Novel Methods for the Feature Subset Ensemble Approach

Mohamed A. Aly
Dept Electrical Engineering, Caltech, Pasadena, CA 91125, USA
mohamedadaly@gmail.com

Amir F. Atiya
Dept Computer Engineering, Cairo University, Giza, Egypt
amiratiya@link.net

Abstract

*Ensemble learning technique attracted much attention in the past few years. Instead of using a single prediction model, this approach utilizes a number of diverse accurate prediction models to do the job. Many methods have been proposed to build such accurate diverse ensembles, of which bagging and boosting were the most popular. Another method, called Feature Subset Ensembles (*FSE*), is thoroughly investigated in this work. This technique builds ensembles by assigning each individual prediction model in the ensemble a distinct feature subset from the pool of available features. In this paper several novel variations to the basic* FSE *are proposed. Extensive comparisons are carried out to compare the proposed* FSE *variants with the basic* FSE *approach.*

1 Introduction

Over the history of machine learning one single learning model was typically built to solve a given problem at hand. From a set of candidate prediction models or networks, only one is chosen to do the job. However, this single model may not be the best one available. Moreover, helping it with other prediction models can prove advantageous in improving the prediction accuracy. The technique of using multiple prediction models for solving the same problem is known as ensemble learning. It has proved its effectiveness over the last few years.

The ensemble approach has been an active research topic in the past few years [8, 9]. In this approach, a group of prediction models are trained and used instead of just employing the best prediction model. The outputs of the individual prediction models are combined together, using simple averaging or voting for example, to produce the ensemble output. This technique has been proved, both theoretically and empirically, to significantly outperform the single prediction model approach [35, 19]. Using multiple prediction models can get around a single prediction model overfitting the data and can decrease the variance in its predictions. However, for the ensemble to produce good performance, the component prediction models not only need to be accurate, but they also need to be diverse i.e. their generalisation errors should be as least correlated as possible [18, 5]. This is intuitive, because nothing can be gained from using prediction models that give identical predictions. Researchers have developed many ways that are capable of producing accurate diverse prediction model ensembles [8, 9, 5]. The most popular methods are bagging [4], and boosting [13, 14]. Both bagging and boosting are based on creating an ensemble of networks, each trained using a different subset of training examples.

A relatively novel ensemble approach, called feature subset ensembles (FSE), has been proposed in the literature. It has been named by different names in the literature, but we propose to unify the naming under this term. The approach is based on creating an ensemble of networks, where each network operate on a different subset of features (input variables). The *FSE* approach has not yet been sufficiently explored in the literature, and this paper serves to shed some light on its capabilities. Specifically, we propose several variants of the *FSE* approach, and conduct a large scale comparison study. Some of the proposed variants turned out to outperform the basic *FSE*.

2 The Feature Subset Ensembles Approach

Feature selection is a very important part of the preprocessing phase in machine learning [23, 8, 25] and statistical pattern recognition [37, 22, 19, 27]. In many real world situations, we are faced with problems having hundreds or thousands of features, some of which are irrelevant to the problem at hand. Feeding learning algorithms with all the features can result in a deteriorating performance, as the algorithm can get stuck trying to figure out which features are useful and which are not. Therefore, feature selection is employed as a preliminary step, to selected a subset of the input features, that contains potentially more useful features. In addition, feature selection tends to reduce the dimensionality of the feature space, avoiding the well-known dimensionality curse problem [19].

The disadvantage of feature subset selection is that some features that may seem less important, and are thus discarded, may bear valuable information. It seems a bit of a waste to throw away such information, that could possibly in some way contribute to improving model performance. This is where Feature Subset Ensemble (FSE) comes into play. It simply partitions the input features among the individual prediction models in the ensemble. Hence, no information is discarded. It utilizes all the available information in the training set, and at the same time not overload a single prediction model with all the features, as this may lead to poor learning.

