receive increased attention. Consequently, the overall purpose of this paper is to evaluate a number of basic techniques for introducing implicit diversity in ANN ensembles. The analysis focuses on the predictive performance of the ensembles, using base classifier accuracy and different diversity measures to explain and discuss the results obtained.

II. Background

The important result that ensemble error depends not only on the average accuracy of the base models but also on their diversity was formally derived by Krogh and Vedelsby in [9]. In their simple formula, the ensemble error *E* can be expressed as:

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

where $\bar{E}$ is the average base model error and $\bar{A}$ is the average ensemble diversity (ambiguity), measured as the weighted average of the squared differences in the predictions of the base models and the ensemble. In a regression context and using averaging to combine predictions, this is equivalent to:

$$E = (\hat{y} - y)^2 = \frac{\sum_{i=1}^{m}(h_i - y)^2}{m} - \frac{\sum_{i=1}^{m}(h_i - \hat{y})^2}{m} \tag{2}$$

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. Actually, it can be shown that if the base classifiers perform better than chance while making independent errors, the resulting error of the ensemble can be made arbitrarily small [10]. Although independent base models is impossible to achieve in practice, all ensemble methods strive to obtain diverse models while maintaining a sufficient level of accuracy for each model.

By assuming that the ensemble is a convex combined ensemble (e.g., using averaging), a bias-variance-covariance decomposition can be obtained for the ensemble MSE:

$$E = (\hat{y} - y)^2 = \overline{bias}^2 + \frac{1}{m}\overline{var} + (1 - \frac{1}{m})\overline{covar} \tag{3}$$

From this it is evident that the error of the ensemble depends critically on the amount of correlation between models, as quantified in the covariance term. As a matter of fact, there are several approaches, referred to as *negative correlated learning* [11] that explicitly minimize the covariance. Negative correlated learning has been applied mainly to ANN ensembles, two fairly recent algorithms are *negbagg* and *negboost* [12]. Still, it must be noted that although it has been shown that negative correlated learning directly controls the covariance term in the bias-variance-covariance trade-off for regression problems [13], it is not obvious how this applies to classification.

For classification ensembles, where the base classifiers are only able to output a class label, the outputs have no intrinsic ordinality between them, thus making the concept of covariance undefined. So, using a zero-one loss function, there is no clear analogy to the bias-variance-covariance decomposition. Instead, a large number of different diversity measures, all trying to quantify the contribution of the differences between the base classifiers to the overall ensemble accuracy have been suggested.

Diversity measures are often divided into pairwise and non-pairwise measures. Pairwise measures calculate the average of a particular distance metric between all possible pairings of classifiers in the ensemble, while non-pairwise measures typically use some variation of entropy, or calculate a correlation of each ensemble member with the averaged output. In this study, we will use two of the pairwise measures and one non-pairwise.

It must be noted that all diversity measures are in fact calculated on what is sometimes called an *oracle output matrix*, i.e., the correct target values are assumed to be known. Let the (oracle) output of each classifier D_i be represented as an N-dimensional binary vector y_i, where $y_{j,i} = 1$ if D_i correctly recognizes instance z_j and 0 otherwise. Let N^{ab} mean the number of instances for which $y_{j,i} = a$ and $y_{j,k} = b$. As an example, N^{11} is the number of instances correctly classified by both classifiers. N is the total number of instances.

The most intuitive diversity measure is probably the *disagreement* measure, which is the ratio between the number of instances on which one classifier is correct and the other incorrect to the total number of instances:

$$Dis_{i,k} = \frac{N^{10} + N^{01}}{N} \tag{4}$$

To find the diversity of a specific ensemble consisting of L classifiers, the averaged Dis over all pairs of classifiers is calculated:

$$Dis = \frac{2}{L(L-1)}\sum_{i=1}^{L-1}\sum_{k=i+1}^{L} Dis_{i,k} \tag{5}$$

Naturally a higher disagreement value implies a larger diversity. The *double-fault* measure was proposed in [14] and is the proportion of instances misclassified by both classifiers:

$$DF_{i,k} = \frac{N^{00}}{N} \tag{6}$$

To calculate the double fault diversity for an ensemble, the double fault values are averaged over all pairs of classifiers, identically to disagreement. For double fault, a lower value indicates higher diversity. Comparing double fault and disagreement, the main difference is that when using disagreement, N^{00} and N^{11} are treated equally, while the double fault measure is not "punished" when both classifiers are correct. In a setting where the purpose is to obtain accurate ensembles, this may appear to be a strong argument for double fault, but it should be noted that, as a consequence, double fault will be positively correlated with the base classifier accuracy.

The non-pairwise *difficulty* measure was introduced by Hansen and Salomon in [10]. Let X be a random variable taking values in $\{0/L, 1/L, \ldots, 1\}$. X is defined as the proportion of classifiers that correctly classify an instance x drawn randomly from the data set. To estimate X, all L classifiers are run on the data set. The difficulty is then defined as the variance of X. For difficulty, lower values mean higher diversity, with the explanation that for a diverse classifier ensemble, every instance can at least be classified correctly by a portion of all the base classifiers, which is likely to result in a lower variance. The opposite would mean that all base classifiers are correct on some instances and wrong on some other instances, which of course would lead to higher variance.

Despite the fact that both intuition and a strong theoretical foundation advocate the benefit of diverse base classifiers, none of the suggested diversity measures is proven superior to the others. As a matter of fact, when Kuncheva and Whitaker studied ten statistics measuring diversity using oracle 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 [15]. In [16], Saitta supports Kuncheva's negative view, as presented in a series of papers, but she also goes one step further and shows not only that no useful measure exists today, but also that it is unlikely that one will ever exist.

Clearly, these results seem to favor implicit methods. As a matter of fact, it may be argued that the best use for diversity is not during the optimization, but rather as a tool for analysis and explanation.

A. Related work.

Naturally, most standard methods for building ensembles have been applied to (and have often been modified for) ANN base classifiers. As an example, both standard bagging and boosting [17] can be readily used on ANN classifiers. Comparing bagging and boosting, bagging has one inherent advantage since the models can be trained in parallel.

To produce more diverse ANN ensembles, Maclin and Shavlik [18] deliberately initialized the weights so different base classifiers would start out in different parts of the weight space. Cherkauer [19], suggested another, even more straightforward method, when producing diversity simply by using different number of hidden nodes in the base classifiers. Oza and Tumer [20] suggested using different subsets of the input features for each ANN, similar to the random subspace method. In a previous study, we evaluated some basic methods for producing implicit diversity in ANN ensembles [21]. The main result was that for the fairly small ensembles studied, standard bagging was actually most often ineffective. Using heterogeneous architectures, on the other hand, generally improved the predictive performance.

