Methods and Applications of Ensemble Learning: A Review

Anjali Bhavan

Undergraduate Student

Department of Mathematics, Delhi Technological University, New Delhi, India

Address: Delhi Technological University, Shahbad Daulatpur, Main Bawana Road, Delhi

Email address: anjalibhavan98@gmail.com

Telephone number: +91-9654647823

Swati Aggarwal*

Assistant Professor

Department of Computer Engineering, Netaji Subhash Institute of Technology, New Delhi, India

Address: Netaji Subhash Institute of Technology, Sector – 3, Dwarka, Delhi

Email address: swati1178@gmail.com

Telephone number: +91-9717995716

**Corresponding author*

Abstract Ensemble learning is the practice of combining multiple estimators under certain rules for much better predictions and enhanced accuracy and other metrics on a variety of problems in data science and machine learning. This branch of machine learning is fast gaining recognition as a competent solution for many challenging problems in the field of machine learning due to its efficiency and performance on a wide range of datasets. This paper reviews the research in ensemble learning, as well as the principles, methods and applications of ensemble learning in various areas. It takes a look at the origins of ensemble learning, the problem of ensemble diversity (particularly diversity in classifier ensembles), the various kinds of ensembles, some highly popular ensemble algorithms such as AdaBoost and Random Forest and their applications and future research areas. We also compare the various ensemble methods on multiple datasets sourced from the UCI Machine Learning Repository, and report the results.

Keywords Ensemble learning, machine learning, bagging, boosting, stacking, classifiers

Introduction

Ensemble learning is a field that has been generating considerable interest in recent years, with the development of various algorithms and libraries containing several popular ensemble algorithms and their variants. The basic idea behind ensemble learning is to use multiple estimators and combine their analysis and predictions to give a final prediction. Hypotheses generated using ensembles made of diverse base estimators have been demonstrated to be far superior to single hypotheses [1]. Quinlan [2] conducted trials over a diverse dataset collection and demonstrated bagging and boosting ensembles as performing noticeably better than the standard versions. Particularly, boosting has been shown to be a considerably better performer than several other ensembles and single estimators, as in the case of optical character recognition [3].

Ensemble methods have been used in a wide variety of areas including feature selection [4] and improvement of generalization error of neural networks using ensembles of neural networks [5]. Some real world applications include crude oil price forecast [6], bankruptcy prediction [7], human activity recognition [8] and credit forecasting [9]. Ensembles are also being used for image deconvolution [10] and for training Hidden Markov Models [11]. AdaBoost [12], one of the most eminent algorithms in this field has been used for improving the performance of a strong estimator such as a neural network for the task of online character recognition, with excellent results [13].

This paper attempts to build on these works by providing a comprehensive review of the research and development of algorithms and systems of ensemble learning, and of the various types of ensemble methods currently prevalent in various fields.

Diversity in Ensembles

The subject of diversity of the base estimators constituting an ensemble has been studied extensively. It has been observed in several experiments and reviews that an ensemble can be more accurate than its base estimators only if the individual estimators are diverse, and that diversity plays a significant role in the performance of the ensemble on the concerned dataset [14]. The reason is that if each estimator makes different errors, then a rule-based combination of these estimators can reduce the total error [15]. Perrone and Cooper [16] demonstrated that the results of their experiments are affected by the distinctness of the neural networks constituting their ensembles.

Methods for quantifying a measure for diversity and generating diverse ensembles have been studied extensively. Quantifying diversity for classifier ensembles continues to be analysed, and does not have as much mature literature supporting it as that for regression ensembles. Diversity for (linearly weighted) regression ensembles has been quantified by two methods: Ambiguity decomposition [17] and bias-variance decomposition [18]. Ambiguity decomposition demonstrates that the error of a convex-combined ensemble will be less than or equal to the average error of its base estimators, while bias-variance decomposition breaks the ensemble generalization error into bias and variance terms [19].

Brown, Wyatt, Harris and Yao [19] provide a comprehensive review of the research in quantification of diversity, particularly in classifier ensembles. They note the works of Turner and Ghosh [20] in the field. Turner and Ghosh [20] demonstrated through their experiments the benefits and disadvantages of reduced correlation among the base estimators of a classifier ensemble – that while diverse base estimators can enhance the performance of the ensemble, the performance of the base estimators themselves often suffers due to them being trained on a subset of the training data. Particularly, if the amount of data available is limited, the cost of training an ensemble of classifiers may far outweigh the performance gains obtained.

Another prominent researcher in this field is Kuncheva [21], who explored the significance and measure of diversity in classifier ensembles by drawing analogies from biology and other life sciences. Sharkey and Sharkey [22] gave a four-layer hierarchy of ensemble diversity, which could be more descriptive if it incorporated the proportion of data for which the ensemble performs at the given levels [19]. A method for deriving the measure of diversity in ensembles was proposed by Brown and Kuncheva [23] as through two factors: the choices of combiner function and the error function. Cunningham and Carney [24] demonstrated that any work with classifier ensembles must have a strong focus on the diversity of its base classifiers, and that entropy is a useful measure of the diversity of classifier ensembles.