Let us give an illustative example. Assume we have ten features, ranked in terms of effectiveness. This means the feature selection algorithm selected feature 1, then feature 2, and so on. Assume that we have selected up to $K = 5$ features, so we use features $\{1, 2, 3, 4, 5\}$ and discard the rest. The discarded features provide some useful information. For example features 6 and 7 might not be much worse than feature 5, and so could be useful to consider. In FSE, we would for example design 4 classifiers each taking a different set of features: $\{1, 2, 5, 7, 9\}$, $\{1, 3, 4, 6, 8\}$, $\{2, 3, 5, 6, 7\}$, and $\{1, 2, 3, 4, 10\}$. The output of the four networks will then be combined by averaging or voting.

3 Review of FSE Approaches

The FSE approach is a fairly new approach. Probably less than around ten papers appeared on that topic. Most of the techniques use random assignment of features among the networks. We believe That there is room for improving FSE's performance by utilizing intelligent assignment techniques or specific weighting techniques, and this is one of the issues studied in this work. First, we will here provide a review about the work that has appeared on FSE.

Ho [20, 21] proposed a technique she called *Random Subspace Method (RSM)*. In this technique, a subset of features was randomly selected for each prediction model. She used C4.5 decision trees [34] as the base prediction model. The number of features selected for each prediction model was half the total number of features. Experiments were undertaken to compare the RSM to bagging, boosting and single tree prediction models employing all features. Four publicly available datasets from Project StatLog [3] were used. In each ensemble technique, a decision forest was grown up to 100 decision trees. RSM showed better performance than bagging, boosting, and single tree prediction model.

Bay [2] introduced a similar idea, called *Multiple Feature Subsets (MFS)*. In this method, he applied random feature selection to nearest neighbor (NN) prediction models. Each NN prediction model was trained using a random subset of features. Bay noticed that other ensemble learning techniques, e.g. bagging or boosting, that depend on subsampling the training set, failed to improve NN ensembles. Therefore, he worked on another method that manipulated input features instead of input patterns. Bay used two sampling functions: sampling with replacement and sampling without replacement. In sampling with replacement, a given feature can be replicated within the same prediction model. In sampling without replacement, however, a given feature can not be assigned more than once to the same prediction model.

Bryll et al. [6] also used a similar approach, which they called *Attribute Bagging (AB)*. In this approach, each prediction model is trained on a subset of randomly selected features (without replacement). They proposed a framework, in which the size of the feature subset is determined first, and this parameter is problem dependent. Then, various subsets of that size are evaluated using the wrapper method [25], and only the best of these subsets are used for voting.

The methods discussed above are nearly similar in that they assign features randomly to each individual prediction model. They differ in the way their parameters (subset and ensemble sizes) are estimated. In addition, they were tested only on classification problems not regression tasks.

Alkoot and Kittler [1] proposed an approach that uses traditional feature selection algorithms in order to maximize the overall ensemble performance. They proposed three different variations for building the ensemble: the parallel system, the serial system, and the optimized conventional system. In the parallel system, each *expert* (the term they used for a *prediction model*) is allowed, in turn, to take one feature such that the overall ensemble performance is optimized on a validation set. In the serial system, in contrast, the first expert is allowed to take all the features that achieve the maximum ensemble accuracy on the validation set. If some features remain, a second expert is used, and so on. The optimized conventional system builds each expert independently, and then features are added/deleted from the ensemble as long as the ensemble performance is increased.

Günter and Bunke [17] proposed an ensemble creation technique based on feature selection algorithms. They tested their method in the context of handwritten word recognition, using Hidden Markov Model (*HMM*) recognizer [28] as the base prediction model. In their approach, each prediction model is given a well performing set of features using any existing feature selection algorithm. Gnter and Bunke used two well known algorithms: floating sequential forward and backward search algorithms [33]. Each prediction model uses one of the two floating search algorithms to get a unique feature subset.

Opitz [31] presented a Genetic Algorithm (*GA*) approach for ensemble creation, called Genetic Ensemble Feature Selection (*GEFS*). Opitz noted that GA can be used to search through the large space of feature subsets, and to select the best of such subsets to create a powerful prediction model ensemble, such that those subsets create a diverse ensemble. He argued that this task is an enormous problem, and can be tackled using global optimization techniques such as *GA*s.

