Int. J. Mach. Learn. & Cyber. (2013) 4:301–317
DOI 10.1007/s13042-012-0094-8

ORIGINAL ARTICLE

Performance of global–local hybrid ensemble versus boosting and bagging ensembles

Dustin Baumgartner · Gursel Serpen

Received: 5 March 2012 / Accepted: 10 April 2012 / Published online: 25 April 2012

Abstract This study compares the classification performance of a hybrid ensemble, which is called the global–local hybrid ensemble that employs both local and global learners against data manipulation ensembles including bagging and boosting variants. A comprehensive simulation study is performed on 46 UCI machine learning repository data sets using prediction accuracy and SAR performance metrics and along with rigorous statistical significance tests. Simulation results for comparison of classification performances indicate that global–local hybrid ensemble outperforms or ties with bagging and boosting ensemble variants in all cases. This suggests that the global–local ensemble has a more robust performance profile since its performance is less sensitive to variation with respect to the problem domain, or equivalently the data sets. This performance robustness is realized at the expense of increased complexity of the global–local ensemble since at least two types of learners, e.g. one global and another one local, must be trained. A complementary diversity analysis of global–local hybrid ensemble and base learners used for bagging and boosting ensembles on select data sets in the classifier projection space provides both an explanation and support for the performance related findings of this study.

Keywords Hybrid classification ensemble · Global–local learning · Heterogeneous–homogeneous diversity · Boosting · Bagging · SAR metric · Statistical testing · Classifier projection space

D. Baumgartner · G. Serpen (✉)
Electrical Engineering and Computer Science Department,
University of Toledo, Toledo, USA
e-mail: gserpen@eng.utoledo.edu

1 Introduction

The global–local hybrid ensemble (GLHE) is a classifier ensemble design that is characterized by two main traits [4]. First, one global learner and one local learner are explicitly used. Second, both heterogeneous and homogeneous diversities are integrated. Mitchell presents an explanation and comparison of how global and local learners work [26]. When all training instances are considered during classification of a query instance, the learner is termed as *global*. When only near training instances are considered during classification of a query instance, the learner is called *local*. Global learners estimate a single target function for the entire instance space, while local learners estimate target functions locally and differently for each query instance [20].

There are advantages and disadvantages for each type of learning algorithm, and their ability for generalization depends on the problem at hand. Generally, global learners do not respond well to isolated data points—those points in a sparsely distributed area. That is, it attempts to have a model that satisfies the majority of points while paying little attention to outliers (similar to how linear regression works). Local learners, alternatively, are better for handling the isolated points since their generalization is instance-based. However, if the target function only depends on a few of the many available attributes, then the instances that are most "similar" may actually be a large distance away [27]. One can argue that these two types of learners may behave in a "complementary" way: when one fails, the other may succeed since it views the problem in such a different manner.

Co-existence of a global learner along with a local learner within the same ensemble framework may offer a powerful mixture of diversity. While traditional ensemble designs in the literature appear to use only one type of diversity, either heterogeneous or homogeneous [7], global–local hybrid ensemble employs both forms. The heterogeneous diversity originates from the use of a global learner and a local learner, while the homogeneous diversity is due to multiple instantiations of each learner with different initial parameterizations (e.g. initial weights for neural networks, number of neighbors for kNN, pruning techniques for decision trees, etc.). The combination of the two diversities is not common in the literature. Although some have experimented with it indirectly [41], others have trivialized the combination to being nothing more than a heterogeneous ensemble in essence [7].

The generic architecture of the GLHE design is shown in Fig. 1. The contribution of GLHE is the design or composition of the base classifiers. There are *n* base-classifiers from a global learner with different parameterizations. Likewise, there are *m* base-classifiers from a local learner with different parameterizations. The global and local learning algorithms provide the main source of diversity (heterogeneity), while instantiation of multiple classifiers from each learner (homogeneity) gives the ensemble an opportunity to benefit from the better performing learner numerous times rather than just once.

A previous GLHE study [4] compared prediction accuracy of the proposed design against those of 47 ensembles reported in six prominent studies in the literature [5, 17, 22, 27, 32, 35]. The ensembles varied from bagging and boosting to hybrid ensembles with up to seven different learning algorithms. It was found that the performance of GLHE is not statistically different when compared to the performances of 45 other ensembles, statistically better than that of C4.5 bagging ensemble, and statistically worse than that of one hybrid ensemble. The results were supportive of the hypothesis that GLHE offers the same robust performance as other, at times more complex, ensemble designs. Although a large number of comparisons were performed and since comparisons were made with studies already reported in the literature, the study was limited by the constraints imposed by those same literature studies: comparisons were made with ensembles designed by others, only the prediction accuracy measurements were available in the literature studies, and at most 27 datasets were considered by any single study. Consequently, conclusions of the Baumgartner and Serpen study [4] facilitated making only general statements about GLHE. This study aims to provide a more precise projection of the utility of the GLHE design through a comparative performance evaluation with data manipulation ensembles, namely boosting and bagging classifier ensembles. Although not directly studied in this work, the GLHE method may be similarly applicable to relatively new problem domains that traditional data manipulation ensembles have been extended to, including adversarial environments [6].

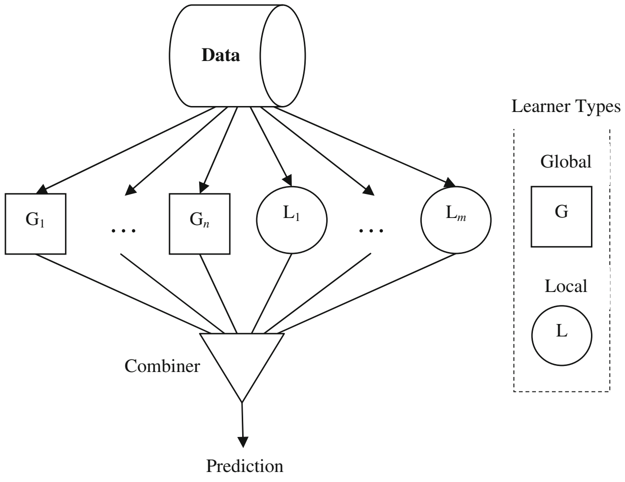


Fig. 1 Generic architecture of global–local hybrid ensemble (GLHE) design

In data manipulation ensembles, the approach to building an ensemble uses a single learning algorithm with different subsets of the original dataset to train different base classifiers. Multiple classifiers are generated from a single learning algorithm through variations of the training data (e.g. different samples of instances and/or different samples of features). Data manipulation ensembles are ideal when a single learning algorithm is known to perform well for a given dataset, as the performance will likely improve. However, there is also risk that the learning algorithm may not perform well for a given dataset, which adversely affects the performance of the ensemble. Regardless of its drawbacks, the simplicity of this method makes it the most widely investigated diversity creation method [30].

Popular data manipulation ensemble techniques are bagging and boosting (AdaBoost). Breiman's bagging, or bootstrap aggregation, uses different instance subsets of the training dataset with a single learning algorithm [8]. Generally, the subset size is near the size of the original dataset; however, random sampling with replacement creates subsets with duplicates and/or omissions of the original instances. The same learning algorithm is used to train on the different subsets of data, each training episode or case resulting in a new base classifier. Given a new instance, the classifier predictions are aggregated with a majority vote to derive the final prediction for the ensemble. Whereas bagging relies on randomness to provide better performance from an ensemble, Boosting takes a more active role. Freund and Schapire's popular boosting variant, AdaBoost [18], explicitly alters the distribution of the training dataset to concentrate on the instances that have not been correctly learned in the previous iterations (i.e.

consecutive training datasets target hard-to-classify instances). A weighted vote of each classifier's prediction is performed to obtain the ensemble's final prediction. A recent approach, bagging–AdaBoost, incorporates components of each classic data manipulation ensembles into a single technique to improve the accuracy, stability, and robustness of the ensemble [42].

In the subsequent sections a comparative simulation-based performance and diversity analysis of global–local hybrid ensemble with boosting and bagging ensembles will be presented. It is of interest to determine if the global–local hybrid ensemble offers a more robust performance as a consequence of its heterogeneous diversity when compared to data manipulation ensembles, while also recognizing that the global–local hybrid ensemble projects a higher level of training cost due to the need to train both global and local learners.

2 Simulation study

The simulation study compares the performance of GLHE against data manipulation ensembles, namely bagging and boosting variants. The study employs 46 datasets from the UCI Machine Learning Repository [2] to profile the performance of global–local hybrid ensemble against boosting and bagging ensembles on the basis of prediction accuracy and SAR metrics. A rigorous statistical significance testing is applied for the performance comparisons.