Methods for *achieving* diversity have also been studied. Merz and Pazzini [25] proposed a method to solve this problem of multi-collinearity among base estimators while combining them into ensembles for regression problems. Melville and Mooney [26] proposed a method for creating diverse ensembles by using artificially generated training examples to directly construct diverse hypotheses. ADDEMUP (Accurate anD Diverse Ensemble-Maker giving United Predictions), an algorithm proposed by Opitz and Shavlik [27] generates ensembles of neural networks keeping in mind the factors of accuracy and diversity. Rosen [28] described a method for decorrelation network training in ensemble neural networks, which produces diverse ensembles of networks. This method particularly works in the case of limited availability of training data, a problem highlighted in [20].

To summarize the work in the subject area of ensemble diversity - it has been demonstrated that diversity is a critical factor in determining ensemble performance, and that ensembles with very similar base estimators generalize poorly and produce high errors on testing –

because they don’t make errors on different regions of the hypothesis space, a fact which would have enabled the model to learn better. Having established that diversity is indeed an important factor to keep in mind while building ensembles, it is important to quantify and define it so that methods could be devised for achieving good diversity in ensembles. Hence the following question must be answered: *how best to define diversity numerically?*

Mature literature exists for regression, but methods continue to be studied and analysed for classification diversity. Regression diversity is usually handled by two methods: ambiguity decomposition and bias-variance decomposition, which were originally devised for linearly weighted ensembles. The problem of quantifying diversity for classifier ensembles continues to be open-ended, and warrants further research and investigation.

History

The concept of using a group of multiple classifiers for prediction was first explored by Dasarathy and Sheela [29]. Hansen and Salamon [5] presented a highly important and excellent analysis of neural network ensembles, and showed that these ensembles can reduce generalization errors of neural networks. Schapire [30] introduced a method for converting weak learners (those whose predictive ability is only slightly better than that of a random predictor), a concept that would be further used in his seminal papers on boosting and the conception of AdaBoost [12, 31] with Freund [32].

Another important work in this field is that of Wolpert [33], who developed the concept of stacked generalization. Stacked generalization works by training on one part of the dataset and testing on other, so the biases and errors are learned through the other part and improved on. It aims to give the highest generalization accuracy for the algorithm (instead of learning accuracy). This ensemble method was further analysed by Ting and Witten [34], and its performance on various datasets demonstrated. Wolpert and Macready [35] proposed an algorithm that combined a bootstrap procedure and stacking and improved the performance of bagging on regression problems. Stacking has been studied for unsupervised learning algorithms [36, 37] and regression as well [38].

Dietterich [39, 40, 41] wrote some of the most prolific reviews on various aspects of ensemble learning. He delineated the various concepts behind ensemble learning, the prevalent methods for measuring diversity and constructing ensembles, and the possible directions research in this field could take. He also provided a comparison of boosting, bagging and randomization for the construction of decision tree ensembles [42], and demonstrated that in situations with a lot of classification noise, bagging outperforms both the methods.

Ensembles have been termed as multiple classifier systems in many works. Ho, Hull and Srihari [43] demonstrated that multiple classifier systems are an excellent approach for various pattern recognition problems. Fumera and Roli [44] presented a thorough analysis of linear combiners as combination rules for multiple classifier systems, while Roli, Giacinto and Vernazza [45] described several methods for designing multiple classifier systems based on the ‘overproduce and choose’ principle. Giacinto and Roli [46] further went on to describe automatic methods for constructing multiple classifier systems, and tested them on the problem of remote-sensing image data classification.

Arguably, the most influential works in the domain of classifier diversity have been written by Kuncheva [21], and these explore the length and breadth of ensemble methods [47, 48], particularly ensemble diversity [49] (especially for classifiers), measures for quantifying it and its properties and influence on ensemble performance [50, 51, 52]. This paper, however, mentions only a few of her seminal research papers. Maclin and Opitz [53] also performed several empirical studies and evaluations on bagging and boosting [54], including for neural networks [55].

Bagging ensembles, which shall be covered in the next section, have seen much research into their concepts, properties and implementations as well. Bagging was first introduced by Breiman [56] as a highly effective ensemble method which aggregated predictions from various trees trained on bootstrap samples.

Ho [57] introduced the random subspace method, which consists of training multiple decision trees by randomly selecting subsets of the feature vector i.e. in randomly selected subspaces. This method has the advantage of improving the generalization accuracy while also maintaining high accuracy on the training data.

From the discussion above one can infer that the majority of developments in ensemble learning occurred in the years 1990-2005. Several works have focused on comparing bagging and boosting, and while bagging has showed remarkable performance on several problems, boosting often outperforms it, at the cost of overfitting the problem. Many other algorithms and variants continue to be developed and tested on a variety of problems, and it remains to be seen what the next big development in this field of study is going to be.

Types of Ensembles

Several types of ensembles have been developed in the past years, and continue to be applied in various areas. We cover some of the most well-known ones in our paper.

1. ***Bagging***

Bagging stands for Bootstrap Aggregating, and was first introduced by Breiman [56]. In Breiman’s words, ‘Bagging goes towards making a silk purse out of a sow's ear, especially if the sow's ear is twitchy.’