More advanced implicit methods include the DECORATE algorithm [22], which creates the diversity by adding different artificial training instances to each base classifier training set. Similarly, Raviv and Intrator [23], proposed a method combining bagging, weight decay and artificial noise to produce the diversity.

The negative correlation learning algorithm [11], mentioned above, trains ANNs simultaneously and interactively, trying to force different ANNs to learn different aspects of the data by introducing a correlation penalty term into the error function.

Finally, there are several evolutionary methods where diversity is a part of a fitness function and some evolutionary algorithm is used to search for an accurate ensemble. One example is the ADDEMUP method, suggested by Opitz and Shavlik in [24]. A very good survey of methods for producing diverse base classifiers, including ANNs, can be found in [4].

III. Method

As described in the introduction, the overall purpose is to compare different ways of producing implicit diversity in ANN ensembles. More specifically, we evaluate four different techniques; varying the number of epochs, using different versions of (instance) bagging, varying the architectures and manipulating the features.

Before the experimentation, each data set was preprocessed in the following way: first all missing numerical values were replaced with the mean value of that specific attribute, while missing categorical values were replaced with the mode values. Secondly, all categorical attributes were converted into binary numerical attributes. All ANNs in the study are fully-connected MLPs utilizing a localist coding (i.e., the number of output units is equal to the number of classes), and the training used is the resilient backpropagation (rprop) learning algorithm. In the first experiment, all ensembles consist of 15 ANNs and in the second experiment there are 51 ANNs in the ensembles. The base classifiers are always combined using majority vote.

In the baseline setup, each and every ANN in the ensemble is trained exactly 150 epochs, no bagging is used (i.e., every ANN is trained on all training data), all MLPs have identical architectures with one hidden layer (for details see below) and there is no manipulation of the features. Naturally, we would expect this setup to produce fairly accurate but quite similar (i.e., not diverse) models.

It is well-known that there is no general rule-of-thumb that will always find an optimal, or even acceptable, number of hidden units in an MLP, based on the data set characteristics. In practice, some kind of internal cross-validation is instead often used to determine the number of hidden units. Nevertheless, there are many rules-of-thumb proposed, and most of them suggest that the number of hidden units should be somewhere between the number of input units and the number of output units, thus resulting in pyramid shaped networks. In this study, we reluctantly decided to use a rule-of-thumb for determining the number of hidden units in the baseline setup in order to simplify the experimentation. More specifically, the number of hidden units *h*, in the baseline setup, is calculated using (7).

$$h = \left\lfloor \frac{\#attributes + \#classes}{2} \right\rfloor \tag{7}$$

A. Setups evaluated

Regarding training, we evaluate two different options; either all ANNs are trained exactly 150 epochs, or the number of epochs is randomized between 100 and 200 for each ANN.

As described in the introduction, standard bagging utilizes bags of the same size as the original training set. It is of course, however, possible to use different sizes for the bags, and the bag size can even be varied between the base classifiers. Here we evaluate three settings; no bagging, standard bagging (bag size 100%) and varying bag sizes where the bag size for each base classifier is randomized between 70% and 120%.

In the baseline setup, all ANNs have identical architectures with one hidden layer where the number of hidden units is determined by the heuristics above. In addition, we evaluate two more settings where the architecture is slightly randomized for each base classifier. In the second setting, all ANNs still have one hidden layer, but the number of hidden units is randomized between $0.5h$ and $1.5h$, where h is the number of hidden units given by the heuristic. In the third setting, the number of hidden layers is first randomized (one or two) for each base classifier. ANNs with one hidden layer have a randomized number of hidden units, identically to the previous setting. For ANNs with two hidden layers, the number of hidden units in the first hidden layer is randomized between $0.5h$ and h, while the number of hidden units in the second hidden layer is randomized between $0.3h_1$ and $0.5h_1$, where h_1 is the number of hidden units in the first hidden layer.

Regarding manipulating the features, the baseline setup of course just uses all features when training every base classifier. In the second setting, the random subspace method is employed. Here, each base classifier is trained using only 90% of the features. It must be noted that which features to remove is randomized for each ANN, and that the feature reduction is performed before the categorical attributes are converted into numerical binary attributes. The third setting, finally, uses ANNs with different (randomzed) link structures. In more detail, starting from a fully-connected MLP, a certain proportion of all links between input units and hidden units, between hidden units in different layers and between hidden units and output units are removed. Exactly which incoming links that are removed is randomized for each ANN and for each hidden and output unit. In ANNs with one hidden layer, 40% of the incoming links to each hidden unit, and 20% of the incoming links to each output units are removed. When there are two hidden layers, the corresponding parameter values are 60% between the input layer and the first hidden layer, 40% between the hidden layers, and 20% between the second hidden layer and the output layer. This setting is heavily inspired by the random forest technique, where the features available for each split when building the random trees is randomized. It is, however, to the best of our knowledge, a novel way of introducing implicit diversity in ANN ensembles.[1] It must be noted that the links are removed *before* training starts, so it is not a pruning technique but rather a technique for producing implicit diversity by varying the architectures. For simplicity, we refer to the resulting ANNs as *sparse nets*.

All in all, the four settings described above can be combined into the 54 different setups evaluated here, see Table I below.

TABLE I
EXP. 1 - SETUPS EVALUATED

setting	1	2	3
Training	150 epochs	[100, 200] epochs	-
Bagging	No bagging	Bag 100%	Bag [70, 120]%
Architecture	Identical	Random #units	Random #layers and #units
Features	All	90%	Sparse nets

In the rest of this paper, the following convention is used for referring to the different setups: All setups are described using four digits. Each digit represents, in order, the settings for *training*, *bagging*, *architectures* and *features*. As an example, 1-1-1-1 is the baseline setup, while 1-2-2-3 would mean that all ANNs were trained exactly 150 epochs, that standard bagging was used, that all ANNs had one hidden layer where the number of hidden units was randomized, and that links were randomly removed to produce sparse nets, according to the procedure described above.

B. Experimental setup

During experimentation, the data sets were randomly split in 75% training and 25% testing. This was repeated ten times, so all results are averaged over ten runs. The 30 data sets used are all well-known UCI data sets [25]. For the sake of completeness, they are listed here: *breast-w*, *cmc*, *colic*, *credit-a*, *credit-g*, *dermatology*, *diabetes*, *ecoli*, *glass*, *haberman*, *heart-c*, *heart-h*, *heart-s*, *hepatitis*, *ionosphere*, *iris*, *labor*, *liver-disorders*, *lung-cancer*, *lymph*, *sonar*, *spect*, *spectf*, *tae*, *tic-tac-toe*, *vehicle*, *vote*, *waveform*, *wine* and *zoo*.