All simulations are executed using the open source Weka software (version 3.5.8) using tenfold cross validation [39], with the large experiment and evaluation tool (LEET) as a front-end [3]. The prediction accuracy estimate for a classifier on a given dataset is an average of the ten samples (10 folds). The variation of these samples is not important for this study, as will be made apparent in the discussion of the statistical significance testing method in a forthcoming section. The 46 publically available datasets from the UCI Machine Learning Repository are listed, with their characteristics, in Table 1. This collection of datasets was obtained as a union of datasets used in six popular studies in the literature that target performance robustness [5, 17, 22, 27, 32, 35]. Only those that could not be found (due to naming discrepancies) and the letter dataset (due to its relatively long required run-times) are not included. Since exclusion of datasets is not dependent on a bias for/against the GLHE design, little or no impact is anticipated on the results and conclusions of this study.

2.1 Statistical significance testing

Demsar [14] argues that comparing classifiers does not satisfy conditions for parametric tests, and instead proposes the use of nonparametric alternatives. Some have applied these tests to their studies [4, 5, 25], and we believe it is a sound approach to follow. Since nonparametric tests rank the classifiers for each dataset, the important source of variations is the (independent) datasets and not the (usually dependent) samples used to calculate the accuracy. This implies that, besides obtaining a precise estimate of accuracy, the sampling method is irrelevant because one does not have to worry about the Type I error generated from it. When comparing multiple classifiers, basic statistics dictates that a certain portion of the reported statistical significance is actually due to random chance [14, 28, 35]. Demsar's suggested procedure of tests resolves this issue.

In this study, first the Friedman test is performed, which is based on the average rank of each classifier across the datasets [19]. Iman and Davenport [21] showed that the Friedman test is undesirably conservative, and derived a new statistic based on its value. This is then used to test the null hypothesis—that all classifiers have equivalent performance. If the null hypothesis is accepted, then there is no statistically significant difference in the classifier performances. Otherwise, the post hoc Bonferroni–Dunn test is conducted, in which a control classifier is compared to all others in the group [16]. The performance of the control and another classifier is significantly different if the corresponding average ranks differ by at least the critical difference (CD).

The results of the statistical significance tests are presented graphically. An example is shown in Fig. 2. The left-most bar is the control classifier used for the Bonferroni–Dunn test, while the remaining bars are the comparison classifiers. The height of each bar represents the average rank of the associated classifier, which indicates relative performance among the classifiers (higher ranks are worse performing than lower ranks). Although examining the average ranks may offer some insight, the more strict and valid comparison involves the statistical significance thresholds resulting from the Bonferroni–Dunn test. Any classifier with an average rank above the high threshold is statistically worse than the control. Alternatively, any classifier with an average rank below the low threshold is statistically better than the control. Those classifiers with ranks within the threshold lines have a performance that is statistically no different than that of the control. Note that the exact values of the thresholds are displayed at the bottom of the graph. This series of statistical significance tests and the graphical presentation are applicable to any measure of performance for multiple classifiers. This includes the prediction accuracy and the SAR metrics as described in the next section.

2.2 Performance metrics

The simulation study employs two performance metrics, namely prediction accuracy and SAR where the latter is an

Table 1 Simulation study datasets and their characteristics

Dataset	Number instances	Number classes	# Binary attributes	# Nominal attributes	# Numeric attributes	% Majority class	% Minority class
Anneal	898	6	19	13	6	76	1
Audiology	226	24	61	8	0	25	0
Autos	205	7	4	6	15	33	1
Balance-scale	625	3	0	0	4	46	8
Breast-cancer	286	2	3	6	0	70	30
Breast-w	699	2	0	0	9	66	34
Car	1,726	4	0	6	0	70	4
cmc	1,473	3	3	4	2	43	23
Colic	367	2	4	16	7	58	1
Credit-a	690	2	4	5	6	56	44
Credit-g	1,000	2	2	11	7	70	30
Diabetes	768	2	0	0	8	65	35
Echocardiogram	74	2	2	0	7	68	32
Glass	214	7	0	0	9	36	4
Haberman	306	2	0	1	2	74	26
Heart-c	303	5	3	4	6	54	46
Heart-h	294	5	3	4	6	64	36
Heart-statlog	270	2	0	0	13	56	44
Hepatitis	155	2	13	0	6	79	21
Hypothyroid	3,772	4	20	1	7	92	0
Ionosphere	351	2	0	0	34	64	36
Iris	150	3	0	0	4	33	33
kr-vs-kp	3,196	2	34	2	0	52	48
Labor	57	2	3	5	8	65	35
Lymphography	148	4	9	6	3	55	1
Monk2	599	2	2	4	0	66	34
Mushroom	8,124	2	4	18	0	52	48
Page-blocks	5,471	5	0	0	10	90	1
Pendigits	10,990	10	0	0	16	10	10
Satellite	6,435	7	0	0	36	24	10
Segment	2,310	7	0	0	19	14	14
Sick	3,772	2	20	1	7	94	6
Solar-flare-c	1,389	8	5	5	0	84	0
Solar-flare-m	1,389	6	5	5	0	95	0
Solar-flare-x	1,389	3	5	5	0	99	0
Sonar	208	2	0	0	60	53	47
Soybean	683	19	16	19	0	13	1
Splice	3,190	3	0	61	0	52	24
Tic-tac-toe	956	2	0	9	0	65	35
Tumor	339	22	14	3	0	25	0
Vehicle	846	4	0	0	18	26	24
Vote	435	2	16	0	0	61	39
Vowel	990	11	2	1	10	9	9
Waveform	5,000	3	0	0	40	34	33
Wine	176	3	0	0	13	40	26
Zoo	101	7	15	1	1	41	4

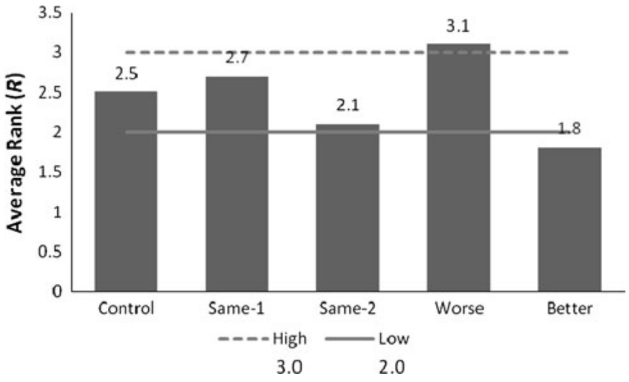


Fig. 2 Example results of the statistical significance tests

aggregate measure based on prediction accuracy, area under the receiver operating curve and root-mean-squared-error.

Prediction accuracy or ACC in short is defined as the proportion of the number of correct predictions that the classifier makes relative to the total number of instances. That is,

$$ACC = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Instances},$$

where larger prediction accuracies indicate better performance.

The receiver operating characteristic (ROC) is a 2-dimensional plot of true positives on the vertical axis against false positives on the horizontal axis. The process to construct the graph is as follows [39]. First, the predictions are sorted by descending probability, without regard to whether or not the prediction is correct. Then the graph is drawn by traversing through the predictions, from most probable to least probable. If the prediction is correct, then move one unit up. Alternately, if the prediction is incorrect, then move one unit to the right. The area under the ROC curve, ROCA, is used as a summary statistic. Larger areas are considered as having better performance. Although it is already in use and accepted in fields such as medicine, ROCA is gaining popularity in the wider machine learning community. For a 2-class problem, a single ROCA value is calculated between the two classes. For a *c*-class problem, there are *c* ROCA values calculated such that a given class is considered against the set of the remaining classes (i.e. a class is considered as 1 and all remaining classes are considered as 0). These values are then weighted by the number of instances from each corresponding class and their sum is divided by the total number of instances. Although calculation of ROCA for multiclass problems is not frequent, the stated method is accepted in the literature [31].

Root-mean-squared-error (RMSE) is widely used in regression problems; however, it can also be adapted for classification problems. RMSE measures how much predictions deviate from the true values. For C classes, let $p_1, p_2, \ldots, p_C$ be the probability values of the predicted classes (such that p_i is the probability that class i is predicted with respect to the total number of instances, and all the probabilities sum to 1), and $a_1, a_2, \ldots, a_C$ be the probability values of the actual classes (such that the true class is 1 and the others are 0). RMSE is defined as [39]:

$$RMSE = \sqrt{\frac{1}{N}\sum\nolimits_{z=1}^{N}\sum\nolimits_{j=1}^{C}\left[(p_j - a_j)^2/C\right]},$$

where N is the number of instances. Smaller RMSE values indicate better performance.