Bagging comprises of making several bootstrap replicas of the training set and growing trees on them, then aggregating the trees’ predictions for the final model prediction. This has been proved to give highly accurate results on both classification and regression problems, at the cost of simplicity of structure.

Now how and why does bagging work? Many works have tried to give a sound theoretical basis for the success of bagging ensembles. Domingos [58] gave two hypotheses (based on Bayesian learning) for the same, and experiments confirmed the second hypothesis as the primary factor in the success of bagging: that bagging changes the prior to a more suitable region of learner space.

The basic algorithm of bagging is given as follows:

Algorithm for Bagging

```
•   Draw k bootstrap samples (that is, samples drawn with replacement) from the training
    dataset.
•   For each boot strap sample i:
            Build a classifier model m
•   Train the k classifiers on the bootstrap samples.
•   Report final predictions by majority voting in case of classification, and averaging in
    case of regression.
```

Bühlmann and Yu [59] analysed bagging and subagging [60], which is based on an alternate aggregation structure and stands for subsample aggregating, while Friedman and Hall [61] studied bagging and proposed models to characterize the performance of bagging. A visual representation of bagged decision trees was given by Rao and Potts [62], and helped make the understanding of bagging simpler by a simple visualization of the bagged decision boundary (in the case of low-dimensional spaces).

Many variants to bagging, such as subagging [60] and attribute bagging [63] have been developed. Attribute bagging improves classifier ensemble accuracy by using random subsets of features, and the accuracy is further enhanced by ranking the feature subsets by their accuracy in classification.

Bagging has also been applied for unsupervised learning [64], for a clustering approach to the problem of tumour classification. Ensembles of support vector machines trained with bagging have shown remarkable improvement over traditional SVM implementations [65]. Each SVM is trained using bootstrap samples of the training set, then the results aggregated using various rules such as majority voting.

The most important development in the subject of bagging is the introduction of the Random Forest [66] algorithm. Random forests consist of a collection of decision trees, each depending on a random vector sampled independently and with identical distribution for all the trees in the forest. Each tree casts a unique vote which goes into deciding the final predictions of the ensemble. The trees are grown on new training sets drawn with replacement from the original training set, and random feature selection is used.

One of the main advantages of random forests is that it tackles overfitting; it reduces variance considerably with the cost of a slight increase in bias. Segal [67] tested these properties on real-world and simulated datasets, and reported that random forests do indeed perform exceptionally, albeit with a bit of tuning. Random forests have indeed demonstrated excellent performance and improvement on generalization accuracy in several experiments, and have been tested on various problems such as remote sensing classification [68] and hyperspectral data classification [69]. Random forests have also been applied in unsupervised learning problems [70] owing to their natural dissimilarity measures.

With studies in bagging comes the method of random subspaces [57]. Random subspace method has been analysed and applied in tandem with research in bagging, for instance in the case of linear classifiers [71]. Random subspaces have been applied in various areas as well, such as face recognition [72] and fMRI classification [73].

There have been several works combining bagging with other ensembles as well. Kotsiantis [74] built an ensemble combining bagging, boosting and their methods and aggregated their predictions by voting, and demonstrated its superior performance compared to simple bagging or boosting ensembles. Similarly, bagging and random subspaces were combined and assessed by Panov and Džeroski [75], and the ensemble performed just as well as random forests, with the added advantage of not requiring randomization.

2. ***Boosting***

Boosting works by training each base estimator to improve on the mis-classifications of the previous estimators, and finally gives a sound prediction that is often more accurate than bagging. What boosting does is use weak learners (those with classification accuracy slightly better than a random classifier) and convert them to a strong ensemble which gives good results.

The concept of boosting was first introduced by Schapire as an algorithm in the distribution free or probably approximately correct (PAC) learning model [30]. In a PAC model, the learner tries to find a hypothesis for accurate predictions based on randomly selected examples of the underlying concept. And weak learners are those whose predictions are only slightly better than random guessing, which is what gives rise to the question – is weak learnability equivalent to strong learnability? This question (called the hypothesis boosting problem) was left for interpretation by Kearns and Valiant [76] and answered by Schapire in his paper: could the hypotheses of a weak learner be boosted and proved equivalent to strong ones? Schapire presented a solution that forced the weak learner to focus on the 'difficult' parts of the problem and work on them recursively.

Freund [32] presented an improvement over this algorithm by proposing two variants: one to boost the algorithm by finding a hypothesis consistent with a large quantity of training samples, and the other to use filtering for boosting (selecting a few training examples as and when they're generated and reject the others). Filtering, being a general improvisation over PAC learners, provides upper bounds on the dependence between time and space complexities and the overall accuracy of the algorithm. The paper also demonstrated the usage of majority vote for boosting learning algorithms.