IV. RESULTS

Table II below shows the average results over all data sets for the 54 setups in the first experiment. Starting with ensemble accuracies, the first reflection is probably that the differences, when comparing results averaged over all data sets, are quite small. This, together with the fact that the mean base classifier accuracies vary a lot more, is actually a very reassuring observation, confirming the robustness of ANN ensembles in general. Or, put in another way, it is obvious that a drop in base classifier accuracy, as a result of the different techniques, is more than made up for by the increase in diversity. Looking specifically at the disagreement results, it is interesting to note that the spread is fairly large, from 10.6% to 18%. As expected, the most effective way of

[1] This technique is the basis of a novel algorithm, tentatively named *random brains*, which is developed and thoroughly evaluated in an ongoing study.

introducing diversity is bagging, all setups using bagging have higher average disagreement than any setup not using bagging.

TABLE II
RESULTS FOR 15 ANN ENSEMBLES SORTED ON ENSEMBLE ACCURACY

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-2-1-3	.823	.791	.074	.158	.130	1-1-3-1	.819	.800	.087	.121	.140
1-1-3-3	.823	.796	.083	.133	.137	2-1-1-1	.819	.801	.092	.111	.144
1-1-2-3	.823	.804	.089	.111	.141	2-3-1-1	.819	.787	.074	.163	.131
2-1-3-3	.822	.798	.083	.132	.136	1-2-2-1	.818	.786	.075	.162	.132
2-2-1-3	.822	.789	.075	.158	.132	2-1-3-1	.818	.800	.087	.121	.139
1-2-3-3	.822	.781	.071	.175	.131	1-1-3-2	.818	.795	.091	.122	.144
1-3-1-3	.821	.790	.075	.158	.131	1-1-1-1	.817	.803	.093	.107	.144
2-1-2-3	.821	.804	.089	.112	.141	1-3-2-1	.817	.785	.075	.164	.132
2-1-1-3	.821	.805	.091	.106	.142	1-1-1-2	.816	.799	.096	.107	.148
1-2-2-3	.821	.788	.074	.160	.132	2-1-2-2	.816	.796	.096	.112	.148
1-2-1-1	.821	.787	.074	.163	.131	1-1-2-1	.816	.801	.092	.111	.144
2-3-1-3	.821	.789	.075	.160	.131	1-3-3-3	.815	.778	.071	.177	.133
1-3-2-3	.821	.786	.074	.163	.132	1-3-1-2	.815	.783	.077	.164	.135
1-3-1-1	.821	.787	.074	.163	.132	2-3-1-2	.814	.780	.079	.165	.138
2-3-2-1	.821	.786	.075	.163	.133	2-1-3-2	.814	.794	.089	.124	.144
2-2-1-1	.821	.787	.074	.163	.131	2-3-3-2	.814	.778	.075	.173	.136
2-3-3-3	.820	.778	.070	.180	.132	2-2-3-2	.814	.776	.077	.172	.138
1-3-3-1	.820	.783	.072	.172	.131	2-2-1-2	.813	.778	.080	.165	.140
2-2-2-3	.820	.786	.075	.161	.133	1-2-2-2	.813	.776	.079	.169	.139
2-3-2-3	.820	.787	.075	.162	.133	1-3-3-2	.812	.774	.076	.177	.137
1-1-1-3	.820	.804	.091	.107	.143	1-3-2-2	.811	.779	.078	.168	.137
2-3-3-1	.820	.783	.072	.170	.132	1-2-1-2	.811	.777	.079	.165	.140
2-2-3-3	.819	.780	.072	.176	.132	2-2-2-2	.810	.778	.079	.162	.141
2-1-2-1	.819	.801	.092	.111	.144	2-1-1-2	.810	.793	.097	.111	.152
2-2-2-1	.819	.786	.074	.164	.132	1-1-2-2	.809	.791	.096	.113	.152
2-2-3-1	.819	.783	.072	.171	.131	1-2-3-2	.806	.772	.078	.171	.142
1-2-3-1	.819	.783	.072	.171	.131	2-3-2-2	.804	.773	.079	.168	.143

To extend this analysis, Table III below shows the average ranks for each setup over all data sets.

TABLE III
15 ANN ENSEMBLES - RANKS SORTED ON ENSEMBLE ACCURACY

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-1-3-3	20.2	18.2	30.9	37.4	28.1	2-1-1-1	27.9	25.5	32.7	33.9	28.3
1-2-1-3	20.3	23.2	23.7	27.1	19.6	2-3-1-2	28.2	33.0	25.4	21.5	24.7
2-1-3-3	20.9	18.2	30.0	36.8	26.8	1-1-3-2	28.5	20.0	33.1	40.6	29.8
2-1-2-3	21.0	10.9	36.9	45.7	33.8	2-2-1-2	28.6	33.5	26.6	24.1	26.4
1-3-2-3	21.4	28.9	22.9	22.2	23.3	2-3-1-1	28.8	31.9	22.0	21.5	22.4
1-1-2-3	21.6	10.1	36.4	45.7	32.5	1-2-2-2	28.8	34.1	24.4	21.0	24.9
1-2-2-3	22.3	27.0	25.5	24.8	23.6	2-1-2-1	29.0	13.7	38.0	45.1	37.6
2-2-1-3	22.5	25.8	27.5	27.2	25.8	1-3-3-3	29.1	37.2	17.5	14.2	20.2
2-3-2-3	23.5	29.8	24.0	22.9	22.5	1-2-2-1	29.4	31.7	23.4	23.0	25.4
1-2-3-3	23.8	33.9	18.1	15.6	19.0	1-1-3-1	29.4	16.8	35.4	41.9	34.8
2-1-1-3	23.9	9.8	36.8	47.1	33.3	1-3-2-1	29.8	34.6	23.5	19.2	27.1
2-2-2-3	24.3	30.0	26.8	23.8	25.9	2-1-2-2	30.0	16.1	36.6	45.7	33.3
1-3-1-3	24.5	26.8	25.0	26.0	22.4	1-3-1-2	30.2	32.6	25.2	21.9	26.1
2-3-3-3	24.5	37.1	16.1	11.7	19.3	2-1-3-1	30.4	15.8	35.7	41.1	34.5
2-3-1-3	24.5	27.4	24.4	24.2	22.0	1-1-1-1	30.8	12.8	38.9	47.1	38.8
2-2-3-3	24.8	35.1	20.4	16.0	23.0	2-1-3-2	30.8	21.6	34.8	40.6	32.9
1-3-3-1	24.9	37.5	18.7	11.3	22.0	2-2-3-2	30.9	38.1	22.1	16.3	23.0
2-3-2-1	24.9	32.7	24.3	21.2	25.9	1-3-3-2	31.0	36.5	19.1	14.7	20.6
1-1-1-3	25.3	10.2	38.1	47.2	34.7	2-3-3-2	31.0	36.6	24.3	12.9	25.9
1-2-3-1	25.8	37.5	18.9	12.4	23.0	1-1-2-1	31.2	14.5	37.9	45.3	38.4
1-2-1-1	26.0	29.1	24.4	23.4	26.6	2-2-2-2	32.3	34.5	29.1	23.2	30.7
2-2-1-1	26.3	29.3	23.3	23.5	26.6	1-3-2-2	32.8	35.5	22.2	19.8	24.8
1-1-1-2	26.4	15.3	39.3	46.9	34.7	1-2-1-2	33.2	35.3	26.3	22.6	29.1
1-3-1-1	26.6	32.2	23.8	20.2	26.6	1-2-3-2	33.5	38.6	26.0	16.4	29.2
2-2-2-1	26.9	31.3	25.0	21.1	26.9	2-1-1-2	34.5	20.9	38.2	44.9	35.7
2-2-3-1	27.0	36.8	21.0	12.2	23.6	1-1-2-2	35.0	22.6	38.5	44.0	36.5
2-3-3-1	27.0	38.2	20.2	12.2	23.8	2-3-2-2	39.1	39.5	26.7	18.9	29.1