Guerra-Salcedo and Whitley [16] proposed another *GA*-based approach for ensemble creation. However, they used table-based prediction models, namely KMA [10] and Euclidean Decision Tables (*EDT*)[15]. They applied the CHC genetic search algorithm [11].

Oliveira et al. [30] also proposed a *GA*-based ensemble creation technique. Their technique also used a *GA* to find good feature subsets, however they employed a hierarchical two-phase approach to ensemble creation. In the first phase, a set of good prediction models are generated using Multi-Objective Genetic Algorithm (*MOGA*) search[29]. The base prediction models used were Artificial Neural Networks (*ANN*), however any type of prediction model can be used. The second phase searches through the space created by the different combinations of these good prediction models, again using *MOGA*, to find the best possible combination i.e. the best ensemble.

Cherkauer [7] introduced a system called *Plannett (*Person-Level Artificial Neural Networks for ExtraTerrestrial Terrain classification), which combines ANNs in order to achieve better accuracy in the difficult task of recognizing volcanoes in radar images of planet Venus.

Liao and Moody [26] proposed a technique called *Input Feature Grouping*. The first step in the technique is to group the input features into clusters based on their mutual information, such that features in each group are greatly correlated to each other, and are as little correlated with features in other groups as possible. In the second step, each member of the ensemble is given a representative of each feature cluster. Liao and Moody applied their technique on an economic forecasting problem having 9 input features. A hierarchical clustering algorithm [12] was used to cluster the input features.

Tumer and Oza [36, 32] presented an approach called *Input Decimation Ensembles (IDE)*. Their method is only applicable to classification tasks. For a classification problem with L class labels, it constructs L prediction models. Each prediction model is given a subset of the input features, such that these features are the most correlated with that class. The ensemble final output is the average of the individual prediction models output.

4 The Proposed Variants

We tried out several variations on the basic *FSE* trying to add more accuracy and stability. First, we tried two sampling functions: with and without replacement. Second, we tried to change the weighting scheme, to give higher emphasis on the more accurate prediction models. Two weighting schemes were added beside the normal equal weighting: weight according to training error and weight according to prediction model relevance. Third, we tried to mix up bagging and boosting with the *FSE*. prediction models were given random feature subsets as usual, however they are not trained using the whole training set, but rather on a subsample thereof. Fourth, in an effort to increase the diversity among different prediction models, we employed feature selection within *FSE*. Individual prediction models are given random feature subsets, but they are not all used, instead sequential feature selection is applied to this subset to select the best components out of it.

4.1 Sampling Functions

The sampling funciton originally used in most of the previous methods, specially RSM [21] and AB [6], was sampling without replacement. In this method, each prediction model is given a subset of features by sampling without replacement from the pool of features i.e. when a feature is chosen in the subset for a certain prediction model, it can not be further chosen for the *same* prediction model, while it can be chosen in other prediction models. On the other hand, MFS [2] used sampling with replacement in addition to sampling with replacement. In this technique, a given feature can be repeated in the subset chosen for a given prediction model. Effectively, this is a way of reducing the number of features chosen as these replicated features add no new information to the prediction model. We experimented with both sampling techniques, in order to see the effect of the sampling function on the output of the ensembles.

4.2 Weighting Functions

Many of the ensemble techniqes relied on equal weighting of the individual prediction models [9]. The prediction models have equal votes (in case of classification) or equal weights in averaging (in case of regression). Furthermore, some ensemble techniques assigned weights to component prediction models relative to their performance, either on the training set or on some validation set [6, 14]. This weighting mechanism proved useful in some situations, specially in boosting, to give larger emphasis on better prediction models.

We tried out the above two weighting approaches: equal weighting and weighting according to prediction model performance on the training set. In equal weighting, each prediction model k was assigned a weight $w_k = \frac{1}{K}$, where K is the ensemble size. In the second method, each prediction model's weight was computed using the softmax formula: $w_k = \frac{\exp(-e_k)}{\sum_{i=1}^{K} \exp(-e_i)}$, where e_k is the training error for prediction model k.