SAR is an aggregate performance metric that has been shown to have better correlation with ten other prominent performance metrics than any component metric alone [10, 11]. It is an average of the prediction accuracy, area under the ROC curve, and root-mean-squared-error. That is,

$$SAR = \frac{(ACC + ROCA + (1 - RMSE))}{3},$$

and a higher value of SAR indicates better performance.

2.3 Ensemble designs

The global–local hybrid ensemble (GLHE) is designed using a decision tree classifier as its global learner and a nearest neighbor classifier as its local learner. Specifically, the global learner is J48 (a port of the C4.5 decision tree classifier) and the local learner is IBk (an instance-based nearest-neighbor classifier [1]). J48 is a re-implementation of C4.5 release 8 (which develops a decision tree based classifier from a set of training data through information entropy measure) in Java for Weka and employs both C4.5's confidence-based post-pruning (default) and subtree raising.

The categorization of J48 as global and IBk as local is in conformity with multiple references in the literature [26, 33]. There are obviously many different learning algorithms that could be used to satisfy the global–local learner requirement of GLHE, some of which may produce better performance (e.g. multi-layer perceptrons for global learning and radial basis functions for local learning). However, initial exploratory in-house assessments indicated that the simplicity and efficiency of J48 and IBk would allow for a large-scale experimentation study while capturing the important factors of the GLHE design.

Although GLHE is a generic ensemble design, our choice of relatively simple learning algorithms (e.g. C4.5 and IBk) over complex ones (e.g. artificial neural networks, Bayesian nets or support vector machines) for empirical evaluation further supports a pursuance of simplicity. This ad-hoc approach could be improved with in-depth testing and analysis of diversity. Both the selection of learning algorithms and the set of classifiers for each promotes reduced complexity in the ensemble design—an important goal of classifier design in general.

The heterogeneous diversity of the GLHE design comes from the use of two different types of base learning algorithms—one global and one local. The homogenous diversity of GLHE is achieved by varying the parameter settings for each learner to create multiple classifier instances. This requires three steps. First, the target number of classifiers from each learner is set. In this study, three classifiers are used from each learner. This sufficiently allows homogeneous tendencies to be incorporated while keeping the total number of base classifiers at a manageable level. Second, the base parameter value settings for each learner is established. For J48, the base parameters are Weka's default. For IBk, the base parameters are Weka's default with the exception of distance weighting which has a value of "1/distance". Third, for each learner different parameter value settings are employed to create the multiple base classifier instances. For J48, the type of pruning is changed: these are standard pruning, reduced error pruning, and no pruning (unpruned). For IBk, the number of nearest-neighbors is changed: the employed values are 1, 5, and 10.

The next step in the design of GLHE is determining the ensemble combination method to be used. Three ensemble techniques or architectures were considered as follows:

- Voting—a simple combination scheme of the base-classifier predictions to derive the final ensemble prediction. In this study, the average of probabilities combination rule is used [15].
- StackingC [36]—stacking uses meta-classification of a dataset created with the prediction values of the base-classifiers as features and the original values as the class [40]. The meta-classifier derives a final prediction from this. StackingC is more efficient with a reduced number of features in the created dataset. In this study, linear regression is used as the meta-classifier.
- Grading [35]—another meta-classification approach that employs an opposite approach to Stacking. A new dataset is created for each base-classifier, with instances containing the original features, but the class indicates whether the base-classifier's prediction was correct. The meta-classifier predicts which base-classifier will be correct. In this study, IBk (an instance-based K-nearest-neighbor classifier) with ten nearest-neighbors is used as the meta-classifier.

A comparison of the prediction accuracies of GLHE for Voting, StackingC, and Grading techniques was performed on the 46 datasets listed in Table 1. The average ranks and statistical significance thresholds for this comparison are shown in the graph of Fig. 3. Since StackingC has the lowest average rank, it is used as the control for the Bonferroni–Dunn test. Voting and Grading are right at the statistical significance threshold, so it is safe to conclude that StackingC has the best prediction accuracy performance of the three GLHE ensembles. Therefore, it is used for comparison with the data manipulation ensembles.

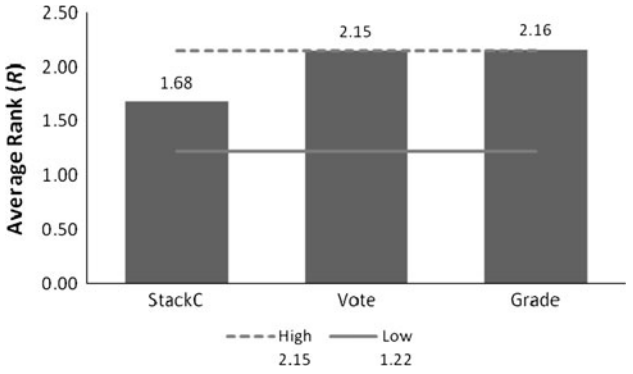


Fig. 3 Prediction accuracy comparison of the GLHE ensembles

The design of data manipulation ensembles was based on two ensemble techniques, namely bagging and AdaBoost, and six learning algorithms. The base learning algorithms and their parameterizations used (default WEKA values unless otherwise stated) are as follows:

- J48—a port of the C4.5 decision tree classifier.
- IBk—an instance-based nearest-neighbor classifier (distance weighting of "1/distance" and 5 nearest-neighbors).
- NB—naïve Bayes classifier (supervised discretization).
- PART—creates rules from partial C4.5 trees.
- KStar—an instance-based entropy-based classifier.
- SMO—sequential minimal optimization algorithm for training a support vector machine classifier.

Six bagging and AdaBoost (AdaBoost.M1 in WEKA) ensembles are produced from the six learning algorithms, namely J48, IBk, NB, PART, KStar, and SMO. The labels "Bag-" and "Boost-" are pre-pended to the learning algorithm names (e.g. bagging of J48 is called Bag-J48). Ten iterations are performed for each ensemble except for those with SMO on the "splice" dataset, for which only five iterations could be performed due to memory limitations of Weka.

2.4 Simulation results: GLHE versus non-ensemble classifiers

This phase of the simulation study entailed comparing prediction accuracy performances of GLHE and individual (non-ensemble) classifiers based on J48, IBk, NB, PART, KStar, and SMO for all 46 UCI Machine Learning Repository datasets presented in Table 1. Simulation results are summarized in Fig. 4, where GLHE has all six base learners and the StackingC as the meta learner, and detailed for prediction accuracy values in Table 2. The Type I error for all tests is $\alpha = 0.05$. The null hypothesis of Iman–Davenport test is rejected, indicating there is statistically significant differences among the performances of the classifiers. The post hoc Bonferroni–Dunn test is then conducted for each comparison with the GLHE as the control, resulting in the proposed ensemble design being statistically better performing than all except SMO. The GLHE scored the first place for 19 of the 46 datasets, second place for 7 datasets, third place for 11 datasets, fourth place for 3 datasets, fifth place for 4 datasets, sixth place for 1 dataset, and never came in the seventh place for any of the datasets. These results clearly indicate that the prediction accuracy performance of GLHE tends to be leading (e.g. claimed the top 3 places for 37 out of 46 datasets) and shows much less variation with respect to dataset or problem domain variation, and hence suggesting a more robust performance profile compared to any non-ensemble classifier listed in Table 2. These results also respond to the expectation that an ensemble should perform better when compared to non-ensemble classifiers.

2.5 Simulation results: GLHE versus data manipulation ensembles

Simulation data for each bagging and boosting ensemble classifier evaluated are presented in terms of prediction accuracy and SAR values in Tables 5, 6, 7 and 8 in the appendix. The design of these ensembles was as presented earlier in Sect. 2.3. The same data is however summarized herein per the requirements of the statistical significance tests performed for comparison purposes. Accordingly, the results are presented in comparison groups. The Type I error for all tests is $\alpha = 0.05$. For all comparisons conducted, the null hypothesis of the Iman–Davenport test has been rejected, so the classifiers within each group do not perform equivalently. The post hoc Bonferroni–Dunn test is then conducted for each comparison with the GLHE design as the control (unless otherwise stated).