Again, the first impression is probably that the differences are fairly small. Looking at ensemble accuracies, the best average rank is 20.2 and the worst 39.1 (with 54 setups the average is of course 27.5). Nevertheless, it is very interesting to observe that the most important setting is clearly the last; i.e., how the features are manipulated. As a matter of fact, the best 10 setups all use the sparse nets setting. At the same time, 11 of the worst 12 setups all use the random subspace setting. Specifically comparing all setups against the baseline, Table IV below shows the number of wins for the different setups.

TABLE IV
15 ANN ENSEMBLES - WINS AGAINST BASELINE SETUP

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-1-3-3	**22.5**	13.5	**27**	**23.5**	**25**	2-3-2-1	17.5	*1*	**29**	**30**	**26.5**
1-2-1-3	**22**	*7*	**28**	**29**	**27**	2-3-3-1	17.5	*1*	**27**	**29**	**27**
1-3-2-3	**22**	*2.5*	**29**	**29**	**25**	2-2-2-1	17	*1*	**28**	**30**	**26.5**
2-1-2-3	**22**	17	**24.5**	13.5	**24.5**	2-2-3-1	17	*1*	**28**	**29**	**27**
1-1-2-3	**21.5**	18.5	**23**	14	**23**	2-3-1-3	17	*5.5*	**28**	**29**	**25**
2-1-1-3	**21**	18.5	18	13.5	**21**	1-1-3-2	16.5	13	14.0	**26**	16
2-1-3-3	**20.5**	15	**27**	**23.5**	**27**	1-1-3-1	16	*8*	**26.5**	**27**	**23**
1-2-3-1	**20**	*1*	**28**	**29**	**27**	1-3-2-1	16	*0.5*	**26.5**	**30**	**26.5**
1-3-1-3	**20**	*4.5*	**27**	**29**	**25.5**	2-3-1-2	15.5	*5*	**22**	**29**	18
1-2-2-1	19.5	*1.5*	**28**	**30**	**26**	1-2-2-2	15	*5.5*	**20**	**30**	19
1-1-1-3	19	17	**20**	13	**24**	1-3-1-2	15	*7.5*	**22**	**29**	**20**
1-2-3-3	19	*2*	**28**	**29**	**26**	2-1-1-1	15	14	15.5	15	16
2-2-1-3	19	*5*	**28**	**29**	**26**	2-1-2-2	15	15	9	15	13.5
2-2-2-3	19	*1.5*	**29**	**29**	**28**	2-1-3-2	15	12	17	**26**	16
2-3-3-3	19	*2.5*	**28**	**29**	**25**	2-1-3-1	14.5	*6*	**26.5**	**28**	**25**
1-2-1-1	18.5	*3.5*	**29**	**30**	**28**	2-3-3-2	14	*7*	**24**	**29**	19.5
1-2-2-3	18.5	*3.5*	**28**	**29**	**27**	1-1-2-1	13.5	12	16.0	**20.5**	15.5
1-3-1-1	18.5	*1.5*	**29**	**30**	**28**	1-3-3-2	13.5	*6*	**21.5**	**29**	19
2-2-1-1	18.5	*1*	**29**	**30**	**28**	2-2-3-2	13	*5.5*	**22**	**29**	19
2-3-2-3	18.5	*2.5*	**28**	**29**	**26.5**	1-2-3-2	12.5	*5.5*	**23**	**29**	19
1-3-3-1	18	*1*	**27**	**29**	**27.5**	2-2-2-2	12	*4*	**22**	**30**	18
2-2-3-3	18	*3*	**28**	**29**	**26**	1-3-2-2	11.5	*4*	**23**	**30**	**20**
1-1-1-2	17.5	15	*9*	12.5	13	1-1-2-2	11	11	13	15	13
1-3-3-3	17.5	3	**28**	**29**	**27**	1-2-1-2	11	*6*	**22**	**29**	19
2-1-2-1	17.5	11.5	19	**24**	17	2-1-1-2	*10*	13.5	11	17	13
2-2-1-2	17.5	5	**21**	**29**	**20**	2-3-2-2	*9*	*4*	**21**	**29**	15.5
2-3-1-1	17.5	2	**29**	**29**	**27**						

Over 30 data sets, a standard sign test requires 20 wins for a significant difference. In Table IV, a bold and underlined number indicates that the setup is significantly better (has higher accuracy or is more diverse) than the baseline setup, while an italicized and underlined number indicates that the baseline setup was significantly better.

Despite the fact that a large majority of all the setups outperformed the baseline setup, only nine produced significantly more accurate ensembles. Of these nine setups, all but one utilize the sparse net setting. Looking at base classifier accuracy and the different diversity measures, a majority of the setups have significantly less accurate base classifiers, but also significantly higher diversity. This is probably what should have been expected, but it is still noteworthy to see that all methods evaluated were indeed capable of increasing the diversity, but almost always at the expense of less accurate base classifiers. Finally, it is particularly interesting that four of the best setups, all utilizing the sparse nets setting but no bagging, obtained their high ensemble accuracy without an increase in disagreement, compared to the baseline setup. On the other hand, for these setups, the base classifier accuracy was comparable, or even slightly better, than the baseline setup. Most importantly, the diversity was significantly higher when measured using a more informed criterion. This is, of course, a very strong indicator that the diversity produced by that method is effective for increasing ensemble accuracy.