In addition to these two techniques, we used a third weighting mechanism. This relied on the relevance of the features selected for each prediction model. The features in the dataset were ranked according to their relevance [23], employing a variant of sequential forward selection (*SFS*) algorithm with 10-fold cross-validation. The algorithm stops when all the features are selected in the subset, and bases its estimate of the subset performance on the result of 10-fold cross-validation [24]. Then, each prediction model is given a weight according to the ranks of the features in its feature subset. The weight given to prediction model k is defined by $w_k = \frac{w'_k}{\sum_{i=1}^{K} w'_i}$ where $w'_k = \sum_{i=1}^{n} r_i$, r_i is the rank of feature i, n is number of features in each subset, and K is the total number of prediction models. This weighting scheme measures the strength of each prediction model in terms of the strength of features it has. It assigns larger weights to hopefully more relevant prediction models, those that have the most relevant features.

4.3 Training Set Subsampling

Training set subsampling has been proved effective in creating accurate diverse ensembles. The two most successful ensemble techniques, bagging and boosting, are examples of training set subsampling. Therefore, we tried to embed both of bagging and boosting into *FSE*, as this has the potential to increase the diversity of the produced ensembles. This seemed attractive, as it combined two orthogonal approaches, feature set subsampling and training set subsampling, and they both perform well independently. Hence, we tried to test their behaviour when working together.

When using bagging within *FSE*, each prediction model gets its own feature subset, and then trains on an independent bootstrap sample drawn from the original training set. The advantage of this approach, like bagging, is that the individual prediction models are completely independent of each other, and can be created and trained in parallel.

On the other hand, using boosting within *FSE* is a bit more complicated. The version of boosting implemented is *AdaBoost.R*. After providing each prediction model with its feature subset, they are trained in sequence, with each one given a different sample of the training set based on the performance of the previous prediction models. Those patterns that exhibit higher error with the previous prediction model are given higher probability of beging represented in the sample of the new prediction model. Thus, harder patterns are given more emphasis than easier ones and the prediction model can focus its attention to learning those hard-to-learn paterns.

4.4 Embedded Feature Selection

We tried yet another technique to help increase diversity of *FSE*. In using randomly selected feature subset for each prediction model, features can get mingled together in a way that might worsen that prediction model's performance. Some researchers already used other feature selection techniques for building prediction model ensembles, e.g. genetic algorithms [16, 30, 31] and other mechanisms [1, 17]. Hence, we thought of using feature selection after the random feature sampling. In this approach, and after giving each prediction model its random share of the features, a feature selection algorithm is applied to choose the best out of those features. Two algorithms were tried: Sequential Forward Selection SFS [22] and Sequential Floating Forward Selection $SFFS$ (algorithm [22]). The former is a simple and well-established feature selection algorithm. The latter has been proved more accurate and stable [22]. The versions of the algorithms employed use 10-fold cross-validation for error estimation and stop when the performance starts to deteriorate.

4.5 Feature Subset Selection Criteria

In addition to the above variations to the basic *FSE*, we tried other systematic feature subset selection criteria. The basic *FSE* uses random feature subsets for each prediction model. We thought that having a more advanced feature selection criteria that can make use of the knowledge about the features, their relevance and correlation, has the potential of providing better results. Two criteria were tried out for feature subset selection: feature relevance and feature correlation.

4.5.1 Feature Relevance Criteria

Each feature has some inherent strength with respect to the dataset it represents. Knowing those strong features can help us partition the features into strong diverse subsets. The features relevances are determined as described earlier in section 4.2 using a variant of sequential forward selection SFS algorithm. Then, this ranked feature list is divided into two equal lists: strong features list and weak features list. Two techniques are employed to select feature subsets for the prediction models, which we will call pure and hybrid techniques. In the pure technique, each prediction model takes its subset randomly from either the strong list or the weak list. Half the prediction models take all their features from the strong list, and the rest take theirs from the weak list. On the other hand, each prediction model in the hybrid technique takes half its subset from the strong list and the other half from the weak list.