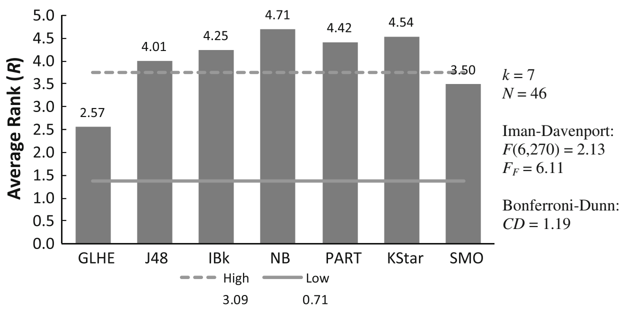


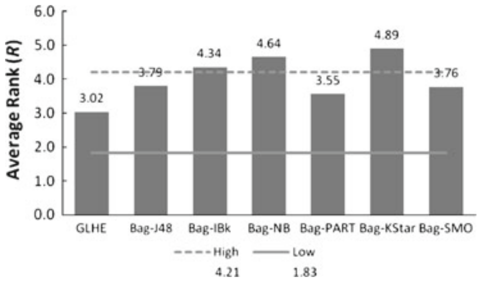
Fig. 4 Average ranks for prediction accuracy for the StackingC GLHE and its base classifiers. CD thresholds are for the ensemble

The average ranks and statistical significance tests for GLHE, bagging, and AdaBoost ensembles for the prediction accuracy metric are shown in Figs. 5 and 6, respectively, where k represents the number of ensembles evaluated and N is the number of data sets. For both data manipulation ensemble types, GLHE has the lowest average rank, while its performance is statistically better than those of bagging and boosting ensembles with IBk, NB, and KStar as the learning algorithms. Performance results based on the SAR metric, on the other hand, projects quite a different perspective. Figures 7 and 8 give the average ranks of and statistical test results for the bagging and AdaBoost ensembles, respectively, for the SAR metric. GLHE is only statistically better than KStar and SMO bagging ensembles. Bag-PART has a slightly lower rank (better performance) than GLHE. In Fig. 8, GLHE has the lowest average rank, with AdaBoost IBk, NB, KStar, and SMO ensembles performing statistically worse.

Results of the statistical significance tests for comparison between GLHE and data manipulation ensembles are summarized in Table 3. Each entry corresponds to a comparison between GLHE StackingC and a data manipulation ensemble, specified by the column label that represents the ensemble technique and the row label that shows the base learning algorithm used. For example, with regards to prediction accuracy, GLHE is statistically the same as bagging with PART, and better than bagging with IBk. Comparison to data manipulation ensembles shows that the GLHE design's prediction accuracy is statistically better than those of three of six bagging ensembles. Considering the same metric, GLHE performs better than three AdaBoost versions (namely IBk, NB and KStar as learners) while scores the same with the rest. For the SAR metric, GLHE scores the same with four versions of bagging while surpassing the performance for the other two. The GLHE outperforms four AdaBoost versions and ties with the other two. In none of the cases, GLHE lags the performance of any bagging or boosting variant. The results in Table 3 indicate that GLHE exhibits a more robust performance, in the sense that it maintains consistently its competitive classification performance over a large set of problem domains or data sets, compared to the data manipulation ensembles—especially those with IBk, NB, KStar and SMO as their base learning algorithms. This finding establishes GLHE as desirable over data manipulation ensembles for the situations where a consistently high-performing classifier is needed to operate for a large set of problem domains.

Table 2 Prediction accuracy values for GLHE and six other non-ensemble classifiers

	Single learner classifiers (non-ensemble)						GLHE
	J48	IBk	NB	PART	KStar	SMO	StackingC
Anneal	98.44	97.88	96.44	98.22	95.77	97.44	98.55
Audiology	77.88	67.70	73.45	78.32	79.20	81.86	77.43
Autos	81.95	75.12	64.88	77.56	73.17	71.22	84.39
Balance-scale	76.64	88.00	72.32	83.52	88.48	87.68	90.88
Breast-cancer	75.52	72.73	71.68	71.33	73.43	69.58	72.03
Breast-w	94.56	97.00	97.14	93.85	95.42	97.00	96.57
Car	92.70	93.22	85.57	95.60	87.43	93.45	94.44
cmc	52.14	45.76	51.05	49.15	50.24	48.20	50.44
Colic	82.88	79.35	76.36	80.43	69.57	79.89	84.24
Credit-a	86.09	85.94	86.52	85.36	78.99	84.93	86.38
Credit-g	70.50	74.20	76.00	70.20	69.40	75.10	74.50
Diabetes	73.83	73.18	74.35	75.26	69.14	77.34	74.61
Echocardiogram	95.95	93.24	97.30	95.95	91.89	93.24	98.65
Glass	66.82	71.96	70.56	68.22	75.23	56.07	68.69
Haberman	72.88	69.28	72.55	69.61	74.18	73.53	73.53
Heart-c	77.56	81.19	83.83	79.87	74.59	84.16	81.19
Heart-h	80.95	81.97	84.01	80.95	77.89	82.65	80.27
Heart-statlog	76.67	78.52	81.11	73.33	75.19	84.07	78.52
Hepatitis	83.87	85.16	83.23	84.52	81.94	85.16	83.23
Hypothyroid	99.58	93.40	98.22	99.42	94.67	93.61	99.58
Ionosphere	91.45	84.90	89.17	91.74	84.62	88.60	91.74
Iris	96.00	95.33	92.67	94.00	94.67	96.00	96.00
kr-vs-kp	99.44	96.31	87.89	99.06	97.03	95.43	99.44
Labor	73.68	85.96	85.96	78.95	89.47	89.47	80.70
Lymphography	77.03	85.14	84.46	76.35	85.14	86.49	82.43
Monk2	63.11	79.13	62.77	79.80	83.31	65.61	82.30
mushroom	100.00	100.00	95.83	100.00	100.00	100.00	100.00
Page-blocks	97.04	96.14	93.38	97.11	97.00	92.89	97.42
Pendigits	96.53	99.30	87.58	96.83	99.24	97.91	99.31
Satellite	85.83	90.68	81.63	86.98	90.69	86.88	91.38
Segment	96.93	96.15	91.30	96.23	97.06	93.07	97.58
Sick	98.81	96.37	97.16	98.62	95.92	93.85	98.81
Solar-flare-c	84.31	82.00	78.83	83.59	83.66	84.31	84.16
Solar-flare-m	95.10	93.59	90.78	94.38	94.38	95.10	95.03
Solar-flare-x	99.14	98.92	96.04	98.99	98.92	99.14	99.06
Sonar	71.15	85.58	80.29	80.29	84.62	75.96	86.06
Soybean	91.51	90.34	92.97	91.95	87.99	93.85	92.68
Splice	94.08	82.13	95.30	92.73	79.03	93.45	94.58
Tic-tac-toe	85.25	99.27	69.77	95.08	95.92	98.54	99.27
Tumor	39.82	44.84	50.15	40.71	37.76	46.90	45.43
Vehicle	72.46	73.29	60.05	71.51	71.39	74.35	73.05
Vote	96.32	92.64	90.11	94.71	93.33	96.09	96.32
Vowel	81.52	96.77	60.10	76.67	98.99	71.41	99.29
Waveform	75.08	78.94	79.86	77.42	73.48	86.68	82.24
Wine	93.75	94.89	97.16	92.61	98.30	98.86	96.02
Zoo	92.08	95.05	93.07	92.08	96.04	96.04	95.05



$k = 7$

$N = 46$

Iman-Davenport:

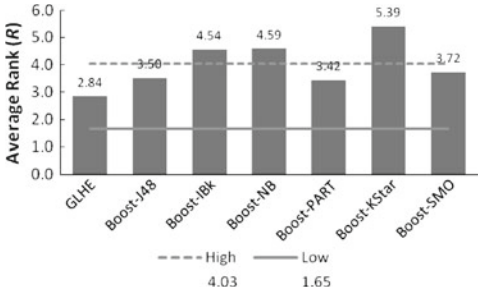
$F(6,270) = 2.13$

$F_F = 4.56$

Bonferroni-Dunn:

$CD = 1.19$

Fig. 5 Average ranks of accuracy for StackingC GLHE and bagging ensembles. CD thresholds for GLHE



$k = 7$

$N = 46$

Iman-Davenport:

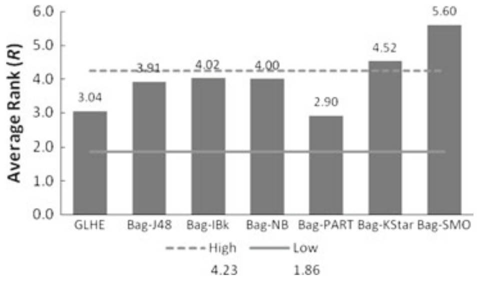
$F(6,270) = 2.13$

$F_F = 8.82$

Bonferroni-Dunn:

$CD = 1.19$

Fig. 6 Average ranks of accuracy for StackingC GLHE and AdaBoost ensembles. CD thresholds for GLHE



$k = 7$

$N = 46$

Iman-Davenport:

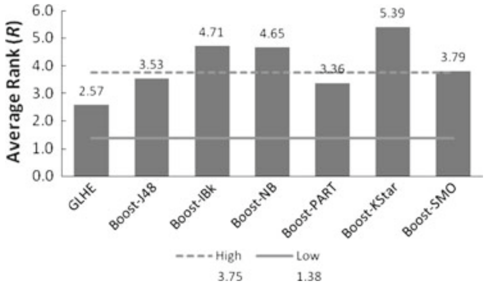
$F(6,270) = 2.13$

$F_F = 9.67$

Bonferroni-Dunn:

$CD = 1.19$

Fig. 7 Average ranks of SAR for StackingC GLHE and bagging ensembles. CD thresholds for GLHE



$k = 7$

$N = 46$

Iman-Davenport:

$F(6,270) = 2.13$

$F_F = 11.23$

Bonferroni-Dunn:

$CD = 1.19$

Fig. 8 Average ranks of SAR for StackingC GLHE and AdaBoost ensembles. CD thresholds for GLHE

3 Diversity analysis

In this section, a diversity analysis will be performed to explore the relationship between ensemble performance and ensemble diversity in light of the simulation results presented in the previous section.

Table 3 Summary of statistical significance tests for GLHE StackingC relative to data manipulation ensembles

Learning algorithms	Prediction accuracy		SAR	
	Bagging	AdaBoost	Bagging	AdaBoost
J48	=	=	=	=
IBk	>	>	=	>
NB	>	>	=	>
PART	=	=	=	=
KStar	>	>	>	>
SMO	=	=	>	>

The cells indicate how well GLHE performs in the comparison. The '>', '<', and '=' symbols indicate that GLHE has better, worse, or equal average rank, respectively

3.1 Diversity creation methods

There are three ways in which diversity can be created among the base classifiers of an ensemble [9, 12].

- Data manipulation—multiple classifiers are generated from a single learning algorithm through variations of the training data (e.g. different samples of instances and/or different samples of features).
- Homogeneous—multiple classifiers are generated from a single learning algorithm through variations of the parameters (e.g. neural networks with different initial weight values). As with data manipulation ensembles, the performance of homogeneous ensembles may suffer from being limited to a single learning algorithm.
- Heterogeneous (hybrid)—multiple classifiers are generated from two or more learning algorithms (i.e. C4.5 and naïve Bayes). Consensus in the literature is that heterogeneous ensembles are more effective at producing diversity, and consequently have more robust prediction accuracy performance [7]. Certain learning algorithms may be experts in an instance space, while others are possibly inexpert. Multiple learning algorithms may help to protect the ensemble from being burdened by poor performance of any single one. However, heterogeneous ensembles pose higher level difficulty compared to data manipulation or homogenous ensembles due the apparent need to train, test, validate and deploy multiple learning algorithms.

3.2 Measuring diversity: pairwise measures

Pairwise measures consider the classifier population definition of diversity and evaluate the diversity between two classifiers at a time. The pairwise diversity measure of interest is the disagreement measure (*Dis*). The disagreement measure was created specifically for characterizing the diversity between two classifiers. It counts the number of times that one classifier was correct and the other incorrect—an intuitive concept of diversity [37]. For two classifiers, D_i and D_j, the disagreement measure is defined as

$$Dis_{i,j} = \frac{N^{10} + N^{01}}{N}.$$

The value of *Dis* ranges between 0 and 1, where 0 indicates no difference and 1 indicates the highest possible diversity.

Since pairwise measures consider only two classifiers at a time, an ensemble of k classifiers produces $k(k-1)/2$ pairwise diversity values. To get a single value, the average across all pairs is taken (i.e. divide the sum of pairwise measures by the number of pairwise diversity values). The average *Dis* measure is referred to as Dis_{av}.

3.3 Visualizing diversity: classifier projection space

Pekalska et al. [29] propose the classifier projection space (CPS) as a 2-dimensional representation of classifiers such that the points correspond to classifiers and the relative pairwise diversities are preserved by the Euclidian distances between the points. They suggest that this method is more appropriate in situations of an ensemble containing both similar and diverse base-classifiers, since in such a case the values may simply average out.

Given k classifiers, a $k \times k$ dissimilarity matrix N is created such that the value of each entry in the matrix is the pairwise measure of diversity between classifiers associated with the row and column labels for that entry. An illustrative dissimilarity matrix for four classifiers is shown in Table 4. In this case, higher values indicate higher diversity, thus the diagonal entries of the matrix are all 0. Furthermore, the matrix is symmetric so only the top half is shown.

A Sammon mapping [34] is a nonlinear multidimensional scaling projection onto a space $\Re^m$, where m is 2 or 3, such that the distances are preserved. For the purposes of CPS, let m be 2. An error function, called stress, is defined that measures the difference between the original dissimilarities and Euclidean distances. Let N be the $k \times k$ dissimilarity matrix with n_{ij} as elements, and $\tilde{N}$ be the distance matrix with $\tilde{n}_{ij}$ as elements for a projected configuration where $i, j = 1, 2, \ldots, k$. The stress is computed as [29]:

$$S = \frac{1}{\sum_{i=1}^{k-1} \sum_{j=i+1}^{k} n_{ij}^2} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} (n_{ij} - \tilde{n}_{ij})^2.$$

An initial distance matrix configuration must be used, and then the process proceeds in an iterative manner until a distance matrix configuration corresponds to a (local) minimum. An implementation of the Sammon mapping for Matlab under the GNU General Public License can be found at [13], which is the implementation used in this paper.

The resulting matrix from the Sammon map can be graphed such that the distance between the points reflects the relative diversity between the classifiers. The relative distance between two points represents the diversity between two classifiers. So, the further apart two points are, the more diverse those same points are from each other. The diversity matrix shown in Table 4 is put through the Sammon mapping, and the resulting CPS is shown in Fig. 9. Notice that points D_3 and D_4, those classifiers with the highest pairwise diversity values in Table 4, are the furthest away from each other. Alternately, points D_2 and D_4 have the lowest diversity values and are correspondingly the closest to each other. Whereas traditionally, a single average value of diversity would be reported for a given set of pairwise diversity measures, it is obvious that such a graph gives a simple and accurate view of the diversity measures. Furthermore, a "TRUE" point can be graphed that denotes all predictions being correct (e.g. the

Table 4 An example dissimilarity matrix of pairwise diversity measures for four classifiers

	D_1	D_2	D_3	D_4
D_1	0	0.69	0.85	0.71
D_2	–	0	0.73	0.55
D_3	–	–	0	0.98
D_4	–	–	–	0

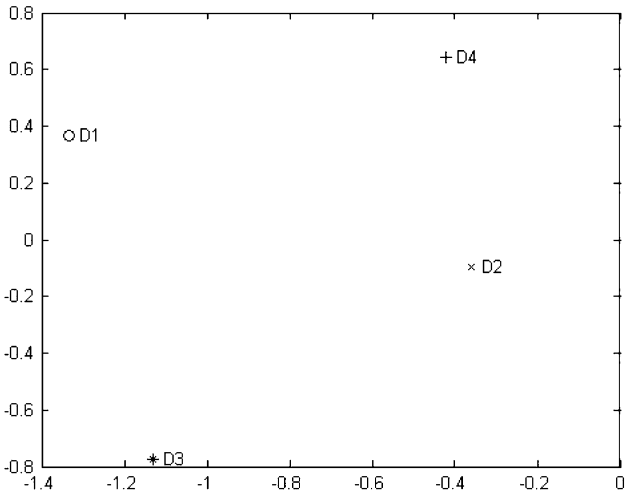


Fig. 9 An example CPS for the classifiers with the diversity matrix from Table 4

closer a classifier is to TRUE, the better prediction performance it has). Although this is not shown in Fig. 9, the TRUE point is used in the following diversity analysis.

3.4 Diversity analysis of GLHE and data manipulation ensembles

Three datasets are considered in this analysis—anneal, pendigits, and waveform—which were chosen to represent a variety of instance counts and class sizes. Each graph contains three J48 and three IBk points to represent the GLHE design, and four other points to specify the remaining learning algorithms in the data manipulation ensembles.

The diversities for the anneal dataset are presented in Fig. 10. Note that the J48 and IBk points are clustered together, each on one side of the TRUE point. The other classifiers are spread throughout the space further away from the TRUE point, especially KStar and NB. This graph further exposes the volatility of performance for a data manipulation ensemble if in fact a low-performing learning algorithm is chosen such as the KStar for the anneal dataset. Figure 11 shows the diversities for the pendigits dataset. Once again, GLHE's local classifiers based on IBk and the KStar are clustered tightly around the TRUE point while the others are spread out. The last diversity presentation is of the waveform dataset, given in Fig. 12. Here, all of the classifiers tend to be more evenly distributed in the space—even the J48 and IBk classifiers spread out. Three IBk-based local classifiers are within comparatively close range of the TRUE point while the J48-based classifiers are far away. The former appears to compensate for the latter set of classifiers for the performance of the GLHE in this case.