Since bagging is arguably the most important basic ensemble creation method, Table V below shows the number of wins for the different setups against bagging.

TABLE V
15 ANN ENSEMBLES - WINS AGAINST BAGGING

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-2-1-3	19.5	18	15	11.5	20.5	2-2-1-2	14	13	*9*	12	11
1-1-2-3	18.5	**25.5**	*4*	*2*	*7*	2-2-2-1	14	14.5	15	19	14
2-3-2-3	18.5	13	13.5	13	15	1-2-2-1	13.5	14.5	12	18	13
1-2-2-3	18	15	11.5	11.5	12.5	2-1-2-2	13.5	**22**	*2*	*1*	*4*
1-1-3-3	17.5	19	*8*	*6.5*	*7.5*	2-2-3-1	13.5	*7*	**21**	**27**	16.5
2-1-2-3	17.5	**24.5**	*5*	*2*	*7*	2-3-1-2	13.5	15.5	*10*	14	13
2-1-3-3	17.5	**20.5**	*6*	*7*	10.5	1-1-3-1	13	**25**	*2*	*1*	*4*
1-3-1-3	17	16.5	13	13.5	17	1-2-2-2	13	14	11	17	12.5
2-2-1-3	17	17.5	11.5	12	14	1-3-3-2	13	12	12	19	15.5
2-2-3-3	17	12	18	16	14.5	1-3-3-3	12.5	10.5	**20.5**	18.5	15.5
1-2-3-3	16.5	13	**22**	17.5	16.5	2-1-2-1	12.5	**29**	*1*	*0*	*3*
1-3-3-1	16.5	*4.5*	**25.5**	**27**	16.5	2-1-3-2	12.5	**19**	*2*	*3*	*7*
2-1-1-3	16.5	**25.5**	*2*	*2*	*5*	1-1-3-2	12	**20.5**	*2*	*2*	*6.5*
2-2-2-3	16.5	13.5	13	13.5	15.5	1-3-2-1	12	10	13	**22.5**	13
2-3-1-3	16.5	14.5	14	11.5	18	2-1-3-1	12	**25.5**	*2*	*1*	*5*
2-3-2-1	16.5	13	16.5	18.5	15	2-3-1-1	12	13.5	16.5	**21.5**	15.5
1-1-1-3	16	**25.5**	*3*	*2*	*6*	2-3-3-2	12	11	14	**25**	15
1-3-2-3	16	15	16	14	15	2-2-2-2	11.5	14	11	13.5	12
2-3-3-3	16	*9.5*	**22**	**22**	16	2-2-3-2	11.5	*9*	13.5	22	13
1-3-1-1	15.5	10.5	13.5	**23.5**	13	1-1-2-1	10.5	28	*2*	*0*	*2*
2-2-1-1	15	18	16	15.5	17.5	1-2-1-2	10.5	12	*8.5*	12	*10*
1-2-3-1	14.5	*8*	**23**	**26.5**	17.5	1-3-2-2	10.5	12	*8*	15	11
1-3-1-2	14.5	14	11	14	14	1-1-2-2	*10*	20	*2*	*2*	*6.5*
2-3-3-1	14.5	*4*	20	27	15	1-2-3-2	*10*	*10*	12	**21**	12
1-1-1-2	14	**22**	4	1	8	2-1-1-2	*10*	19	*2*	*2*	*5*
2-1-1-1	14	15	10	11	14	2-3-2-2	*7.5*	*8.5*	*8*	19	*7*

First of all it can be noted that standard bagging actually fares pretty well. No setup was significantly better than bagging, and a majority of setups lose more data sets than they win against bagging. There are, however, several setups that outperform bagging, and almost all of them utilize the sparse nets setting. In addition, looking at it the other way around, all setups but one using the sparse nets setting, won at least 16 data sets against bagging.

Summarizing Experiment 1, all techniques for producing diversity were successful. Specifically, even if base classifier accuracies decreased, the resulting ensembles were most often more accurate than the baseline setup. The only exception was the random subspace setting, where the resulting base classifier accuracies actually were too low for the diversity to compensate for. Comparing the different settings, it is obvious that using sparse nets was the most successful. As a matter of fact, almost all top ranked setups utilized the sparse net setting.

Turning to Experiment 2, where the ensembles consist of 51 ANNs, Table VI below shows the averaged results over all data sets.

TABLE VI
RESULTS FOR 51 ANNS ENSEMBLES, SORTED ON ENSEMBLE ACCURACY

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-2-1-3	.826	.790	.072	.156	.132	2-3-1-1	.822	.786	.070	.164	.132
2-2-3-3	.826	.781	.067	.176	.131	1-1-3-1	.822	.800	.084	.121	.139
1-3-3-3	.825	.779	.066	.180	.131	1-3-2-1	.821	.785	.070	.166	.132
2-1-2-3	.824	.803	.087	.111	.141	2-2-2-2	.821	.783	.075	.162	.136
1-3-1-3	.824	.789	.071	.160	.131	2-1-1-3	.821	.804	.088	.108	.142
1-2-2-3	.824	.789	.071	.160	.131	2-1-3-1	.821	.800	.084	.121	.139
1-2-3-3	.824	.781	.067	.176	.131	2-1-2-1	.820	.801	.089	.112	.144
2-1-3-3	.824	.797	.080	.132	.137	2-3-2-2	.819	.783	.074	.163	.135
1-1-3-3	.824	.797	.080	.132	.137	1-1-1-2	.819	.798	.093	.108	.148
2-3-3-3	.824	.779	.067	.178	.131	1-1-2-1	.819	.800	.089	.112	.144
1-3-2-3	.823	.787	.070	.163	.131	2-1-1-1	.817	.802	.090	.109	.144
2-3-1-3	.823	.787	.071	.160	.133	1-1-1-1	.817	.802	.090	.107	.145
2-2-1-3	.823	.790	.072	.157	.132	1-2-3-2	.817	.778	.072	.173	.136
1-1-1-3	.822	.805	.088	.106	.142	1-1-3-2	.817	.794	.088	.122	.144
2-2-3-1	.822	.784	.069	.169	.131	2-1-2-2	.816	.796	.093	.113	.147
1-2-2-1	.822	.787	.071	.162	.132	2-2-1-2	.816	.778	.076	.165	.140
1-3-1-1	.822	.786	.070	.165	.131	1-3-1-2	.816	.778	.074	.167	.139
1-3-3-1	.822	.783	.068	.172	.131	2-1-3-2	.816	.794	.086	.124	.144
2-3-3-1	.822	.783	.068	.172	.131	1-3-3-2	.815	.771	.072	.179	.139
2-3-2-1	.822	.786	.070	.165	.131	2-3-3-2	.815	.776	.073	.172	.138
2-2-2-3	.822	.788	.071	.160	.132	1-2-2-2	.815	.776	.076	.168	.140
2-2-2-1	.822	.787	.071	.162	.132	1-3-2-2	.814	.775	.074	.170	.139
1-1-2-3	.822	.803	.087	.112	.141	1-1-2-2	.812	.791	.093	.113	.152
2-2-1-1	.822	.788	.072	.159	.133	2-3-1-2	.811	.773	.076	.170	.142
2-3-2-3	.822	.787	.070	.163	.131	2-2-3-2	.811	.774	.072	.174	.139
1-2-3-1	.822	.785	.068	.169	.131	2-1-1-2	.810	.793	.095	.110	.153
1-2-1-1	.822	.787	.072	.160	.133	1-2-1-2	.808	.771	.076	.170	.144