After these papers, Freund and Schapire then finally developed AdaBoost [12] and improved on it [31]. AdaBoost has been one of the most successful algorithms ever developed in the field of machine learning, and has sprung numerous research initiatives, variants and improvements on its structure and concept. For instance, Friedman, Hastie and Tibshirani [77] suggested that the AdaBoost algorithm proposed by Freund and Schapire be called Discrete AdaBoost (because it provides discrete output), and proposed Real AdaBoost for real valued outputs. They also proposed that AdaBoost be seen as a method for fitting an additive model and thus developed LogitBoost, a model that came on applying the cost functional of logistic regression to the additive model – that is, it optimizes the log-likelihood. A similar algorithm named GentleBoost was proposed as well. The algorithm for AdaBoost, as given by Freund and Schapire, is given below.

Algorithm for AdaBoost

Given: $(x_1, y_1), \ldots (x_m, y_m)$ where $x_i \in X, y_i \in Y = \{-1, 1\}$

Initialize $D_1(i) = 1/m$

For $t = 1 \ldots T$:

- Train base estimator using distribution D_t.
- Get base estimator $h_t: X \to R$.
- Choose $\alpha_t \in R$.
- Update:

$$D_{t+1}(i) = \frac{D_t(i)\exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Where Z_t is a normalization factor.

- Final output:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x))$$

Boosting algorithms using decision trees or stumps are usually difficult to interpret and analyse, so simpler structures go a long way in improving the algorithms further. Freund and Mason [78] proposed alternating decision trees as a new classifier rule that is much easier to interpret, and also demonstrated boosting procedures on them.

Freund [79] developed BrownBoost, a boost-by-majority algorithm with adaptive abilities as in AdaBoost, which derived from the boosting by majority concept proposed by him previously [32].

It was shown by Rätsch, Onoda and Müller [80] that AdaBoost tends to overfit; they then went on to propose an improvisation over AdaBoost, called AdaBoost$_{\text{reg}}$ that avoided the problem of overfitting and thus demonstrated enhanced performance.

Several such variations and improvisations on AdaBoost have come into being; Vezhnevets and Vezhnevets [81] proposed Modest AdaBoost, an algorithm which produces lesser generalization error at the cost of higher training error compared to (Discrete) AdaBoost, GentleBoost and Real AdaBoost. Zhu, Zou, Rosset and Hastie [82] introduced AdaBoost for multiple classes, while Li, Wang and Sung [83] demonstrated that support vector machines can be used as base classifiers for AdaBoost, contrary to popular notion that AdaBoost can work only for weak learners like decision trees and stumps. Some other developments include log-loss Boost [84], MAdaboost [85] and RobustBoost [86]. Many other variants continue to be developed to this day.

A comprehensive review of boosting algorithms can be found in [87].

3. ***Bayesian ensembles***

Studies in ensemble learning would be incomplete if they did not include ensembles built using Bayesian model averaging or combining. While far better ensemble methods (like boosting) have been introduced and are widely used, Bayesian learning theory [88] is still an important idea in ensemble learning and should be analysed.

Usually while selecting a model for a certain problem, an exercise is carried out which zeroes in on the best model, and the model is then used. The problem with this approach, however, is that it does not account for the uncertainty of the selected model itself.

Bayesian model averaging overcomes this problem of not accounting for model uncertainty. It samples hypotheses in the hypothesis space, then combines them using Bayes' law. It is often considered similar to Bayesian model combination, but the latter is an algorithmic improvement over the former which, though more computationally expensive, yields far better results. Bayesian model combination samples from the space of possible ensembles (model combinations) instead of sampling individual models. Methods have been demonstrated for converting Bayesian model averaging to Bayesian model combination [89], and Bayesian model averaging for linear regression has also been studied [90].

Problems associated with Bayesian model averaging (which are addressed by the simpler, more effective algorithms of boosting and bagging) are the reason why it is used very sparingly as an ensemble method, the chief problem being its tendency to produce ensembles that overfit the data. Domingos [91] studied Bayesian model averaging to determine if it reduces or avoids overfitting, but reported that it doesn't, as an outcome of which it demonstrates higher error rates than bagged ensembles.

Some excellent reviews of Bayesian ensemble methods can be found in [92] and [93].

4. ***Stacked generalization***

Stacked generalization, as noted above, works by figuring out the biases of the estimator(s) with respect to the given training set. A stacked ensemble has two (or more, in recent times) levels – the first level (called level 0) consists of the base estimators which are trained on the dataset and produce predictions, which are fed into the second level (level 1) called the meta-estimator for the final predictions from the ensemble.

Ting and Witten [94] explored stacked generalization and the factors that go into making good stacked ensembles: the kind of level 1 generalizer to choose for final predictions and the attributes that must go into level 0 generalizers as input. They demonstrated the usage of stacked generalization on various datasets with excellent results. Sesmero, Ledezma and Sanchis [95] also presented a thorough review of stacking and several instances where it is applied. Stacking is being studied and applied in several areas as well, like for spam filtering in emails [96] and document annotation [97].

Is Stacked Generalization better than Bagging and/or Boosting?

Is stacking predictive models better than boosting or bagging procedures? This is still an open question. Many researchers have applied bagging and boosting on a variety of application areas, but stacking still needs to be studied further.