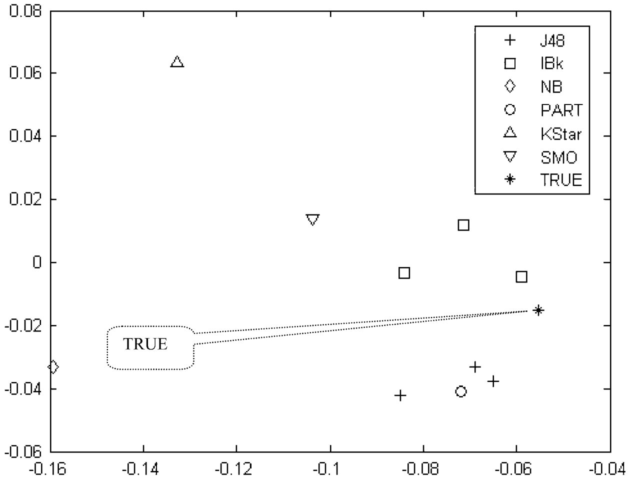


Fig. 10 Base learning algorithms in the CPS per the disagreement measure for anneal dataset. *Each point* represents a classifier while triple instances of J48 and IBk (as in GLHE) are shown, and the *distance between the points* indicate their pairwise diversity. The *TRUE point* has all predictions correct

Base classifier instances of GLHE tend to be closer to the TRUE classification point in the CPS graphs for all three datasets compared to other base classifiers which helps indicate, at least in part, the performance superiority of the GLHE over the data manipulation ensembles. It is also important however to note that current research is cautious not to put too much emphasis on diversity–performance relationship [23, 24, 38].

The visualizations of diversity suggest that co-existence of global and local learning algorithms offer high levels of diversity among its six instantiations of base learners. GLHE possesses higher levels of diversity due to two

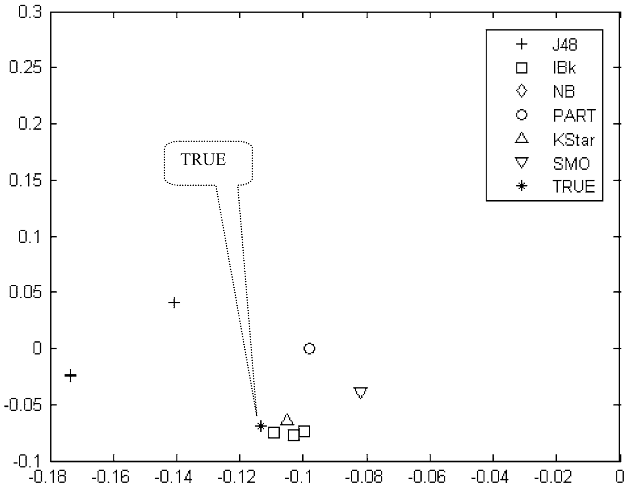


Fig. 11 Base learning algorithms in the CPS per the disagreement measure for pendigits dataset. *Each point* represents a classifier while triple instances of J48 and IBk (as in GLHE) are shown, The *TRUE point* has all predictions correct

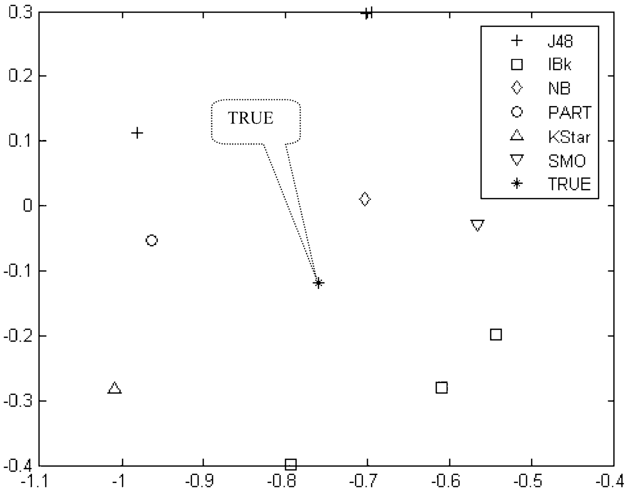


Fig. 12 Base learning algorithms in the CPS per the disagreement measure for waveform dataset. *Each point* represents a classifier while triple instances of J48 and IBk (as in GLHE) are shown. The *TRUE point* has all predictions correct

Table 5 Prediction accuracy performance of the bagging ensembles

	Bag-J48	Bag-IBk	Bag-NB	Bag-PART	Bag-KStar	Bag-SMO
Anneal	98.89	97.44	96.33	98.66	95.99	97.44
Audiology	79.65	68.58	71.68	82.30	79.20	78.32
Autos	84.88	74.15	68.78	80.49	73.66	72.68
Balance-scale	82.24	88.48	81.60	86.40	88.16	87.84
Breast-cancer	73.43	74.48	72.03	70.28	74.13	68.53
Breast-w	95.85	97.28	97.14	95.85	95.57	96.71
Car	93.11	93.97	85.34	96.99	87.02	93.45
cmc	54.11	45.76	51.26	52.48	50.85	49.36
Colic	82.34	79.35	76.09	80.71	69.29	82.07
Credit-a	85.36	86.67	86.09	85.94	79.28	85.22
Credit-g	74.00	75.10	75.70	74.10	71.10	75.30
Diabetes	74.09	72.14	76.95	74.35	69.27	77.47
Echocardiogram	95.95	93.24	97.30	95.95	91.89	93.24
Glass	71.03	71.03	70.56	73.83	75.70	55.61
Haberman	73.20	71.24	74.51	71.24	73.53	73.53
Heart-c	79.21	82.84	84.49	83.83	75.58	85.15
Heart-h	78.91	81.97	83.33	80.95	78.91	82.99
Heart-statlog	80.00	78.89	82.96	77.78	74.81	84.44
Hepatitis	83.23	83.23	83.87	83.87	81.94	85.81
Hypothyroid	99.58	93.53	98.17	99.66	94.51	93.58
Ionosphere	93.16	85.19	89.17	92.02	84.90	89.17
Iris	95.33	96.00	93.33	95.33	94.67	96.00
kr-vs-kp	99.44	96.46	87.83	99.37	97.09	95.87
Labor	84.21	84.21	87.72	82.46	91.23	87.72
Lymphography	79.05	85.81	87.16	85.81	84.46	85.81
Monk2	64.27	77.96	62.60	84.98	79.30	65.61
Mushroom	100.00	100.00	95.80	100.00	100.00	100.00
Page-blocks	97.33	96.13	93.40	97.53	97.02	93.26
Pendigits	98.02	99.29	87.85	98.68	99.19	98.02
Satellite	90.05	90.66	81.60	91.06	90.66	86.81
Segment	97.40	96.36	91.21	97.36	97.01	92.86
Sick	98.73	96.00	97.11	98.62	95.92	93.93
Solar-flare-c	84.31	80.85	78.69	83.51	83.51	84.31
Solar-flare-m	95.10	93.45	91.00	94.74	94.74	95.10
Solar-flare-x	99.14	98.70	96.33	99.06	98.92	99.14
Sonar	74.52	85.10	75.96	77.40	84.13	76.92
Soybean	93.27	90.78	92.53	92.97	87.85	93.27
Splice	94.48	82.95	95.30	93.70	79.12	94.73
Tic-tac-toe	93.31	99.06	70.29	99.48	95.50	98.54
Tumor	42.18	45.13	49.56	44.54	38.35	48.67
Vehicle	76.60	72.22	62.41	75.06	71.39	74.94
Vote	96.32	92.64	90.11	95.86	93.56	95.40
Vowel	90.40	96.26	65.86	89.60	98.79	70.10
Waveform	81.30	79.02	80.34	83.14	73.70	86.72
Wine	95.45	96.02	98.86	94.32	97.73	97.73
Zoo	93.07	94.06	94.06	92.08	96.04	95.05