Comparing this to the results in Table II, it can be noted that the ensemble accuracies are, as expected, slightly higher when there are more base classifiers. As a side note, since both double fault and disagreement are pairwise measures, they are, just like the mean base classifier accuracy, insensitive to the size of the ensemble. Or, put in another way, the increased ensemble accuracy when using more base classifiers can not be explained using these two diversity measures. Difficulty, on the other hand, has the proper behavior since it decreases (i.e., shows an increase in diversity) for the larger ensembles.

Looking at the overall ranks in Table VII below, the picture is quite similar to Experiment 1. One important difference is, however, that the baseline setup here is one of the worst. So, when using larger ensembles, diversity appears to become even more important; i.e., larger ensembles favor using less accurate but more diverse base classifiers. This is also evident from the fact that several of the worst setups (with regard to ensemble accuracy) are ranked among the last in diversity. Looking at the specific settings, the use of sparse nets was again the most successful. Remarkably, 17 of the 18 setups using the sparse net setting were among the best 18 over all. Using the random subspace setting still appears to be the worst choice, but this is less apparent for the larger ensembles. Actually, combining the random subspace setting with bagging obtained relatively accurate ensembles.

TABLE VII
51 ANN ensembles - Ranks sorted on ensemble accuracy

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
2-1-3-3	20.7	18.2	32.4	37.6	29.5	2-3-3-1	27.2	38.4	18.8	10.0	22.7
1-2-1-3	20.7	24.0	26.0	29.0	20.6	1-3-2-1	27.5	35.6	23.4	16.9	28.1
2-2-3-3	20.7	34.5	16.4	14.8	17.9	1-2-2-2	27.7	34.9	26.0	22.3	26.5
2-1-2-3	21.5	10.1	37.3	45.9	35.1	2-1-1-3	27.8	9.5	38.1	47.6	35.1
1-3-1-3	21.5	27.6	25.3	24.0	22.5	2-2-2-1	27.8	30.9	24.4	23.9	26.0
1-1-3-3	22.3	17.3	32.3	38.3	29.8	1-1-3-1	27.9	17.2	34.2	40.7	33.4
1-2-3-3	22.3	34.9	18.7	15.5	20.4	1-2-1-1	28.3	31.3	26.6	24.7	28.4
1-3-3-3	22.4	37.1	15.2	11.1	17.8	1-1-1-2	28.9	15.8	39.6	46.7	35.7
2-3-3-3	22.5	37.1	16.1	13.2	17.5	2-3-2-2	29.2	32.1	25.0	21.6	24.7
1-2-2-3	22.9	26.8	26.3	24.9	21.9	1-3-2-2	29.7	37.0	23.4	18.3	24.6
2-3-1-3	23.5	29.2	25.8	24.4	24.3	2-1-3-1	29.8	16.6	34.3	40.7	34.8
1-3-2-3	23.7	30.6	23.3	21.4	21.4	1-3-1-2	29.9	34.8	20.4	18.8	21.6
1-1-1-3	24.3	8.0	38.9	47.9	36.4	1-3-3-2	29.9	41.5	19.0	10.7	21.7
2-2-2-3	24.4	27.7	26.9	25.5	23.5	2-1-3-2	30.0	22.5	34.4	40.0	32.1
2-3-2-3	24.7	29.3	21.7	21.1	20.4	1-2-3-2	31.2	36.0	23.7	15.4	25.5
2-2-1-3	24.8	25.2	28.1	28.6	23.2	2-1-2-1	31.4	13.3	37.6	45.0	38.7
1-3-3-1	25.2	39.4	16.7	9.7	21.2	1-2-1-2	31.4	35.7	26.6	20.4	30.2
1-1-2-3	25.4	10.6	36.1	44.7	33.5	1-1-3-2	32.1	22.1	33.6	40.1	30.3
2-3-2-1	25.9	33.3	22.1	18.2	25.1	2-3-3-2	32.2	39.3	21.8	13.7	23.1
1-2-2-1	26.2	32.1	26.4	22.4	28.3	2-1-2-2	32.4	15.5	36.8	45.3	34.0
2-2-3-1	26.2	37.7	19.6	13.3	22.8	1-1-2-1	32.8	13.8	37.3	44.8	38.5
2-2-1-2	26.9	33.6	27.2	24.6	25.7	2-3-1-2	32.9	36.8	24.1	20.1	24.6
2-3-1-1	26.9	33.8	23.0	20.1	26.0	2-2-3-2	33.5	37.5	22.1	16.1	25.3
2-2-1-1	27.0	29.4	26.4	25.9	27.2	2-1-1-1	33.6	12.6	38.1	46.6	38.5
2-2-2-2	27.0	32.2	28.1	23.7	27.8	1-1-1-1	34.9	12.9	38.7	47.5	39.1
1-3-1-1	27.1	32.9	23.0	19.5	25.6	1-1-2-2	35.3	21.5	38.9	44.3	37.4
1-2-3-1	27.1	36.5	19.5	13.4	22.6	2-1-1-2	36.3	21.1	39.8	46.1	37.2

The results in Table VIII below show that when using the larger ensembles, a majority of the setups evaluated produced significantly more accurate ensembles, compared to the baseline setup.