For the case of ensemble diversity, bagging and boosting, while using identical base estimators, derive diversity by using different subsets of the data (whether it be features or samples). Stacked ensembles use different estimators for achieving diversity. In [94], Ting and Witten described the merits and demerits of stacking over bagging and boosting: while bagging and boosting both require a significant number of base estimators to function, stacking can produce competitive, if not better results even with just two or three base estimators (chosen appropriately). Stacking can work on a wide variety of datasets; however, it might not do as well on small datasets as compared to bagging or boosting.

To study these ideas we collected five datasets of varying sizes and natures from the UCI Machine Learning Repository [98] and trained various ensembles of bagging, boosting and stacking on them. Information about the datasets is given in the below table:

Dataset	Number of samples	Number of features
Immunotherapy	90	8
Fertility	100	10
Banknote authentication	1372	5
HTRU2	17898	9
Contraceptive methods	1473	9

Table 1: Datasets information

Procedure: We used the classifiers in the Scikit-learn library [99] for our analysis, and used Mlxtend [100] for the stacking classifier.

The training procedure was as follows: within the cross validation loop (ten-fold), we fitted the ensembles on nine folds and tested on the tenth fold (all folds created randomly). Accuracy was chosen as the evaluation metric, and the average accuracy of all ten folds was returned.

For bagging we used the BaggingClassifier provided in Scikit-learn, and for boosting we used Gradient Tree Boosting (a generalization of tree-based boosting to multi-class and regression problems) and AdaBoost. For stacking we initially started with logistic regression, support vector machine (with linear kernel) and a decision tree as the base estimators with logistic regression as the meta-estimator, but tested other combinations and classifiers and reported the best-performing ensembles here.

Dataset name	Bagging	Boosting	Stacking
Immunotherapy	87.8%	Gradient Boosting: 89% AdaBoost: 89%	Logistic Regression+ Support Vector Classifier+ Decision Tree Classifier: 87.8%
Fertility	86%	Gradient Boosting:70% AdaBoost: 90%	Logistic Regression+ Support Vector Classifier: 90%
Banknote authentication	99.13%	Gradient Boosting:98.03% AdaBoost: 98.00%	Logistic Regression+ Support Vector Classifier+ Decision Tree Classifier: 98.99%
HTRU2	98.04%	Gradient Boosting: 97.94% AdaBoost: 97.71%	Logistic Regression+ Support Vector Classifier: 97.8%
Contraceptive methods	54.8%	Gradient Boosting: 68.32% AdaBoost: 64.42%	Support Vector Classifier+ Multi-Layer Perceptron: 54.6%

Table 2: Observations from ten-fold cross validation

Conclusion

We have attempted to provide a comprehensive review of various ensemble methods, their history and development, and the problem of diversity in ensembles, particularly those for classification. Stacking in particular hasn’t been studied as extensively as bagging or boosting, and it continues to be an area of interest. Possible avenues of further research could involve investigating methods to build efficient stacked ensembles, and the effects of having more than two levels in it. There is also a need for developing a sound method for assessing ensemble diversity (particularly classifier diversity) numerically, so it can be analysed and systematic methods drawn up to achieve it.

There is also one observation one can infer from all this: that most of ensemble learning has been built from working around decision trees and/or stumps (and neural networks, rather sparingly). Further research could investigate ensembles built with other estimators, and also their usage for tasks like speech and image recognition, problems which have traditionally been handled using deep learning networks. Could ensemble learning surpass deep learning in some (or several) areas?

In our experiments above, we observe that stacking performs at par with both bagging and boosting, if not better. The only dataset where stacking does not perform well compared to the others is the contraceptive methods dataset, and even there it performs almost identically to the bagging estimator. Further testing, more studies about stacking and its properties and ways to improve its ensembles need to be conducted; however, the results above do show that stacking indeed is an ensemble method at par with the others, and can produce good results with very few base classifiers.

Stacked generalization has its problems, though: the selection of the best hyperparameters for each model could take time, as could the training of multiple models and then training another model on top of them. One can also argue about the additional computational and space complexities. Countering these could be some way forward for further research in this field.

Ensemble methods can go a long way in several avenues, and contain immense potential for further research and development in the coming years. We have tried our best to include most of the work that has been done in this field; however, some omissions must happen, and we apologize for them.

References

[1] Hendrik Jacob van Veen, Le Nguyen The Dat, Armando Segnini. 2015. Kaggle Ensembling Guide. https://mlwave.com/kaggle-ensembling-guide/ [accessed 2018 Feb 6].

[2] Quinlan, J. R. Bagging, boosting, and C4. 5. In AAAI/IAAI, Vol. 1 (pp. 725-730). (1996, August).

[3] Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. Boosting and other ensemble methods. Neural Computation, 6(6), 1289-1301. (1994).

[4] Guan, D., Yuan, W., Lee, Y. K., Najeebullah, K., & Rasel, M. K. A review of ensemble learning based feature selection. *IETE Technical Review*, *31*(3), 190-198. (2014).

[5] Hansen, L. K., & Salamon, P. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, *12*(10), 993-1001. (1990).

[6] Yu, L., Wang, S., & Lai, K. K. Forecasting crude oil price with an EMD-based neural network ensemble learning paradigm. *Energy Economics*, *30*(5), 2623-2635. (2008).

[7] Tsai, C. F., & Wu, J. W. Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert systems with applications*, *34*(4), 2639-2649. (2008).