4.5.2 Feature Correlation Criteria

Correlation between features is a major issue in mahine learning. Using highly correlated features adds little information to the learning process. We tried to have a measure of the corrlation between features, and wanted to roughly divide them into a group correlated features and antoher of uncorrelated features. We employed a heuristic approach to achieve this objective. First, the correlation correlation coefficient matrix is calculated for the features from the training set. Then, the features are ranked descendingly according to the abosolute value of their mutual correlation. This is achieved by choosing the maximum correlation value, extracting the two features intersecting in this value, and adding them to the output list. Then, the next highest value is found, and so on. After that, the feature list is again divided into two equal lists: the correlated list and the uncorrelated list. Likewise, two techniques are used to select feature subsets for the prediction models from these two lists. In the pure technique, half the prediction models take all their features from the strong list and the rest take theirs from the weak list. In the hybrid, on the other hand, each prediction model chooses a mixture of features from the two lists.

5 Simulations Experiments

We carried out a large scale simulation to assess the effectiveness of the developed variants of FSE. We considered here only regression problems. For classification problems a similar study has yet to be performed. We performed the comparison simulations on six datasets obtained from the UCI repository, namely the datasets called bank32nh, ailerons, syn, house16h, comp, and pole.

The prediction models used were the following. The first one is the least square error model (*LSE*) which is simply a linear regression model, i.e. it combines the features linearly. The other model used is classification and regression trees *CART*. As is well-known, *CART* builds a binary tree splitting the space according to the training data. Then, it can get the predicted value of any unknown input by following the path from the root to the leaves of the tree.

To be able to get a fair comparison between *FSE* and the proposed variants, we performed more than one run on the datasets. Ten independent runs for each dataset were performed. In each of the runs, a random subset is chosen from the dataset for training, with the remaining used for testing. The training set size was chosen to be 200 patterns. For each prediction model, the *FSE* variants are compared against the basic *FSE* as well as the single prediction model using all the features (i.e. using one network rather than an ensemble), *SFS*, and *SFFS* (both are well-known feature selection algorithms based on sequential selection of features [22]). Some of the model parameters are tuned using 10-fold validation, while others are fixed once and for all, based on small experimental tuning on data sets other than the ones considered. Table 1 gives the abbreviations used for each of the used *FSE* methods. The performance measure used is the Normalized Mean Squared Error (*NMSE*), which is defined by $NMSE = \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i \hat{y}_i^2} \times 100\,\%$, where $\hat{y}_i$ is the predicted value and y_i is the target value. Tables 2 and 3 give the average of the out of sample *NMSE* for respectively the LSE model and the $CART$ model.

It can be seen that *FSE-06* and *FSE-07*, i.e. the FSE methods with embedded feature selection, are superior for the LSE case. For $CART$, *FSE-04* (the FSE with bagging implemented in it), is the clear winner. We believe this might be attributed to the fact that using FSE and bagging provides the maximal diversity, which one of the most important criteria for ensemble models. It provides maximal diversity because the diversity comes from two aspects: training set subsampling and feature subset selection. *FSE-04* on the other hand did not perform too well for the LSE model. The reason, perhaps, is the the well-known fact that bagging does not peform too well for linear models. It is particularly suited for nonlinear models.

6 Conclusions

In this study we have performed a large scale comparison of FSE methods for regression problems. There are many ways to partition features among the ensemble models, and we proposed several ways for doing that, based on sampling method, weighting functions, training set subsampling, embedded feature selection, feature correlation, and others. A possible future study is to assess these methods for classification problems.

References

[1] Fuad M. Alkoot and Joseph Kittler. Feature selection for an ensemble of classifiers. In *Proceedings of the 4^{th} Multiconference on Systematics, Cybernatics, and Informatics*, pages 379–384, Orlando, Florida, 2000.