TABLE VIII
51 ANN ensembles - Wins against baseline

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
2-1-2-3	**25**	18.5	**22.5**	14.5	**23**	1-3-1-1	**20**	*1*	**30**	**30**	**28**
1-1-3-3	**24.5**	16	**25.5**	**23**	**27**	2-2-2-3	**20**	*4*	**30**	**29**	**28**
1-1-3-1	**24**	*5.5*	**29**	**29**	**30**	2-3-1-3	**20**	*4*	**29**	**29**	**28**
1-2-1-3	**24**	*4*	**29**	**29**	**28**	1-2-2-2	19	*3.5*	**22**	**30**	19
1-3-1-3	**23.5**	*5.5*	**30**	**29**	**28**	2-1-1-1	19	15	**19.5**	19	19
2-1-3-1	**23.5**	*8.5*	**30**	**29**	**30**	2-3-1-1	19	*0.5*	**30**	**30**	**28**
2-1-3-3	**23**	13.5	**28**	**23.5**	**27.5**	1-2-1-1	18.5	*0.5*	**30**	**30**	**28**
1-1-2-3	**22.5**	18	**24**	16	**24**	1-3-2-1	18	*1*	**29**	**30**	**27**
1-2-3-3	**22.5**	*2.5*	**30**	**29**	**27**	2-2-1-2	18	*5.5*	**22**	**29**	18.5
1-3-3-3	**22.5**	*3*	**30**	**29**	**25.5**	1-1-2-1	17.5	15	**25**	**23.5**	**22.5**
2-2-3-3	**22.5**	*2.5*	**29**	**29**	**26**	1-3-2-2	17	*3*	**22**	**30**	18.5
1-1-1-3	**22**	**20**	**21**	14	**23.5**	2-3-2-2	16.5	*7*	**23**	**30**	19
1-2-2-3	**22**	*4.5*	**30**	**29**	**28**	1-3-1-2	16	*5*	**23**	**30**	19
1-3-3-1	**22**	*1*	**29**	**29**	**28**	1-1-1-2	15.5	15	9.5	13	12
2-3-2-1	**22**	*1*	**30**	**30**	**28**	1-3-3-2	15.5	*3*	**21**	**29**	19
2-3-2-3	**22**	*5.5*	**29.5**	**29**	**27**	2-2-2-2	15	*6*	**25**	**30**	**21**
2-1-2-1	**21.5**	15	**22**	**23**	**20**	2-3-1-2	15	*5*	**20**	**29**	18
2-2-1-3	**21.5**	*3*	**30**	**29**	**28**	1-2-3-2	14.5	*6.5*	**24**	**29**	18
2-1-1-3	**21**	18.5	**22.5**	13	**22.5**	2-1-3-2	14.5	12	**19.5**	**27**	16
2-2-1-1	**21**	*1*	**29**	**30**	**28**	2-3-3-2	14.5	*5.5*	**23**	**29**	18
2-2-3-1	**21**	*1*	**30**	**29**	**28**	1-2-1-2	14	*5*	**20**	**29**	16
2-3-3-1	**21**	*1*	**30**	**29**	**28**	1-1-3-2	13	12	15.5	**27**	15.5
2-3-3-3	**21**	*2.5*	**29**	**29**	**26**	2-1-2-2	13	14	*9*	14	12
1-2-2-1	**20.5**	*1*	**30**	**30**	**27**	1-1-2-2	12.5	11	12	14	11
1-3-2-3	**20.5**	*4*	**29**	**29**	**27**	2-2-3-2	12	*6*	**23**	**29**	17.5
2-2-2-1	**20.5**	*0.5*	**30**	**30**	**27**	2-1-1-2	11	13	*10*	13	12
1-2-3-1	**20**	*1*	**29**	**29**	**28**						

This is obviously an excellent result for the general strategy of using techniques producing implicit diversity. It may be noted that the only setup with significantly higher base classifier accuracy than the baseline setup is 1-1-1-3, i.e., the only difference is the use of the sparse nets setting. This is a clear indication that even individual sparse nets are able to generalize well.

The direct comparison with standard bagging in Table IX below shows that with larger ensembles, a majority of the setups are at least as good as bagging. In addition, several setups actually obtain significantly higher ensemble accuracy than bagging.

TABLE IX
51 ANN ensembles - Wins against bagging

setup	eAcc	mBAcc	Diff	Dis	DF	setup	eAcc	mBAcc	Diff	Dis	DF
1-3-1-3	**22**	17.5	**22**	16.5	**22**	2-2-1-2	15.5	13.5	11	13	12.5
2-1-2-3	**21**	**27.5**	5	*2*	*8.5*	2-2-2-2	15	13	12.5	15	13
2-2-3-3	**21**	13.5	**23**	**20**	18.5	2-1-2-1	14.5	**29**	*1*	*1*	*2*
1-2-1-3	**20.5**	**20.5**	16.5	12.5	**21.5**	2-2-2-1	14.5	18	16	15	18.5
1-3-3-1	**20**	5.5	**26.5**	**28**	**22**	1-1-1-2	14	**22.5**	*4*	*1*	*8*
2-1-3-3	**20**	**20.5**	7.5	7	*8.5*	1-2-2-2	14	12	*9*	14	11.5
2-3-1-3	**20**	16.5	18	15	18	1-3-3-2	14	11	13.5	23	14
2-3-3-3	**20**	11	**23**	**22.5**	19	2-1-1-3	14	**27.5**	*4*	*2*	6.5
1-2-2-3	19	18.5	19	14	**20.5**	1-1-3-1	13.5	**26**	*2*	*1*	5.5
1-3-2-3	19	15.5	**21**	16	17	2-3-1-1	13.5	13	**25**	**25**	**23**
1-3-3-3	19	12	**27**	**23.5**	19.5	1-1-2-1	13	**29**	*2*	*1*	*2*
1-3-1-1	18	12	**27.5**	**27.5**	**24**	1-2-1-2	13	13	12	16.5	12.5
2-2-1-3	18	18	15.5	14	19	1-3-1-2	13	14	11	16.5	12.5
1-1-3-3	17.5	**21.5**	6	*6*	*8.5*	1-3-2-2	12.5	11.5	13	17	13
1-2-2-1	17.5	14.5	**22**	**21.5**	**20**	2-1-3-2	12.5	**20**	*4*	2	*10*
2-2-2-3	17.5	17.5	17	14	18	2-3-1-2	12.5	13	*10*	15	12.5
2-2-3-1	17.5	*6*	**26**	**28**	**22**	2-3-2-2	12.5	13	11	15	12
2-3-2-3	17.5	16.5	**23**	16	**20**	2-3-3-2	12.5	*8.5*	14	**22**	13.5
1-1-1-3	17	**29**	*4.5*	*1*	*6*	1-1-3-2	12	**20**	*3*	*1*	*7.5*
1-1-2-3	17	**27.5**	*4*	*2*	*8*	1-2-3-2	12	12	16.5	**21.5**	13.5
1-2-3-1	17	*7*	**26.5**	**27**	**24**	2-1-2-2	12	**24**	*3*	*1*	*5.5*
1-2-3-3	17	13.5	**25.5**	18.5	19	2-2-3-2	12	11	15	19.5	13
1-3-2-1	16.5	*8*	**25**	**27**	**20**	2-1-1-1	11.5	**30**	*1*	*0*	*2*
2-3-2-1	16	13.5	**27**	**25.5**	**22.5**	2-1-3-1	*10*	**26.5**	*2*	*1*	*4.5*
2-3-3-1	16	*5*	**27**	**29**	**21**	2-1-1-2	*9*	**20**	*2*	*2*	*5*
2-2-1-1	15.5	**21**	13.5	13.5	17	1-1-2-2	*8*	**20**	*2*	*2*	*4*