[8] Catal, C., Tufekci, S., Pirmit, E., & Kocabag, G. On the use of ensemble of classifiers for accelerometer-based activity recognition. *Applied Soft Computing*, *37*, 1018-1022. (2015).

[9] Wang, G., Hao, J., Ma, J., & Jiang, H. A comparative assessment of ensemble learning for credit scoring. Expert systems with applications, 38(1), 223-230. (2011).

[10] Miskin J., MacKay D.J.C. Ensemble Learning for Blind Image Separation and Deconvolution. In: Girolami M. (eds) Advances in Independent Component Analysis. Perspectives in Neural Computing. Springer, London. (2000).

[11] MacKay, D. J. *Ensemble learning for hidden Markov models* (pp. 362-378). Technical report, Cavendish Laboratory, University of Cambridge. (1997).

[12] Freund, Y., & Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences, 55(1), 119-139. (1997).

[13] Schwenk, H., & Bengio, Y. Adaboosting neural networks: Application to on-line character recognition. In International Conference on Artificial Neural Networks (pp. 967-972). Springer, Berlin, Heidelberg. (1997, October).

[14] Webb, G. I., & Zheng, Z. Multistrategy ensemble learning: Reducing error by combining ensemble learning techniques. IEEE Transactions on Knowledge and Data Engineering, 16(8), 980-991. (2004).

[15] Robi Polikar Ensemble learning. Scholarpedia, 4(1):2776. (2009)

[16] Perrone, M. P., & Cooper, L. N. When networks disagree: Ensemble methods for hybrid neural networks. In How We Learn; How We Remember: Toward An Understanding Of Brain And Neural Systems: Selected Papers of Leon N Cooper (pp. 342-358). (1995).

[17] Krogh, A., & Vedelsby, J. Neural network ensembles, cross validation, and active learning. In Advances in neural information processing systems (pp. 231-238). (1995).

[18] Ueda, N., & Nakano, R. Generalization error of ensemble estimators. In Neural Networks, 1996., IEEE International Conference on (Vol. 1, pp. 90-95). IEEE. (1996, June).

[19] Brown, G., Wyatt, J., Harris, R., & Yao, X. Diversity creation methods: a survey and categorisation. Information Fusion, 6(1), 5-20. (2005).

[20] K. Tumer, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connection Science 8 (3–4) 385–403. (1996)

[21] Kuncheva, L. I. That elusive diversity in classifier ensembles. In Iberian Conference on Pattern Recognition and Image Analysis (pp. 1126-1138). Springer, Berlin, Heidelberg. (2003, June).

[22] Sharkey, A. J., & Sharkey, N. E. Combining diverse neural nets. The Knowledge Engineering Review, 12(3), 231-247. (1997).

[23] Brown, G., & Kuncheva, L. I. "Good" and "bad" diversity in majority vote ensembles. In International Workshop on Multiple Classifier Systems (pp. 124-133). Springer, Berlin, Heidelberg. (2010, April).

[24] Cunningham, P., & Carney, J. Diversity versus quality in classification ensembles based on feature selection. In European Conference on Machine Learning (pp. 109-116). Springer, Berlin, Heidelberg. (2000, May).

[25] Merz, C. J., & Pazzani, M. J. Combining neural network regression estimates with regularized linear weights. In Advances in neural information processing systems (pp. 564-570). (1997).

[26] Melville, P., & Mooney, R. J. Constructing diverse classifier ensembles using artificial training examples. In IJCAI (Vol. 3, pp. 505-510). (2003, August).

[27] Opitz, D. W., & Shavlik, J. W. Generating accurate and diverse members of a neural-network ensemble. In Advances in neural information processing systems (pp. 535-541). (1996).

[28] Rosen, B. E. Ensemble learning using decorrelated neural networks. Connection science, 8(3-4), 373-384. (1996).

[29] Dasarathy, B. V., & Sheela, B. V. A composite classifier system design: Concepts and methodology. Proceedings of the IEEE, 67(5), 708-713. (1979).

[30] Schapire, R. E. The strength of weak learnability. Machine learning, 5(2), 197-227. (1990).

[31] Freund, Y., & Schapire, R. E. Experiments with a new boosting algorithm. In Icml (Vol. 96, pp. 148-156). (1996, July).

[32] Freund, Y. Boosting a weak learning algorithm by majority. Information and computation, 121(2), 256-285. (1995).

[33] Wolpert, D. H. Stacked generalization. Neural networks, 5(2), 241-259. (1992).

[34] Ting, K. M., & Witten, I. H. Issues in stacked generalization. Journal of artificial intelligence research, 10, 271-289. (1999).

[35] Wolpert, D., & Macready, W. G. Combining stacking with bagging to improve a learning algorithm. Santa Fe Institute, Technical Report. (1996).

[36] Smyth, P., & Wolpert, D. Stacked density estimation. In Advances in neural information processing systems (pp. 668-674). (1998).

[37] Smyth, P., & Wolpert, D. Linearly combining density estimators via stacking. Machine Learning, 36(1-2), 59-83. (1999).