Table 1. $FSEs$ variations

Name	Description
FSE-00	Basic FSE with random partitioning, sampling without replacement, and equal weighting
FSE-01	FSE with sampling with replacement
FSE-02	FSE with relevance weighting
FSE-03	FSE with training error weighting
FSE-04	FSE with bagging
FSE-05	FSE with boosting
FSE-06	FSE with SFS
FSE-07	FSE with $SFFS$
FSE-08	FSE with pure feature relevance criteria
FSE-09	FSE with hybrid feature relevance criteria
FSE-10	FSE with pure feature correlation criteria
FSE-11	FSE with hybrid feature correlation criteria

Table 2. Summary for all methods using LSE prediction models

Method	bank32nh	ailerons	syn	house16h	comp	pole
Single	39.71	4.43	47.81	41.79	4.36	47.43
SFS	38.01	4.35	42.50	42.98	3.77	40.87
$SFFS$	38.05	4.35	43.46	42.98	3.77	41.41
FSE-00	38.37	4.34	38.58	33.59	4.33	40.46
FSE-01	41.03	4.89	38.47	31.40	13.09	40.05
FSE-02	38.54	4.34	38.64	33.48	4.41	40.30
FSE-03	38.42	4.37	38.50	33.99	4.35	40.51
FSE-04	38.48	4.39	39.33	55.16	4.68	46.03
FSE-05	51.60	5.52	43.30	100.20	4.79	43.17
FSE-06	36.21	4.14	38.82	51.11	3.82	38.76
FSE-07	36.57	4.32	37.54	53.83	3.79	39.13
FSE-08	44.49	5.33	42.73	48.62	19.15	45.55
FSE-09	38.13	4.34	39.52	52.24	3.78	39.05
FSE-10	45.01	5.10	43.84	49.10	22.09	44.03
FSE-11	38.11	4.35	39.42	51.19	4.49	39.26

Table 3. Summary for all methods using $CART$ prediction models

Method	bank32nh	ailerons	syn	house16h	comp	pole
Single	75.98	8.71	130.5	59.18	0.30	14.57
SFS	68.21	9.21	141.1	61.48	0.26	20.43
$SFFS$	68.49	9.22	100.8	52.92	0.26	20.39
FSE-00	49.90	6.43	69.21	35.84	0.17	11.72
FSE-01	49.20	6.43	69.04	36.17	0.19	15.61
FSE-02	49.30	6.39	69.61	35.57	0.17	11.61
FSE-03	49.61	6.36	69.16	35.99	0.17	11.47
FSE-04	42.13	5.09	66.17	34.47	0.16	10.98
FSE-05	49.08	5.33	63.57	37.05	0.17	11.28
FSE-06	49.77	7.56	85.38	38.89	0.17	12.67
FSE-07	68.61	7.21	100.1	39.04	0.16	11.19
FSE-08	54.42	8.51	70.25	36.10	0.79	24.32
FSE-09	49.26	6.82	68.51	35.43	0.16	10.55
FSE-10	54.92	8.49	69.94	36.25	0.76	24.30
FSE-11	49.39	6.81	68.79	35.57	0.17	10.53

[2] Stephen D. Bay. Combining nearest neighbor classifiers through multiple feature subsets. In *Proceedings of the 17^{th} International Conference on Machine Learning*, pages 37–45, Madison, WI, 1998.

[3] P. Brazdil. Project StatLog, LIACC, University of Porto, http://www.liacc.up.pt/ML/statlog/datasets.html, 1999.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24:124–140, 1996.

[5] Gavin Brown, Jeremy Wyatt, Tachel Harris, and Xin Yao. Diversity creation methods: A survey and categorisation. *Pattern Recognition Society*, 2004.

[6] Robert Bryll, Ricardo Gutierrez, and Francis Quek. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern Recognition Society*, 2002.

[7] Kevin J. Cherkauer. Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks. In *Proceedings of the 13^{th} National Conference on Artificial Intelligence*, pages 15–21, Portland, OR, 1996. AAAI Press.

[8] Thomas G. Dietterich. Machine learning research: Four current directions. *AI Magazine*, 18:97–136, 1997.

[9] Thomas G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[10] W. Dillon and M. Goldstein. *Multivariate Analysis Methods and Applications*. John Wiley and Sons, 1984.

[11] L. Eshelman. *The CHC Adaptive Search Algorithm. How to have safe search when engaging in nontraditional genetic recombination*, pages 265–283. Morgan Kaufmann, 1991.

[12] B. Everitt. *Cluster Analysis*. Heinemann Educational Books, 1974.

[13] Yoav Freund and Robert Shapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2^{nd} European Conference on Computational Learning Theory*, pages 23–37, San Francisco, CA, 1995. Springer-Verlag.

[14] Yoav Freund and Robert Shapire. Experiments with a new boosting algorithm. In *Proceedings of the 13^{th} International Conference on Machine Learning*, pages 148–156, San Francisco, CA, 1996. Morgan Kaufmann.

[15] César Guerra-Salcedo and Darrell Whitley. Genetic search for feature subset selection: A comparison between CHC and GENESIS. In *Proceedings of the third annual Genetic Programming Conference*. Morgan Kaufmann, 1998.

[16] César Guerra-Salcedo and Darrell Whitley. Genetic approach to feature selection for ensemble creation. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 236–243, 1999.

[17] Simon Günter and Horst Bunke. Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In *Proceedings of the 8^{th} IWFHR*, pages 183–188, Niagara-on-the-lake, Canada, 2002.

[18] L. Hansen and P. Salamon. Neural network ensembles. *IEEE Transaction On Pattern Analysis And Machine Intelligence*, 12:993–1001, 1990.

[19] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2001.

[20] T. K. Ho. Random decision forests. In *Proc. of the 3rd Int'l Conference on Document Analysis and Recognition*, pages 278–282, Montreal, Canada, August 1995.

[21] Tin Kam Ho. The random subspace method for costnructing decision forests. *IEEE Transaction On Pattern Analysis And Machine Intelligence*, 20(8):832–844, August 1998.

[22] Anil Jain and Douglas Zongker. Feature selection: Evaluation, applicaiton, and small sample performance. *IEEE Transaction On Pattern Analysis And Machine Intelligence*, 19(2):153–158, 1997.

[23] George H. John, Ron Kohavi, and Karl Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the 11^{th} International Conference on Machine Learning*, pages 121–129, San Francisco, CA, 1994. Morgan Kaufmann.

[24] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 1995.

[25] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[26] Yuansong Liao and John Moody. Constructing heterogeneuos committees using input feature grouping: Application to economic forecasting. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[27] Huan Liu and lei Yu. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transaction On Knowledge and Data Engineering*, 17(4):491–502, April 2005.

[28] U. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[29] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection using multi-objective genetic algorithms for hand written digit recognition. In *Proceedings of the 16^{th} ICPR*, pages 568–571, 2002.

[30] L. S. Oliveira, R. Sabourin, F. Bortolozzi, and C. Y. Suen. Feature selection for ensembles: A hierarchical multi-objective genetic algorith approach. In *Proceedings of the 7^{th} ICDAR*, pages 676–680, 2003.

[31] David Opitz. Feature selection for ensembles. In *Proceedings of the 16^{th} International Conference on Artificial Intelligence*, pages 379–384, 1999.

[32] Nikunj C. Oza and Kagan Tumer. Input decimation ensembles: Decorrelation through dimensionality reduction. In *Proceedings of the 2^{nd} International Workshop on Multiple Classifier Systems*, pages 210–217, Cambridge, UK, 2001.

[33] P. Pudil, F. J. Ferri, J. Novovičová, and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, pages 279–283, 1994.

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

[35] Amanda J.C. Sharkey, editor. *Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems*. Springer-Verlag, 1999.

[36] Kagan Tumer and Nikunj C. Oza. Decimated input ensembles for improved generalization. In *Proceedings of the International Joint Conference on Nerual Networks*, Washington, DC, 1999.

[37] Douglas Zongker and Anil Jain. Algorithms for feature selection: An evaluation. In *Proceedings of the International Conference on Pattern REcognition*, pages 18–22. IEEE, 1996.