Table 6 SAR performance of the bagging ensembles

	Bag-J48	Bag-IBk	Bag-NB	Bag-PART	Bag-KStar	Bag-SMO
Anneal	97.79	96.55	95.12	97.39	94.94	88.46
Audiology	87.82	83.11	84.44	88.99	87.42	84.26
Autos	87.26	81.17	77.72	85.37	80.11	77.57
Balance-scale	82.00	85.79	80.66	85.96	85.28	82.07
Breast-cancer	64.39	65.39	65.84	63.88	64.93	61.85
Breast-w	92.05	93.54	93.43	92.45	92.05	92.28
Car	92.12	91.25	86.74	95.26	87.88	86.29
cmc	60.26	53.43	59.11	59.48	58.44	56.68
Colic	75.00	75.38	70.62	74.05	68.83	75.97
Credit-a	81.92	81.47	81.59	82.30	75.47	78.94
Credit-g	69.03	69.06	71.22	69.90	64.95	68.38
Diabetes	70.71	68.86	72.68	71.78	64.55	68.46
Echocardiogram	91.94	89.67	94.11	91.94	87.51	87.91
Glass	78.41	78.51	78.32	80.35	81.89	66.67
Haberman	63.86	62.70	66.93	62.20	67.21	59.30
Heart-c	80.73	82.72	84.36	83.93	77.69	81.70
Heart-h	81.11	82.97	84.08	82.84	79.47	80.22
Heart-statlog	76.19	75.44	78.85	76.58	72.98	77.45
Hepatitis	76.40	77.16	79.62	78.74	74.53	78.33
Hypothyroid	98.28	85.11	96.56	98.37	92.70	74.49
Ionosphere	88.58	82.96	84.40	87.66	80.56	82.56
Iris	92.14	93.73	91.56	92.16	92.81	88.70
kr-vs-kp	97.48	91.55	84.34	97.40	91.95	91.79
Labor	77.07	83.34	84.27	79.36	87.69	84.67
Lymphography	80.29	84.07	85.10	84.21	83.86	81.48
Monk2	61.63	76.35	56.49	81.35	73.71	52.32
Mushroom	100.00	99.91	92.67	99.96	99.97	100.00
Page-blocks	95.56	94.00	92.35	95.79	95.26	78.95
Pendigits	97.35	98.63	90.88	97.84	98.55	90.20
Satellite	91.43	92.02	85.47	91.90	91.92	84.33
Segment	96.53	95.70	92.00	96.51	96.29	86.95
Sick	96.18	90.57	92.44	96.04	91.72	75.75
Solar-flare-c	73.55	75.81	77.41	79.03	78.20	68.49
Solar-flare-m	78.07	86.14	85.34	85.99	87.42	71.24
Solar-flare-x	81.42	91.55	90.79	92.33	91.88	75.03
Sonar	73.24	81.73	73.43	74.69	81.14	73.98
Soybean	94.96	93.91	94.55	94.94	92.61	90.43
Splice	91.68	83.51	93.14	91.28	80.78	87.95
Tic-tac-toe	88.92	91.87	67.31	95.82	89.09	94.78
Tumor	66.35	68.80	71.89	68.02	65.63	69.64
Vehicle	80.60	77.42	69.85	80.19	76.08	76.26
Vote	92.33	89.57	85.97	92.24	90.28	91.36
Vowel	92.35	95.55	80.28	91.85	98.03	79.71
Waveform	82.39	80.79	80.95	83.66	76.26	83.20
Wine	92.31	93.81	96.36	91.88	95.89	89.82
Zoo	93.40	94.22	94.49	93.05	95.59	87.69

Table 7 Prediction accuracy performance of the AdaBoost ensembles

	Boost-J48	Boost-IBk	Boost-NB	Boost-PART	Boost-KStar	Boost-SMO
Anneal	99.55	97.88	99.55	99.33	95.77	99.33
Audiology	84.96	71.24	77.88	84.51	77.88	82.74
Autos	86.34	73.66	68.78	82.44	73.66	77.07
Balance-scale	78.88	88.00	75.68	82.08	76.32	87.68
Breast-cancer	69.58	70.63	64.69	70.63	65.38	69.58
Breast-w	95.71	97.00	95.99	94.85	93.99	96.71
Car	96.06	93.22	89.86	98.67	92.99	93.57
cmc	50.78	44.94	51.05	48.88	46.50	48.07
Colic	82.88	74.18	77.17	80.43	67.93	80.16
Credit-a	84.20	85.94	86.81	83.33	78.26	83.77
Credit-g	69.60	74.20	76.30	71.90	69.60	75.00
Diabetes	72.40	73.18	74.35	74.35	67.97	77.34
Echocardiogram	94.59	93.24	95.95	94.59	90.54	91.89
Glass	74.30	71.96	70.56	75.23	74.77	57.01
Haberman	70.26	67.97	72.55	67.32	69.93	73.86
Heart-c	82.18	81.19	84.49	78.55	72.94	84.82
Heart-h	78.57	79.59	84.35	79.59	76.53	82.31
Heart-statlog	80.37	78.52	81.11	80.37	70.74	84.07
Hepatitis	85.81	83.23	85.81	83.87	77.42	81.29
Hypothyroid	99.58	88.28	98.78	99.60	95.15	94.99
Ionosphere	93.16	84.90	90.88	92.88	87.75	88.60
Iris	93.33	95.33	92.00	95.33	94.00	98.00
kr-vs-kp	99.50	96.31	94.96	99.62	96.50	97.18
Labor	89.47	84.21	82.46	85.96	82.46	84.21
Lymphography	81.08	85.14	83.78	79.73	83.11	83.78
Monk2	77.46	79.13	62.77	88.98	75.13	69.12
Mushroom	100.00	100.00	100.00	100.00	100.00	100.00
Page-blocks	97.06	96.02	95.92	97.08	96.23	92.89
Pendigits	99.12	99.30	92.83	98.96	98.94	98.15
Satellite	90.58	90.68	81.63	90.80	89.28	86.88
Segment	98.48	96.15	93.98	98.31	96.84	93.29
Sick	99.18	95.49	97.53	98.91	96.39	94.41
Solar-flare-c	83.01	82.00	78.83	83.59	83.15	84.31
Solar-flare-m	93.30	93.38	90.78	93.30	92.94	95.10
Solar-flare-x	98.70	98.49	98.20	98.78	98.63	99.06
Sonar	77.88	85.58	80.29	80.29	85.58	75.96
Soybean	92.83	90.34	92.83	94.44	86.97	92.68
Splice	93.17	82.13	93.73	94.01	78.97	93.86
Tic-tac-toe	95.61	99.27	79.81	98.95	98.01	97.91
Tumor	40.12	42.18	50.15	42.77	35.40	46.90
Vehicle	76.24	73.29	60.05	76.60	69.74	74.35
Vote	95.86	93.33	95.17	94.94	91.72	95.63
Vowel	93.33	96.77	60.10	91.41	98.59	78.79
Waveform	80.48	78.94	79.86	81.84	70.30	86.68
Wine	97.16	94.89	96.59	94.32	97.16	97.73
Zoo	95.05	95.05	93.07	94.06	97.03	96.04

Table 8 SAR performance of the AdaBoost ensembles

	Boost-J48	Boost-IBk	Boost-NB	Boost-PART	Boost-KStar	Boost-SMO
Anneal	98.76	96.81	98.46	98.14	93.78	98.20
Audiology	89.98	82.22	86.34	90.07	85.92	88.01
Autos	87.69	77.25	77.50	85.59	78.39	79.99
Balance-scale	79.41	84.93	75.98	82.04	77.92	87.22
Breast-cancer	60.36	58.36	59.88	61.55	56.31	61.65
Breast-w	91.26	93.46	91.98	90.66	89.08	91.90
Car	94.07	91.04	89.58	96.88	91.24	92.33
cmc	56.72	48.38	58.37	55.75	52.37	54.24
Colic	74.67	68.85	70.82	73.72	63.10	73.47
Credit-a	79.28	78.58	81.61	78.63	71.69	80.26
Credit-g	63.78	68.57	69.11	65.46	61.05	68.61
Diabetes	66.64	69.07	68.23	69.50	59.93	70.42
Echocardiogram	89.72	90.30	93.91	89.78	84.49	89.88
Glass	80.32	78.33	77.82	79.96	79.26	65.20
Haberman	60.58	57.19	62.69	58.92	61.84	64.49
Heart-c	81.53	81.69	82.81	79.67	72.85	83.67
Heart-h	79.30	79.75	83.14	79.83	74.87	83.72
Heart-statlog	74.64	74.94	73.46	74.80	66.00	78.05
Hepatitis	76.97	73.01	76.05	76.18	68.30	73.54
Hypothyroid	98.26	80.60	96.81	98.25	91.81	90.05
Ionosphere	87.81	82.83	85.36	87.78	80.91	81.73
Iris	89.27	93.50	88.42	91.77	90.31	95.62
kr-vs-kp	97.51	91.58	91.12	97.80	92.40	93.96
Labor	82.07	78.43	76.87	80.90	78.03	76.70
Lymphography	80.92	83.88	81.67	79.34	80.63	81.15
Monk2	71.64	77.38	54.06	84.47	68.81	67.42
Mushroom	100.00	100.00	99.98	100.00	99.97	100.00
Page-blocks	95.07	90.75	93.79	95.16	93.84	90.09
Pendigits	98.36	98.65	93.76	98.18	98.11	97.47
Satellite	91.16	91.96	85.04	91.28	90.01	87.23
Segment	97.27	95.58	93.81	97.15	95.76	93.28
Sick	96.41	87.20	92.74	95.87	90.67	89.51
Solar-flare-c	76.86	74.76	73.64	76.67	76.10	77.09
Solar-flare-m	84.31	83.24	82.35	84.31	84.16	85.16
Solar-flare-x	85.76	81.26	88.86	90.55	88.39	90.61
Sonar	73.89	82.00	73.75	76.25	80.67	72.20
Soybean	94.55	92.42	94.64	95.40	90.76	94.36
Splice	89.76	82.96	90.87	90.90	78.41	89.67
Tic-tac-toe	92.35	91.99	77.00	96.47	94.89	95.06
Tumor	63.61	65.21	68.22	66.06	62.98	66.19
Vehicle	78.78	77.63	68.08	78.85	72.34	77.91
Vote	91.98	87.64	91.30	90.61	86.71	92.03
Vowel	94.24	96.24	77.26	93.06	97.79	86.74
Waveform	79.98	80.31	79.89	80.99	70.92	84.39
Wine	94.31	92.99	94.00	91.09	94.41	94.38
Zoo	93.45	94.61	94.27	92.61	95.87	88.07

sources as indicated by these diversity visualization graphs. The first one is the presence of both global and local base learners: global learner J48 and local learner IBk clusters are located apart from each other, which translates into large pairwise diversity. Secondly, each separate instantiation of either base learner in GLHE, namely J48 and IBk, creates additional diversity, although not as large as it is between a global instance and local instance of a base learner. The inherently high levels of diversities for the GLHE is likely to be a major source of performance enhancement compared to the data manipulation ensembles discussed in this study although it is most likely not the only major one.

4 Conclusions

This paper presented a comparative performance study of global–local hybrid classification ensemble with data manipulation ensembles based on simulation and diversity analysis. The simulation study employed 46 datasets from the UCI Machine Learning Repository. Statistical significance tests were employed to compare the performance of one ensemble with that of another. Diversity analysis employed the dissimilarity measure and the classifier projection space methodology to analyze the inherent diversities which each ensemble possessed. Simulation results indicated that global–local hybrid ensemble offers superior performance in terms of prediction accuracy and SAR metrics compared to six data manipulation ensembles based on bagging and boosting variants. Global–local hybrid ensemble was shown to have prediction accuracy better than three of the bagging and boosting ensembles while also demonstrating an equivalent performance with the rest. In terms of the SAR metric, global–local hybrid ensemble had better performance than two bagging and four boosting ensembles while scoring a tie with the rest. The diversity analysis further provided support and explanation for the performance superiority of global–local hybrid ensemble. Global–local hybrid ensemble projects a more robust performance profile when compared to boosting and bagging ensembles when a large number of problem domains or, equivalently, data sets are considered.

Appendix: Classification performance for data manipulation ensembles

See Tables 5, 6, 7 and 8.

References

1. Aha D, Kibler D (1991) Instance-based learning algorithms. Mach Learn 6:37–66
2. Asuncion A, Newman DJ (2007) UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine. http://www.ics.uci.edu/~mlearn/MLRepository.html
3. Baumgartner D, Serpen G (2009) Large experiment and evaluation tool for WEKA classifiers. In: 5th international conference on data mining. Las Vegas, pp 340–346
4. Baumgartner D, Serpen G (2012) A design heuristic for hybrid ensembles. Intell Data Anal 16(2):233–246
5. Banfield RE, Hall LO, Bowyer KW, Bhadoria D, Kegelmeyer WP (2007) A comparison of decision tree ensemble creation techniques. IEEE Trans Pattern Anal Mach Intell 29(1):173–180
6. Battista B, Fumera G, Roli F (2010) Multiple classifier systems for robust classifier design in adversarial environments. Int J Mach Learn Cybern 1:27–41
7. Bian S, Wang W (2007) On diversity and accuracy of homogeneous and heterogeneous ensembles. Int J Hybrid Intell Syst 4:103–128
8. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
9. Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Inf Fusion 6:5–20
10. Caruana R, Niculescu-Mizil A, Crew G, Ksikes A (2004) Ensemble selection from libraries of models. In: Proceedings of the 21st international conference on machine learning, pp 137–144
11. Caruana R, Niculescu-Mizil A (2004) Data mining in metric space: an empirical analysis of supervised learning performance criteria. In: Proceedings of the 10th ACM SIGKDD international conference on knowledge discovery and data mining, Seattle, pp 69–78
12. Canuto AM, Abreu MC, Oliveira LM, Xavier JC, Santos AM (2007) Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. Pattern Recognit Lett 28:472–486
13. Cawley GC, Talbot NL (n.d.) Miscellaneous Matlab Software. http://theoval.sys.uea.ac.uk/~gcc/matlab/default.html. Accessed January 2009
14. Demsar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
15. Dietterich TG (2000) Ensemble methods in machine learning. Lect Notes Comput Sci 1857:1–15
16. Dunn OJ (1961) Multiple comparisons among means. J Am Stat Assoc 56:52–64
17. Dzeroski S, Zenko B (2004) Is combining classifiers with stacking better than selecting the best one? Mach Learn 54:255–273
18. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13th international conference on machine learning, pp 148–156
19. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat 11:56–92
20. Hand DJ, Vinciotti V (2003) Local versus global models for classification problems: fitting models where it matters. Am Stat 57(2):124–131
21. Iman RL, Davenport JM (1980) Approximations of the critical region of the Friedman statistic. Commun Stat 571–595
22. Kotsiantis SB, Pintelas PE (2004) A hybrid decision support tool—using ensemble of classifiers. Int Conf Enterp Inf Syst (ICEIS) 2:448–456
23. Kuncheva LI, Whitaker CJ (2003) Measures of diversity in classifier ensembles and their relationship to ensemble accuracy. Mach Learn 51:181–207
24. Kuncheva LI (2003) That elusive diversity in classifier ensembles. Lect Notes Comput Sci 2652:1126–1138
25. Luengo J, Garcia S, Herra F (2007) A study on the use of statistical tests for experimentation with neural networks. In:

Proceedings of the 9th international work-conference on artificial neural networks. Lecture notes on computer science, vol 4507, pp 72–79

26. Mitchell TM (1997) Machine learning. McGraw-Hill, NY
27. Opitz D, Maclin R (1999) Popular ensemble methods: an empirical study. J Artif Intell Res 11:169–198
28. Ott RL, Longnecker M (2001) An introduction to statistical methods and data analysis, 5th edn. Duxbury, Pacific Grove
29. Pekalska E, Duin RP, Skurichina M (2002) A discussion on the classifier projection space for classifier combining. In: Roli F, Kittler J (eds) 3rd international workshop on multiple classifier systems, MCS02, vol 2364. Springer, Cagliari, pp 137–148
30. Polikar R (2006) Ensemble based systems in decision making. IEEE Circuits Syst Mag 6(3):21–45
31. Provost F, Domingos P (2003) Tree induction for probability-based ranking. Mach Learn 52(3):199–215
32. Quinlan JR (1996) Bagging, boosting, and C4.5. In: Proceedings of the 13th national conference on artificial intelligence, pp 725–730
33. Ricci F, Aha DW (1998) Error-correcting output codes for local learners. In: 10th european conference on machine learning, ECML. Springer, Berlin, pp 280–291
34. Sammon JW (1969) A nonlinear mapping for data structure analysis. IEEE Trans Comput 18:401–409
35. Seewald K, Furnkranz J (2001) An evaluation of grading classifiers. In: Proceedings of the 4th international conferences on advances in intelligent data analysis, pp 115–124
36. Seewald K (2002) How to make stacking better and faster while also taking care of an unknown weakness. In: Proceedings of the nineteenth international conference on machine learning, pp 554–561
37. Skalak D (1996) The sources of increased accuracy for two proposed boosting algorithms. In: AAAI '96 workshop on integrating multiple learned models for improving and scaling machine learning algorithms
38. Tang EK, Suganthan PN, Yao X (2006) An analysis of diversity measures. Mach Learn 65:247–271
39. Witten H, Frank E (2005) Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco
40. Wolpert DH (1992) Stacked generalization. Neural Netw 5(2):241–260
41. Yates WB, Patridge D (1996) Use of methodological diversity to improve neural network generalization. Neural Comput Appl 4(2):114–128
42. Zhiwen Y, Zhongkai D, Wong HS, Tan L (2010) Identifying protein kinase-specific phosphorylation sites based on the bagging-Adaboost ensemble approach. IEEE Trans Nanobiosci 9(2):132–143