[38] Breiman, L. Stacked regressions. Machine learning, 24(1), 49-64. (1996).

[39] Ditterrich, T. G. Machine learning research: four current directions. Artificial Intelligence Magzine, 4, 97-136. (1997).

[40] Dietterich, T. G. Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer, Berlin, Heidelberg. (2000, June).

[41] Dietterich, T. G. Ensemble learning. The handbook of brain theory and neural networks, 2, 110-125. (2002).

[42] Dietterich, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Machine learning, 40(2), 139-157. (2000).

[43] Ho, T. K., Hull, J. J., & Srihari, S. N. Decision combination in multiple classifier systems. IEEE transactions on pattern analysis and machine intelligence, 16(1), 66-75. (1994).

[44] Fumera, G., & Roli, F. A theoretical and experimental analysis of linear combiners for multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(6), 942-956. (2005).

[45] Roli, F., Giacinto, G., & Vernazza, G. Methods for designing multiple classifier systems. In International Workshop on Multiple Classifier Systems (pp. 78-87). Springer, Berlin, Heidelberg. (2001, July).

[46] Giacinto, G., & Roli, F. An approach to the automatic design of multiple classifier systems. Pattern recognition letters, 22(1), 25-33. (2001).

[47] Kuncheva, L. I. A theoretical study on six classifier fusion strategies. IEEE Transactions on pattern analysis and machine intelligence, 24(2), 281-286. (2002).

[48] Kuncheva, L. I. Combining pattern classifiers: methods and algorithms. John Wiley & Sons. (2004).

[49] Shipp, C. A., & Kuncheva, L. I. Relationships between combination methods and measures of diversity in combining classifiers. Information fusion, 3(2), 135-148. (2002).

[50] Kuncheva, L. I. Diversity in multiple classifier systems. (2005).

[51] Kuncheva, L. I., & Whitaker, C. J. Ten measures of diversity in classifier ensembles: limits for two classifiers. In Intelligent Sensor Processing (Ref. No. 2001/050), A DERA/IEE Workshop on (pp. 10-1). IET. (2001, February).

[52] Skurichina, M., Kuncheva, L. I., & Duin, R. P. Bagging and boosting for the nearest mean classifier: Effects of sample size on diversity and accuracy. In International Workshop on Multiple Classifier Systems (pp. 62-71). Springer, Berlin, Heidelberg. (2002, June).

[53] Opitz, D., & Maclin, R. Popular ensemble methods: An empirical study. Journal of artificial intelligence research, 11, 169-198. (1999).

[54] Maclin, R., & Opitz, D. An empirical evaluation of bagging and boosting. AAAI/IAAI, 1997, 546-551. (1997).

[55] Opitz, D. W., & Maclin, R. F. An empirical evaluation of bagging and boosting for artificial neural networks. In Neural Networks, 1997., International Conference on (Vol. 3, pp. 1401-1405). IEEE. (1997, June).

[56] Breiman, L. Bagging predictors. Machine learning, 24(2), 123-140. (1996).

[57] Ho, T. K. The random subspace method for constructing decision forests. IEEE transactions on pattern analysis and machine intelligence, 20(8), 832-844. (1998).

[58] Domingos, P. M. Why Does Bagging Work? A Bayesian Account and its Implications. In KDD (pp. 155-158). (1997, August).

[59] Bühlmann, P., & Yu, B. Analyzing bagging. The Annals of Statistics, 30(4), 927-961. (2002).

[60] Andonova, S., Elisseeff, A., Evgeniou, T., & Pontil, M. A simple algorithm for learning stable machines. In ECAI (pp. 513-517). (2002, July).

[61] Friedman, J. H., & Hall, P. On bagging and nonlinear estimation. Journal of statistical planning and inference, 137(3), 669-683. (2007).

[62] Rao, J. S., & Potts, W. J. Visualizing Bagged Decision Trees. In KDD (pp. 243-246). (1997, August).

[63] Bryll, R., Gutierrez-Osuna, R., & Quek, F. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern recognition, 36(6), 1291-1302. (2003).

[64] Dudoit, S., & Fridlyand, J. Bagging to improve the accuracy of a clustering procedure. Bioinformatics, 19(9), 1090-1099. (2003).

[65] Kim, H. C., Pang, S., Je, H. M., Kim, D., & Bang, S. Y. Support vector machine ensemble with bagging. In Pattern recognition with support vector machines (pp. 397-408). Springer, Berlin, Heidelberg. (2002).

[66] Breiman, L. Random forests. Machine learning, 45(1), 5-32. (2001).

[67] Segal, M. R. Machine learning benchmarks and random forest regression. (2004).

[68] Pal, M. Random forest classifier for remote sensing classification. International Journal of Remote Sensing, 26(1), 217-222. (2005).

[69] Ham, J., Chen, Y., Crawford, M. M., & Ghosh, J. Investigation of the random forest framework for classification of hyperspectral data. IEEE Transactions on Geoscience and Remote Sensing, 43(3), 492-501. (2005).

[70] Shi, T., & Horvath, S. Unsupervised learning with random forest predictors. Journal of Computational and Graphical Statistics, 15(1), 118-138. (2006).

[71] Skurichina, M., & Duin, R. P. Bagging, boosting and the random subspace method for linear classifiers. Pattern Analysis & Applications, 5(2), 121-135. (2002).

[72] Wang, X., & Tang, X. Random sampling for subspace face recognition. International Journal of Computer Vision, 70(1), 91-104. (2006).

[73] Kuncheva, L. I., Rodríguez, J. J., Plumpton, C. O., Linden, D. E., & Johnston, S. J. Random subspace ensembles for fMRI classification. IEEE transactions on medical imaging, 29(2), 531-542. (2010).

[74] Kotsiantis, S. Combining bagging, boosting, rotation forest and random subspace methods. Artificial Intelligence Review, 35(3), 223-240. (2011).

[75] Panov, P., & Džeroski, S. Combining bagging and random subspaces to create better ensembles. In International Symposium on Intelligent Data Analysis (pp. 118-129). Springer, Berlin, Heidelberg. (2007, September).

[76] Kearns, M. and Valiant, L.G. Cryptographic limitations on learning Boolean formulae and finite automata. Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing (pp. 433-444). New York, NY: ACM Press. (1989).

[77] Friedman, J., Hastie, T., & Tibshirani, R. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). The annals of statistics, 28(2), 337-407. (2000).

[78] Freund, Y., & Mason, L. The alternating decision tree learning algorithm. In icml (Vol. 99, pp. 124-133). (1999, June).

[79] Freund, Y. An adaptive version of the boost by majority algorithm. Machine learning, 43(3), 293-318. (2001).

[80] Rätsch, G., Onoda, T., & Müller, K. R. An improvement of AdaBoost to avoid overfitting. In Proc. of the Int. Conf. on Neural Information Processing. (1998).

[81] Vezhnevets, A., & Vezhnevets, V. Modest AdaBoost-teaching AdaBoost to generalize better. In Graphicon (Vol. 12, No. 5, pp. 987-997). (2005, September).

[82] Hastie, T., Rosset, S., Zhu, J., & Zou, H. Multi-class adaboost. Statistics and its Interface, 2(3), 349-360. (2009).

[83] Li, X., Wang, L., & Sung, E. AdaBoost with SVM-based component classifiers. Engineering Applications of Artificial Intelligence, 21(5), 785-795. (2008).

[84] Collins, M., Schapire, R. E., & Singer, Y. Logistic regression, AdaBoost and Bregman distances. Machine Learning, 48(1-3), 253-285. (2002).

[85] Domingo, C., & Watanabe, O. MadaBoost: A modification of AdaBoost. In COLT (pp. 180-189). (2000, June).

[86] Freund, Y. A more robust boosting algorithm. arXiv preprint arXiv:0905.2138. (2009).

[87] Ferreira, A. J., & Figueiredo, M. A. Boosting algorithms: A review of methods, theory, and applications. In Ensemble machine learning (pp. 35-85). Springer, Boston, MA. (2012).

[88] Bernardo, J. M., & Smith, A. F. Bayesian theory. 1994. John Willey and Sons. Valencia (España). (1994).

[89] Monteith, K., Carroll, J. L., Seppi, K., & Martinez, T. Turning Bayesian model averaging into Bayesian model combination. In Neural Networks (IJCNN), The 2011 International Joint Conference on (pp. 2657-2663). IEEE. (2011, July).

[90] Raftery, A. E., Madigan, D., & Hoeting, J. A. Bayesian model averaging for linear regression models. Journal of the American Statistical Association, 92(437), 179-191. (1997).

[91] Domingos, P. Bayesian averaging of classifiers and the overfitting problem. In ICML (Vol. 2000, pp. 223-230). (2000, June).

[92] Wasserman, L. Bayesian model selection and model averaging. Journal of mathematical psychology, 44(1), 92-107. (2000).

[93] Hoeting, J. A., Madigan, D., Raftery, A. E., & Volinsky, C. T. Bayesian model averaging. In Proceedings of the AAAI Workshop on Integrating Multiple Learned Models (Vol. 335, pp. 77-83). (1998, May).

[94] Ting, K. M., & Witten, I. H. Stacked Generalization: when does it work? (1997).

[95] Sesmero, M. P., Ledezma, A. I., & Sanchis, A. Generating ensembles of heterogeneous classifiers using stacked generalization. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 5(1), 21-34. (2015).

[96] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., & Stamatopoulos, P. Stacking classifiers for anti-spam filtering of e-mail. arXiv preprint cs/0106040. (2001).

[97] Chidlovskii, B. U.S. Patent No. 7,890,438. Washington, DC: U.S. Patent and Trademark Office. (2011).

[98] Dua, D. and Karra Taniskidou, E. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. (2017).

[99] "Scikit-learn: Machine Learning in Python", Pedregosa et al., JMLR 12, pp. 2825-2830. (2011).

[100] Sebastian Raschka, Reiichiro Nakano, James Bourbeau, Will McGinnis, Guillaume Poirier-Morency, Colin, … Adam Erickson. rasbt/mlxtend: Version 0.11.0 (Version v0.11.0). Zenodo. (2018, June 15). http://doi.org/10.5281/zenodo.1198892