Again, the sparse nets setting is the most successful, and it is very interesting to see that it can outperform bagging either by using more accurate, but less diverse ANNs (e.g., 2-1-2-3) or by adding more diversity to bagging (e.g., 2-3-3-3). Moreover, it is also possible to outperform bagging by further increasing the diversity, for instance by using randomized bag sizes with or without varying architectures (e.g., 1-3-1-1 or 1-3-3-1). So, when using more base classifiers, standard bagging is still a strong option, but as demonstrated here, there are a number of ways to obtain even more accurate ensembles, still just targeting diversity implicitly.

Table X below, finally, shows the Spearman's rank correlation coefficients between the different measures.

TABLE X
SPEARMAN'S RANK CORRELATION COEFFICIENTS

	15 ANN ensembles				51 ANN ensembles			
	eAcc	mBAcc	Diff	Dis	eAcc	mBAcc	Diff	Dis
mBAcc	0.26				-0.09			
Diff	0.15	-0.88			0.39	-0.91		
Dis	-0.02	-0.96	0.96		0.28	-0.96	0.98	
Df	0.35	-0.74	0.94	0.85	0.54	-0.81	0.94	0.89

For the smaller ensembles, only the two more informed diversity measures are positively correlated with ensemble accuracy. For the larger ensembles, however, all three diversity measures obtain fairly strong and positive correlations with the ensemble accuracy. At the same time, mean base classifier accuracy is only positively correlated with the ensemble accuracy for the smaller ensembles. Consequently, these results confirm the observation that diversity is relatively more important for the larger ensembles. It is also interesting, but of course expected, to see that mean base classifier accuracy is strongly and negatively correlated with all diversity measures, and that all three diversity measures have strong positive correlations with each other.

V. CONCLUSION

In this paper, we have evaluated several means to produce implicit diversity in ANN ensembles. From the results, it is obvious that although all settings succeeded in producing diversity, the predictive performance of the resulting ensembles varied greatly. Most importantly, a majority of all the setups evaluated clearly outperformed the baseline setup, demonstrating that the increase in diversity produced by the different methods was generally beneficial for ensemble accuracy. Especially when using larger ensembles, a majority of the setups evaluated produced significantly more accurate ensembles than the baseline setup. In addition, several setups also outperformed standard bagging. As a matter of fact, for the larger ensembles, a majority of the setups were at least as good as bagging, and a number of setups even obtained significantly higher ensemble accuracy than bagging. The levels of increased diversity produced by the methods evaluated in this study normally resulted in increased ensemble accuracy, i.e., diversity was produced without lowering the base classifier accuracy too such extent that the ensemble was weakened. The overall conclusion is thus that the produced diversity more than compensated for the decrease in base classifier accuracy. The analysis also showed that diversity is actually more important for the larger ensembles. Comparing, finally, the individual settings, the novel way of producing diversity by using ANN base classifiers with different and slightly randomized link structures, was generally the most successful, while using randomized feature sets resulted in the least accurate ensembles.

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*, ser. Lecture Notes in Computer Science, J. Kittler and F. Roli, Eds., vol. 1857. Springer, 2000, pp. 1–15.

[2] D. W. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research (JAIR)*, vol. 11, pp. 169–198, 1999.

[3] T. G. Dietterich, "Machine-learning research: Four current directions," *The AI Magazine*, vol. 18, no. 4, pp. 97–136, 1998.

[4] G. Brown, J. Wyatt, R. Harris, and X. Yao, "Diversity creation methods: a survey and categorisation," *Journal of Information Fusion*, vol. 6, no. 1, pp. 5–20, 2005.

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

[6] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, 1998.

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, October 2001.

[8] E. Tang, P. Suganthan, and X. Yao, "An Analysis of Diversity Measures," *Machine Learning*, vol. vol 65, pp. 247–271, 2006.

[9] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," *Advances in Neural Information Processing Systems*, vol. 2, pp. 231–238, 1995.

[10] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine*, vol. 12, no. 10, pp. 993–1001, 1990.

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

[12] M. M. Islam, X. Yao, Shahriar, M. A. Islam, and K. Murase, "Bagging and Boosting Negatively Correlated Neural Networks," *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, vol. 38, no. 3, pp. 771–784, 2008.

[13] G. Brown, *Diversity in Neural Network Ensembles. PhD thesis.* University of Birmingham, 2004.

[14] G. Giacinto and F. Roli, "Design of effective neural network ensembles for image classification purposes," *Image and Vision Computing*, vol. 19, no. 9-10, pp. 699–707, 2001.

[15] L. I. Kuncheva and C. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine Learning*, vol. 51, no. 2, pp. 181–207, 2003.

[16] L. Saitta, "Hypothesis Diversity in Ensemble Classification," in *Foundations of Intelligent Systems*. Springer, 2006, pp. 662–670.

[17] Y. Freund and R. Schapire, "Experiments with a new boosting algorithm," in *Proc. of the International Conference on Machine Learning*, 1996, pp. 148–156.

[18] R. Maclin and J. W. Shavlik, "Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks," in *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, 1995, pp. 524–530.

[19] K. Cherkauer, "Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks," in *Working Notes of the AAAI Workshop on Integrating Multiple Learned Models*, 1996, pp. 15–21.

[20] N. C. Oza and K. Tumer, "Input decimation ensembles: Decorrelation through dimensionality reduction," in *LNCS*. Springer, 2001, pp. 238–247.

[21] U. Johansson, T. Löfström, and L. Niklasson, "Evaluating standard techniques for implicit diversity," in *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2008, pp. 592–599.

[22] P. Melville and R. J. Mooney, "Creating diversity in ensembles using artificial data," *Information Fusion*, vol. 6, pp. 99–111, 2004.

[23] Y. Raviv and N. Intrator, "Bootstrapping with noise: An effective regularization technique," *Connection Science*, vol. 8, pp. 355–372, 1996.

[24] D. W. Opitz, J. W. Shavlik, and O. Shavlik, "Actively searching for an effective neural-network ensemble," *Connection Science*, vol. 8, pp. 337–353, 1996.

[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml