Ensemble Classification and Regression – Recent Developments, Applications and Future Directions

Ye Ren•, *Student Member, IEEE,* Le Zhang•, *Student Member, IEEE,* and P. N. Suganthan*, *Fellow, IEEE*

***Abstract*—Ensemble methods use multiple models to get better performance. Ensemble methods have been used in multiple research fields such as computational intelligence, statistics and machine learning. This paper reviews traditional as well as state-of-the-art ensemble methods and thus can serve as an extensive summary for practitioners and beginners. The ensemble methods are categorized into conventional ensemble methods such as bagging, boosting and random forest, decomposition methods, negative correlation learning methods, multi-objective optimization based ensemble methods, fuzzy ensemble methods, multiple kernel learning ensemble methods and deep learning based ensemble methods. Variations, improvements and typical applications are discussed. Finally this paper gives some recommendations for future research directions.**

***Index Terms*—Ensemble Classification, Ensemble Regression, Stacking, Multiple Kernel Learning, Deep Learning**

I. Introduction

Classification aims to identify the discrete category of a new observation by studying a training set of data:

$$y_c = f_c(\mathbf{x}, \boldsymbol{\theta}_c),\ y_c \in \mathcal{Z} \tag{1}$$

where $\mathbf{x}$ is the new observation as a feature vector, y_c is the category that the new observation belongs to, $f_c(\cdot)$ is the trained classification function, $\boldsymbol{\theta}_c$ is the classification function's parameter set. $\mathcal{Z}$ is the set of class labels.

Regression, instead of discrete categories, deals with continuous decisions:

$$y_r = f_r(\mathbf{x}, \boldsymbol{\theta}_r),\ y_r \in \mathcal{R} \tag{2}$$

where $\mathbf{x}$ is the new observation vector, y_r is the output, $f_r(\cdot)$ is the regression function and $\boldsymbol{\theta}_r$ is the regression function's parameter set.

The main idea behind the ensemble methodology is to aggregate multiple weighted models to obtain a combined model that outperforms every single model in it. Dietterich [1] explained three fundamental reasons for the success of ensemble methods: statistical, computational and representational. In addition, bias-variance decomposition [2] and strength-correlation [3] also explain why ensemble methods work.

Given the theoretical justifications behind ensemble methods, it is not surprising that a vast number of ensemble methods are available in classification, regression and optimization fields. This paper provides an introductory yet extensive review on the conventional as well as state-of-the-art ensemble methods such as multiple kernel learning and deep learning.

There are surveys on ensemble classification methods [4]–[8] and ensemble regression methods [9], [10] in the literature. However, there is no comprehensive review on ensemble classification and regression together. Some surveys focus on classification only [6], [7]. Some surveys focus on one particular machine learning method such as, conventional ensemble classification methods with neural networks [4] and ensemble methods applied to bio-informatics [8]. In [10], the survey focused on ensemble time series forecasting in renewable energy applications only. In [9], the authors had an extensive survey on ensemble regression but several methods were directly imported from classification without demonstrated applications to regression. Because classification and regression have similarities and differences, there are common and specific ensemble methods for classification and regression. It is therefore more suitable to discuss the ensemble methods for classification and regression together.

The paper is organized as follows: in Section II, we review the theories of ensemble methods; Ensemble methods for classification and regression are presented in Section III; In Section IV, we summarize the paper. We suggest future research directions in Section V.

II. Theory

The main theory behind ensemble methods is bias-variance-covariance decomposition. It offers theoretical justification for improved performance of an ensemble over its constituent base predictors. The key to ensemble methods is diversity, which includes data diversity, parameter diversity, structural diversity, multi-objective optimization and fuzzy methods.

A. Bias-Variance-Covariance Decomposition

Researchers initially investigated the theory behind ensemble methods by using regression problems. In the context of regression, there is a theoretical proof to show that a proper ensemble predictor can guarantee to have smaller squared error than the average squared error of the base predictors. The proof is based on ambiguity decomposition [11], [12]. However, ambiguity decomposition only applies to a single dataset with ensemble methods. For multiple datasets, bias-variance-covariance decomposition is introduced [12]–[15] and the equation is shown:

Y. Ren, L. Zhang and P. N. Suganthan are with the Department of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore e-mail: ({re0003ye, lzhang027, epnsugan}@ntu.edu.sg).

•The first two authors contributed equally to the work.

*Corresponding author: P. N. Suganthan(epnsugan@ntu.edu.sg).

TABLE I: Nomenclature

ABBREVIATION	DEFINITION
AdaBoost	ADAPTIVE BOOSTING
ANFIS	ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
ANN	ARTIFICIAL NEURAL NETWORK
ARCING	ADAPTIVELY RESAMPLE AND COMBINE
ARIMA	AUTO-REGRESSIVE INTEGRATED MOVING AVERAGE
ARTMAP	PREDICTIVE ADAPTIVE RESONANCE THEORY
Bagging	BOOTSTRAP AGGREGATION
CNN	CONVOLUTIONAL NEURAL NETWORK
DENFIS	DYNAMIC EVOLVING NEURAL-FUZZY INFERENCE SYSTEM
DIVACE	DIVERSE AND ACCURATE ENSEMBLE LEARNING ALGORITHM
DNN	DEEP NEURAL NETWORK
EMD	EMPIRICAL MODE DECOMPOSITION
GLM	GENERALIZED LINEAR MODELS
IMF	INTRINSIC MODE FUNCTION
KNN	K NEAREST NEIGHBOR
LR	LINEAR REGRESSION
LS-SVR	LEAST SQUARE SUPPORT VECTOR REGRESSION
MKL	MULTIPLE KERNEL LEARNING
MLMKL	MULTI-LAYER MULTIPLE KERNEL LEARNING
MLP	MULTIPLE LAYER PERCEPTRON
MPANN	MEMETIC pARETO ARTIFICIAL NEURAL NETWORK
MPSVM	MULTI-SURFACE PROXIMAL SUPPORT VECTOR MACHINE
NCL	NEGATIVE CORRELATION LEARNING
PCA	PRINCIPAL COMPONENT ANALYSIS
PSO	PARTICLE SWARM OPTIMIZATION
QCQP	QUADRATICALLY CONSTRAINED QUADRATIC PROGRAM
SMO	SEQUENTIAL MINIMAL OPTIMIZATION
SVM	SUPPORT VECTOR MACHINE
SVR	SUPPORT VECTOR REGRESSION
RBFNN	RADIAL BASIS FUNCTION NEURAL NETWORK
RNN	RECURRENT NEURAL NETWORK
RT	REGRESSION TREE
RVFL	RANDOM VECTOR FUNCTIONAL LINK

$$
\begin{aligned}
&E[\overline{f}-t]^2 = bias^2 + \frac{1}{M}var + (1-\frac{1}{M})covar \\
&bias = \frac{1}{M}\sum_i (E[f_i]-t) \\
&var = \frac{1}{M}\sum_i E[f_i - E[f_i]]^2 \\
&covar = \frac{1}{M(M-1)}\sum_i \sum_{j\neq i} E[f_i - E[f_i]](f_j - E[f_j])
\end{aligned}
\tag{3}
$$

where t is the target and f_i is the output from each model and M is the size of ensemble. The error is composed of the average bias (which measures the average difference between the prediction of the base learner and the desired output), plus a term involving their average variance (which measures the average variability of the base learners), and the third term involving their average pairwise covariance term (which measures the average pairwise difference of different base learners).

There exist several theoretical insights about the soundness of using ensemble methods such as strength-correlation [3], stochastic discrimination [16] and margin theory [17]. They have been shown to be equivalent to bias-variance-covariance decomposition [18].

From the equation, we can see the term $covar$ can be negative, which may decrease the expected loss of the ensemble while leaving $bias$ and var unchanged. Beside the $covar$, the number of models also plays an important role. As it increases, the proportion of the variance in the overall loss vanishes whereas the importance of the covariance increases. Overall, this decomposition shows that if we are able to design low-correlated individual learners, we can expect an increase in performance.

The categorical nature of discrete class labels prevents the direct application of the above decomposition of error to classification tasks. Fortunately, a number of ways to decompose error into bias and variance terms in classification tasks have been proposed [2], [19]–[22]. Each of these definitions is able to provide some valuable insight into different aspects of a learning algorithm's performance.

Numerous research indicates that some ensemble methods (such as bagging based ensemble learning) can significantly

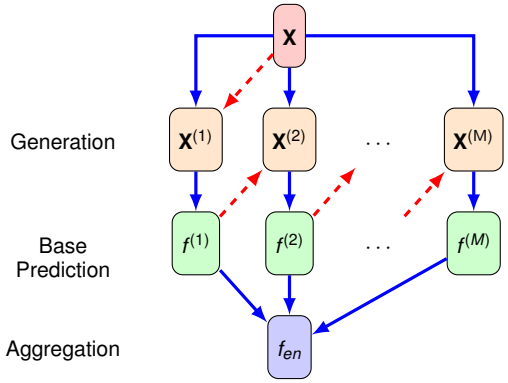

Fig. 1: Framework of Conventional Ensemble Methods, $\mathbf{X}$ is the original dataset, $\mathbf{X}^{(i)}, i \in \{1, \ldots M\}$ are the generated datasets, $f^{(i)}$ are the base predictors and f_{en} is the aggregation function. The dashed lines in the generation and base prediction parts denote boosting [27] related ensemble framework.

reduce the variance of the base classifiers [23], [24], while other ensemble methods (such as boosting type approach) can achieve significant bias and variance reduction [25], [26].

B. Diversity

Diversity is a key to ensemble methods and the importance of diversity was explained in [1], [28]. There are mainly three ways to create diversity in ensemble methods: data diversity, parameter diversity and structural diversity.

1) Data Diversity: Data diversity generates multiple datasets from the original dataset to train different predictors. The datasets should be different from each other so that there will be diverse decisions from the outputs of the trained predictors. For data diversity, bootstrap aggregation (Bagging) [29], Adaptive Boosting (AdaBoost) [27], random subspace [30], [31] and Random Forest [3] are commonly used.

In Fig. 1, the solid blue lines show the flow of bagging, random subspace and Random Forest which is realized in parallel. The dashed red lines show the flow of boosting and arching type of learning which is in a sequential fashion.

Output data variation is another direction to create data diversity. Instead of generating multiple datasets in the input space, multiple outputs are created to supervise the base predictors. A well known ensemble method of this kind is called 'output smearing' [32] which introduces diversity to the output space using random noise.

2) Parameter Diversity: Parameter diversity uses different parameter sets to generate different base predictors. Even with the same training dataset, the outputs of the base predictors may vary with different parameter sets. For example, in meteorology, ensemble forecasting [33] alters the initial conditions of the differential equations to model the meteorology. Multiple kernel learning (MKL) [34] is an ensemble method that combines the advantages of multiple kernels for classification or regression. The parameters of each kernel are tuned as well as the combining parameter. It is considered to be a parameter diversity enhancing method.

3) Structural Diversity: The structural diversity is induced by having different architectures or structures of the predictors. To construct an ensemble predictors, several base predictors can be used. These base predictors can vary in size, parameters and architecture and such kind of an ensemble predictor is also known as a heterogeneous ensemble [9].

4) Divide and Conquer: Divide and conquer [35] is a method that is often seen in time series forecasting applications. It divides the original dataset into a collection of datasets either in parallel or hierarchical, forming a collection of subtasks. Then predictors are applied to each sub-task and finally the outputs of all predictors are aggregated. The datasets in the sub-tasks usually have different characteristics and the predictors usually differ from each other. Thus, divide and conquer can possess multiple diversities.

5) Multi-Objective Optimization: Multi-objective approaches are natural for ensemble learning since they enforce the training process to yield a collection of optimal and diverse predictors instead of just a single predictor. The Pareto optimal predictors can be used to form an ensemble of predictors. The candidate predictors are usually generated by a population-based approach and these predictors are refined by the multi-objective optimization approach so that only Pareto optimal predictors are retained [36]. The above-mentioned approach not only encourages that the accurate predictors are selected in the ensemble framework but also that the predictors are distributed along the Pareto optimal front. This is in accordance with the bias-variance trade off because the training procedure converges to the Pareto front but the predictors are distributed as diversely as possible within the Pareto front [37].

6) Fuzzy Ensemble: A fuzzy classifier is the one that uses fuzzy sets or fuzzy logic in the course of its training or operation [38]. Fuzzy classification is a widely studied research topic in the literature due to its interpretability and ability to give soft labels [39]. Early work on fuzzy methods for voting in ensemble can be found in [40], [41]. Fuzzy ensembles are usually developed from conventional methods which rely on data, parameter or structural diversity. For example, in a fuzzy ensemble framework, conventional methods such as boosting and random forest are used to create diversity and the fuzzy logic is used to deal with imperfect data [42] or missing values [43].

III. Ensemble Classification and Regression Methods

A. Conventional Ensemble Methods

The conventional ensemble methods include bagging, boosting and stacking based methods. These methods have been well studied in recent years and applied widely in different applications. In addition to direct applications, variations and improvements have also been reported vastly in the literature.

Bagging is widely used in classification and regression [29]. For classification, there are improved versions of bagging such as online bagging (and boosting) [44], double-bagging [45], wagging [46]. For regression, we can find different bagging based applications: bagging neural network for Nosiheptide

fermentation product concentration prediction [47]; bagging support vector regression (SVR) for wireless sensor network target localization [48] and many others. Besides homogeneous predictors, heterogeneous (different types of predictors) ensemble prediction can be implemented via bagging to achieve both data diversity and structural diversity. In [49], a bagging predictor with the parallel usage of linear regression (LR) and regression tree (RT) algorithms was presented. The evaluation results on UCI datasets showed that the bagging LR+RT algorithm achieved better correlation coefficient than the bagging LR or the bagging RT algorithm alone.

Random subspace is also widely used in regression. It is similar to bagging but applies bootstrapping to the feature space instead of the sample space. A predictor was formed by combining bagging and random subspace together with local linear map, forming a regression ensemble [50]. A random subspace regression predictor was employed for near-infrared spectroscopic calibration of tobacco sample prediction [31] and the accuracy was better than other LR methods and was less sensitive to over-fitting even with small sample size.

AdaBoost is a sequential ensemble method [27] that was originally developed to enhance classification trees. There are several AdaBoost variants for classification such as: confidence-rated AdaBoost for multi-class classification [51]; soft margin AdaBoost with regularization [52]; Modest AdaBoost [53] and its improved modification with a weighting system [54]; SpatialBoost [55] that incorporates spatial reasoning; Adaptively resample and combine (Arcing) based AdaBoost or Arc-x4 [21]; Real AdaBoost [56]; and parallel implementing Ivoting [57].

Although AdaBoost was originally developed for classification problems, there are variants for regression such as: gradient boosting [58], [59]; big error margin boosting [60]; AdaBoost.R [56] (developed from AdaBoost.M2 [61]); AdaBoost.RT [62] (developed from AdaBoost.M1 [63]); and AdaBoost+ [64] (modified version of AdaBoost.RT).

Random forest [3] combines the concepts of bagging and random subspace. Breiman pointed out random forest is similar to AdaBoost even though AdaBoost is sequential whereas random forest is parallel. Lin and Jeon [65] showed that random forest is a kind of adaptive nearest neighbour. Random forest has gained popularity in high-dimensional and ill-posed problems [66], [67]. Researchers have investigated methodologies of feature and tree selection and best-split selection. There are improved versions of random forest reported in the literature: stratified random forest [68] with weighted feature sampling; Instance based random forest [69] for better tree selection; Extremely randomized trees [70] with randomly generated threshold for randomly selected features and its extended versions such as: oblique random forest with additional random combination effect [23] and oblique random forest with ridge regression in each node to obtain the 'best-split' [71]; Hybrid oblique random forest [72] with different transformation methods in each node and multi-surface proximal SVM (MPSVM) [73]–random forest [24] where MPSVM was employed to obtain the 'best-split' hyperplane; Rotation forest [74]–[79] with Principle Component Analysis (PCA) applied to the feature space; and a combination of AdaBoost and rotation forest called RotBoost [26].

Stacking, or stacked generalization [80] is a generalized method to bagging, random subspace or boosting because the aggregation stage is different yet more generalized. It uses another machine learning algorithm to estimate the weights of the outputs from each base predictor. The aggregation algorithm can also be a supervised learning algorithm. For example, SVM was used in the aggregation stage in [81]–[83]. Another advantage is that since the aggregation stage is supervised, the choice of base predictors are wider and it usually results in forming heterogeneous ensemble methods such as the methods reported in [84], [85].

B. Decomposition based Ensemble Methods

Time series forecasting is a popular research area under regression. A time series consists of sequential data with values associated with time intervals. Conventional ensemble methods can be applied to time series forecasting. However, in the literature, decomposition based ensemble methods are also popular for time series forecasting. A decomposition based ensemble methods can be further categorized into divide and-conquer and hierarchical ensemble methods.

A divide and-conquer ensemble methods decompose the original time series into a collection of time series from which the original time series can be completely reconstructed [35]. The purpose of decomposition is to make the complex original time series into a collection of simpler time series, and subsequently apply prediction algorithms on each decomposed time series. Finally the overall prediction is obtained by aggregating the predictions of the decomposed individual time series together.

Certain time series are related to seasons such as solar irradiation, rain fall, carbon dioxide concentration, etc. These time series can be decomposed by seasonal decomposition to reveal yearly, monthly or daily patterns and study on the residual data. In [86], hydro power data was studied with the original time series decomposed into trend, seasonal and irregular components. Least square SVR (LS-SVR), a modified version of SVR with equality constraints, was employed to predict on each component. Then the predictions were aggregated by another LS-SVR. The seasonal decomposed ensemble method was better than the single LS-SVR method.

Wavelet transform is a commonly used time series decomposition algorithm. It decomposes the original time series into certain orthonormal sub series by looking at the time-frequency domain. Then some prediction algorithms such as neural networks or regression trees can be applied to the sub series. As the completeness of the sub series is satisfied, the prediction outputs of the sub series can be aggregated by summation to obtain the final prediction. In the literature, wavelet transforms were used with several machine learning algorithms for time series forecasting such as wavelet-auto-regressive integrated moving average (ARIMA) [87] and wavelet-particle swarm optimization (PSO)-adaptive neuro-fuzzy inference system (ANFIS) [88] for wind speed forecasting, wavelet-fuzzy predictive adaptive resonance theory (ARTMAP) [89] for wind power forecasting and wavelet-recurrent neural networks [90] for solar irradiance forecasting.

Further, in [88], the authors applied particle swarm optimization (PSO) to determine the optimal parameters of ANFIS in order to achieve better prediction results.

Empirical mode decomposition (EMD) is another time series decomposition algorithm. Unlike wavelet transform, EMD processes the time series in the time domain. It is based on the local characteristic time scale of the time series and the process is adaptive. EMD decomposes the original time series into certain orthogonal sub series called intrinsic mode functions (IMFs) and then each IMF can be modelled and predicted by a prediction algorithm. The final prediction is the summation of all predictions from the IMFs and the residue since completeness is satisfied [91].

In [92], an EMD-LS-SVR method was reported. The EMD-LS-SVR was evaluated with a wind speed time series. The authors applied RBF kernel LS-SVR on the first 5 IMFs and Polynomial kernel LS-SVR on the 6th IMF and the residue. The results showed a better performance of the EMD-LS-SVR than LS-SVR and EMD-Regular Least Square methods. Zhang *et al.* [93] developed a wind power forecasting method based on EMD, RBFNN (for high frequency IMFs) and LS-SVR (for low frequency IMFs). In addition, the input vectors were constructed by chaotic phase space reconstruction. The reported method achieved a relatively low error. But, this paper did not report a comparative study against other methods. Other EMD based ensemble regression methods include: EMD-gene expression programming (GEP) method for short term load forecasting [94] and EMD based ensemble methods for wind speed forecasting [35], [95]–[97].

The hierarchical ensemble regression method decomposes time series data hierarchically. It applies a predictor to the original time series and then applies another predictor to the residue from the first predictor.

In time series forecasting, hierarchical ensemble regression is usually structured with a combination of statistical regression and a computational intelligence based regression algorithm such as ARIMA with neural networks, ARIMA with SVR [98], [99]. An ARIMA-dynamic evolving neural-fuzzy inference system (DENFIS) model for wind speed forecasting was reported in [100]. The authors proposed to use ARIMA to forecast the linear portion of the wind speed and use DENFIS to forecast the non-linear portion. The proposed hierarchical ensemble model outperformed ARIMA or DENFIS model alone. A solar irradiance forecasting model was formed by a combination of ARIMA and neural networks in [99]. The model applied ARIMA to forecast the linear portion of the time series and applied neural networks to forecast the non-linear portion of the time series (residue after ARIMA forecasting). The hierarchical ARIMA-neural network model outperformed single ARIMA or neural network model.

However, ARIMA-SVR and ARIMA-neural network models for wind power forecasting were re-evaluated experimentally in [98]. The improvements were marginal compared with the ARIMA model though. This implies that the hierarchical ensemble regression may be data specific.

A set of expert systems that were trained on a dataset with pre-partitioned input spaces was reported in [101], [102]. In these two papers, a mixture of experts was associated with a cooperative co-evolutionary algorithm to optimize the input space decomposition to improve the performance, which is verified by evaluation on several datasets.

Instead of applying regression throughout the hierarchy, classification and regression methods can be applied at different hierarchical levels. A hierarchical ensemble predictor of this kind was proposed for predicting hurricane and rainfall activity [103]. The authors first applied a classification predictor to predict the phase and then applied a regression predictor to estimate the magnitude of that particular phase the system was in. With the hierarchical structure, the prediction of the multi-variate spatio-temporal time series became feasible. A hierarchical age estimation model was reported in [104]. It is also a hierarchical classification + regression structure. A classification algorithm first classified face images to different age groups and then a regression algorithm further estimated the age of each face image. The hierarchical model was more accurate than other published results.

C. Negative Correlation Learning based Ensemble Methods

Negative correlation learning (NCL) is a well-known ensemble learning/training algorithm [105]. It introduces strong diversity among base learners with the same training dataset for all base learners. NCL has a convenient way to balance the bias-variance-covariance trade-off. In [105], NCL was evaluated with several classification and regression datasets and the comparison was on the aggregation methods: simple averaging and winner-takes-all. For regression, NCL with simple averaging is the only feasible configuration and for classification, NCL with winner-takes-all outperformed simple averaging. NCL was also compared with other ensemble methods (EPNet, Evo-En-RLS, etc.) and NCL had the best overall performance.

In [106], NCL was employed for regression. Two neural networks were used as base learners: multiple layer perceptron (MLP) and radial basis function neural network (RBFNN). The NCL-MLP was compared with several ensemble and non-ensemble methods and it had better performance. The NCL-RBFNN method outperformed NCL-MLP method based on the evaluation datasets. The authors have extended NCL from algorithm level to framework level and claimed that it was suitable to be applied to ensembles of any non-linear regression predictors. Additionally, A random Vector Functional Link (RVFL) based ensemble with NCL regularization can be found in [107], where all output weights of the neural network can be derived analytically.

NCL for classification was reported in [108]. The authors enhanced the performance of an ANN classifier ensemble by using NCL. Results based on several datasets have shown that the reported NCL based ANN classifier had better generalization performance.

Ordinal regression is a problem of predicting rankings or ordering of patterns, which has shared properties of both classification and regression. In [109], NCL was employed for ordinal regression. The authors proposed two threshold methods for an NCL based neural network ensemble predictor. The first threshold method was fixed and the second threshold

method was adaptive during training. Compared with other algorithms such as ORBoost, oNN, etc., the proposed NCL based ensemble method achieved competitive generalization performance.

D. Multi-Objective Optimization based Ensemble Methods

Multi-objective optimization uses state-of-the-art optimization algorithms such as evolutionary algorithms to find the Pareto front of the optimal predictors, thereby offering an ensemble learning framework. In [110], the authors used multi-objective evolutionary algorithm to solve the problem of the ANN's regularization terms. They found that multi-objective optimization can result in a sound set of ANNs thus forming an ANN-based ensemble predictor with a diverse ensemble, yet good performance. A well-tested work in this area is called Diverse and Accurate Ensemble Learning Algorithm (DIVACE) [36], [111], [112]. DIVACE originates from memetic Pareto ANN (MPANN) [113], [114] employing Pareto differential evolution [115], and NCL algorithm. Diversity was treated as a separate quantitative objective by NCL. Similar ideas can also be found in [116]. Based on the same multi-objective formulation, a multi-level evolutionary framework for the construction of hybrid ensembles was also developed in [117].

In [118], an explicit treatment of both diversity and accuracy within an evolutionary setup for constructing ensembles was also carried out. The proposed method, ADDEMUP, works by first creating an initial population followed by genetic operators to continually create new networks. During the training procedure, it keeps a set of networks that are as accurate as possible while disagreeing with each other as much as possible. In [119] the authors proposed the use of an evolutionary multi-objective algorithm and Bayesian Automatic Relevance Determination to automatically design and train an ensemble where almost all the parameters of the ensemble were determined automatically. The multi-objective evaluation of the fitness of the networks favors those networks with lower error rate as well as fewer features. Multi-objective regularized negative correlation learning (MRNCL) was introduced in [120], where the authors proposed a multi-objective regularized NCL algorithm which incorporated an additional regularization term for the ensemble and used the evolutionary multi-objective algorithm to design ensembles. Other recent developments of multi-objective optimization based ensembles on classification are found in [121]–[127].

Multi-objective optimization based ensemble methods can also be applied to regression. A software effort estimation based on multi-objective ensemble was reported in [128]. In [129], [130], a multi-objective evolutionary algorithm was applied together with a recurrent neural network (RNN) for time series forecasting. Another ensemble time series forecasting method based on multi-objective optimization was reported in [131].

E. Fuzzy Ensemble Methods

A fuzzy ensemble methods incorporate fuzzy logic into the ensemble learning framework to enhance performance. The advantages of some fuzzy combination methods are addressed in [132], [133].

A fuzzy random forest was proposed in [42] to combine the robustness of multiple classifier systems, the power of the randomness to increase the diversity of the trees, and the flexibility of fuzzy logic and fuzzy sets for imperfect data management. Various combination methods to obtain the final decision of the multiple classifier system were proposed and compared. The weighted combination based on membership function gave better results. A method to determine fuzzy integral density based on membership matrix was introduced in [134] to reduce subjective factors in building a fuzzy classifier and improve performance of the classification system. A fuzzy ensemble on data with missing values was addressed in [43], where logical neuro-fuzzy systems was integrated into the AdaBoost ensemble method. A fuzzy ensemble with high dimensional data was addressed in [135].

In some cases, it is not reasonable to classify the samples into a single label or multiple labels. For example, an experienced herb doctor can offer several diagnoses with different confidences according to the symptoms. In this case, a fuzzy ensemble can also be employed to fit the ranking of confidences of fuzzy classes [136]. The authors showed that the ranking of output confidences could fit the ranking of real confidences well, and the fitting error would be reduced while the number of the weak classifiers increases.

Fuzzy logic can be combined with multi-objective optimization to form an ensemble method as well. A multi-classifier coding scheme and an entropy-based diversity criterion for evolutionary multi-objective optimization algorithms were proposed in [137] for the design of fuzzy ensemble classifiers. The use of evolutionary multi-objective optimization to construct an ensemble of fuzzy rule-based classifiers with high diversity is described in [138]. In [139], a multi-objective evolutionary hierarchical algorithm was proposed to obtain a non-dominated fuzzy rule classifier set with interpretability and diversity preservation. Moreover, a reduce-error based ensemble pruning method was utilized to decrease the size and enhance the accuracy of the combined fuzzy rule classifiers.

F. Multiple Kernel Learning based Ensemble Methods

In recent years, multiple kernel learning (MKL) has become a hot topic in machine learning and computational intelligence [34]. MKL combines an ensemble of different kernels which may correspond to using different notions of similarity or may be using information coming from multiple sources (different representations or different feature subsets). These trained ensembles of kernels are combined using some classification methods, among which the most commonly used is the support vector machine. MKL can also be used as base learners of ensemble methods [140], [141].

The selection of different kernel functions and the optimization of their corresponding parameters have been extensively studied. MKL is applied in this context by using an ensemble of kernel functions. Numerous research works have demonstrated the advantage of multiple kernels over a pre-defined

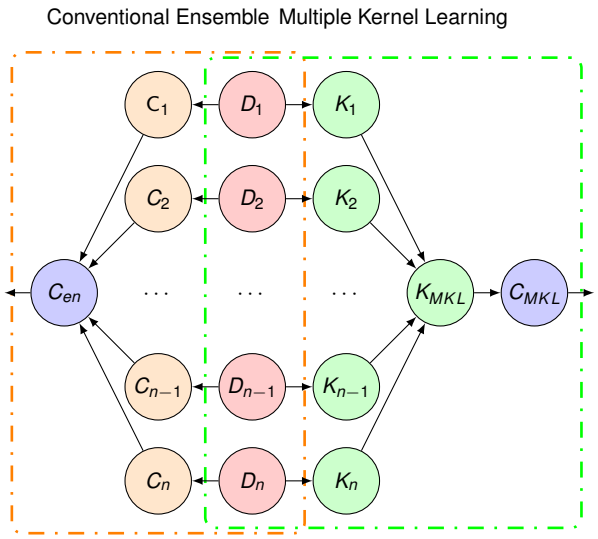

Fig. 2: Comparison of conventional ensemble methods and multiple kernel learning. C denotes a classifier, D denotes data and K denotes a kernel.

kernel function with optimized parameter values. This kind of kernel ensemble is defined as follows:

$$k_\eta(x_i, x_j) = f_\eta(k_m(x_i, x_j)_{m=1}^p) \tag{4}$$

where the key lies in the combination function f_η which can be a linear or nonlinear function. Here η is used to parameterize the pre-defined kernel function. Fig. 2 shows the difference between the MKL and conventional ensemble methods mentioned in Section III-A.

Generally speaking, there are two versions of MKL. In the first case, different kernels are used corresponding to different notions of similarity. The learning algorithm attempts to either find an optimal combination of kernels or pick one from them if using a specific kernel may be a source of bias. In the second case, we may use different versions of input (i.e, different representations possibly from different sources or modalities).

Linear combination methods are popular and have two basic versions: unweighted combination and weighted combination. Unweighted is similar to bagging which is discussed in Section III-A. In the weighted case, different evaluation criteria can be used to learn the optimal weights.

In [142], the weights of different kernels were associated with the alignment of the kernel to a pre-defined 'target' kernel. In their study, kernels with higher alignment to the target were proved to have better generalization ability. The authors proposed an eigen decomposition method to optimize the alignment of the combined kernel function. Lanckriet *et al.* [143] directly optimized the alignment of the combined kernel function with semi-definite programming . However, semi-definite programming has a very large computational complexity. Hence, it is intractable for a 'big data' problems. In [144], the authors solved this problem by rewriting it as a semi-infinite programming problem that can be efficiently solved by recycling the standard SVM solver. Based on the notion of alignment, [145] proposed a two-stage learning method for MKL. Cortes *et al.* [146] proposed the notion of local Rademacher complexity to design new algorithms for learning kernels which lead to a satisfactory result with a faster convergence rate. Moreover, in Orabona *et al.*'s work [147], faster convergence rate was achieved by stochastic gradient descent. Xu *et al.* [148] proposed a framework of soft margin MKL and demonstrated that the commonly used loss function could be readily incorporated into this framework. In [149], SpicyMKL which iteratively solved the smoothed minimization problems without solving SVM, LP, or QP internally was proposed. Experiments showed SpicyMKL was faster than other methods especially when the size of the kernel ensemble is large (several thousands). Kloft *et al.* [150] extended MKL to arbitrary norms to allow for robust kernel mixtures that generalize well. Experiments showed the advantage of this L_p norm of MKL over the conventional L_1 norm based sparse MKL method. Motivated by deep learning structure, in [151], the authors proposed a two-layer MKL which differed from the conventional 'shallow' MKL methods. In their work, the MLMKL offered higher flexibility through multiple feature mappings. In [152], the authors proposed to train the MKL with sequential minimal optimization (SMO) which is simple, easy to implement and adapt, and efficiently scales to large problems. 'Support Kernel Machine' based on the dual formulation of the quadratically constrained quadratic program (QCQP) as a second-order cone programming was proposed in [153]. This work also shows how to exploit the Moreau-Yosida regularization to yield a formulation which can be combined with SMO. In [154], Simple MKL was proposed where the MKL problem results in a smooth and convex optimization problem, which is actually equivalent to other MKL formulations available in the literature. In [155], the authors proposed MKL for joint feature maps which provided a convenient and principled way to employ MKL for solving multi-class problems.

There are other types of ensemble based kernel learning research in the literature. In [156], boosting was used to design a kernel which had a better generalization ability. In [157], the authors showed that each decision tree is actually a kernel. Then an MKL algorithm was employed to prune the decision tree ensemble.

G. Deep Learning based Ensemble Methods

Recently deep learning [158] has been a hot topic in computational intelligence research. In deep learning, deep structure which is composed of multiple layers of non-linear operations is able to learn high-level abstraction. Such high-level abstraction is a key factor leading to the success of many state-of-the-art systems in vision, language, and other AI-level tasks. Complex training algorithms combined with carefully chosen different types of parameters (e.g. learning rate, mini-batch size, number of epochs) may lead to deep neural networks (DNN) with high-performance. We can find that ensemble methods successfully boost the performance of DNN in various scenarios.

Convolutional Neural Networks (CNN) [159] have been successfully applied to solve many tasks such as digit, object

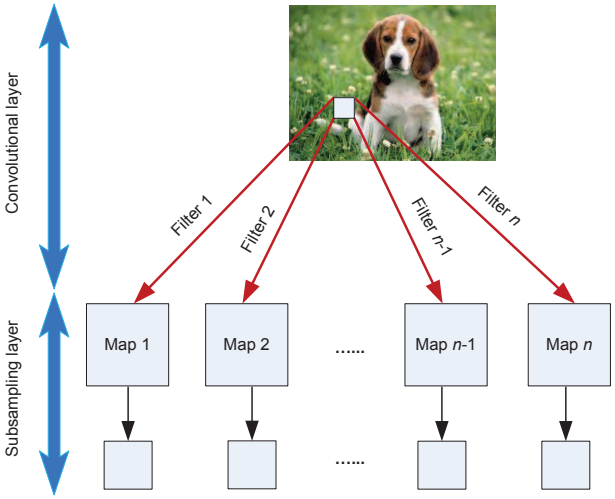

Fig. 3: Basic structure of a convolutional neural network. The output is the input of a pre-defined classifier.

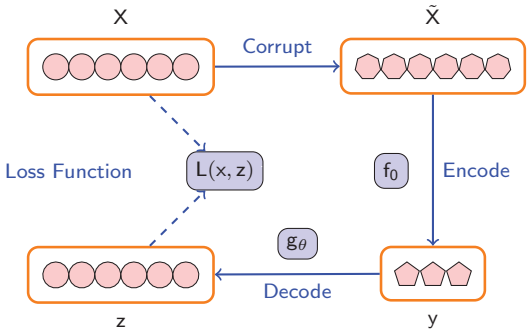

Fig. 4: Basic structure of a denoising autoencoder

and speech recognition. CNN combines three architectural ideas to incorporate shift, scale and distortion invariances: shared weights, sub-sampling and local receptive fields. A simple CNN is shown in Fig. 3.

We can easily stack this architecture into deep architectures by setting the output of one CNN to be the input of the next. CNN employs ensemble methods in the inner structure. Recently, researchers have successfully demonstrated the power of an ensemble of CNNs. In [160], [161], 'Multi-column Deep Neural Networks' were proposed where each 'column' is actually a CNN. The outputs of all columns were averaged. The proposed method improved state-of-the-art performance on several benchmark data sets.

Autoencoder [162] is also a popular building block of deep learning structure. An autoencoder can be decomposed into two parts: encoder and decoder. The encoder is a deterministic mapping that maps the input x to the hidden representation y through: $f_\Theta(x) = s(Wx + b)$ where $\Theta = \{W, b\}$, and s is some non-linear activation function such as the sigmoid. In the decoder, the hidden representation is then mapped back to reconstruct the input x. This mapping is achieved by $g'_\Theta(y) = s(W'y + b')$. Fig. 4 shows the structure of a denoising autoencoder. In a denoising autoencoder, firstly the input is corrupted by some noise and the autoencoder aims to reconstruct the 'clean' input.

One can easily generalize this basic denoising autoencoder to some deep structure by repeatedly mapping one hidden representation to another and then decoding each mapping in the decoder. In [163], the author proposed an ensemble of stacked sparse denoising autoencoders [164]. In that work, the weight of the output of each base stacked sparse denoising autoencoder (or each column) is optimized by a stand out network.

In [165], the author proposed an ensemble of deep SVM for image categorization. The first hidden layer of SVM was used for extracting latent variables $f(x|\Theta)_i$, where i is the index of the SVM in each layer and Θ is the parameter set for a particular SVM. In the second hidden layer, one SVM was used to approximate the target function using the extracted feature vector as input. One can easily generalize this structure to multiple layers. In [165], the problem was decomposed into several one-vs-all sub-problems. For each problem, a two-layer SVM was trained. Then the results were aggregated using a product rule [166]. An ensemble of deep SVM was combined with Spatial Pyramids in [167] resulting in improved performance.

It is common sense that a large feed-forward neural network trained on a small training set will typically have poor performance on test data. To tackle this problem, Hinton [168] proposed a method called 'dropout' to prevent co-adaptation of feature detectors. In this method, the key idea is to randomly, with some pre-defined probability, drop units (along with their connections) from a neural network during training. This prevents the units from co-adapting too much. Dropping units creates thinned networks during training. Dropout can be seen as an extreme form of bagging in which each model is trained on a single case and each parameter of the model is very strongly regularized by sharing it with the corresponding parameter in all the other models. During testing, all possible thinned networks were combined using an approximate model averaging procedure. The idea of dropout is not limited to feed-forward neural networks. It can be more generally applied to graphical models such as Boltzmann Machines. Random dropout gives big improvements on many benchmark tasks and sets new records for speech and object recognition. In [169], the author proposed 'adaptive dropout' where a stand out network was used to adaptively learn the probability of each node.

In [170], the DropConnect network was proposed for regularizing large fully-connected layers within a neural network. This can be regarded as a generalization of dropout. DropConnect can be regarded as a larger ensemble of deep neural networks than dropout. When training with Dropout, a randomly selected subset of activations is set to zero within each layer. DropConnect instead sets a randomly selected subset of weights within the network to zero. In the testing phase, DropConnect uses a sampling-based inference which is shown to be better than the mean-inference which is used in dropout. The author also gave some theoretical insight on why DropConnect regularizes the network.

Besides the methods mentioned above, we are witnessing a

rapid, revolutionary change in the computer vision community, mainly caused by ensembles of deep neural networks. Ensemble methods always achieve top performance in the 'Large Scale Visual Recognition Challenge' (ILSVRC) [171].

In the literature, there is little research reported on deep learning for regression [172], [173]. The paper [172] reported an energy load forecasting algorithm with deep neural networks. A deep feed forward neural network and a deep recurrent neural network were compared in the paper and the deep recurrent neural network performed better.

In [173], a deep SVM was introduced for regression. The lower-level layers of SVM were used for feature extraction only and the final layer of SVM was used for prediction. The results on ten regression datasets showed that the deep SVM outperformed SVM significantly.

Ensemble deep learning for regression was first introduced in [174] and this is currently the only one literature for deep learning ensemble regression. In this paper, an ensemble of deep belief networks (DBN) was introduced for regression as well as time series forecasting. The base predictors were DBN with different parameters. The aggregation algorithm was an SVR. Several time series data and regression data were used for evaluation and the proposed deep DBN had superior performance compared with other benchmark methods.

To summarize, deep learning can be employed as a base model in conventional ensemble learning approaches such as [160], [161], where the final prediction can be an average of all outputs of the base neural networks. More examples and improvements can be found in the top rank models in the 'Large Scale Visual Recognition Challenge' (ILSVRC) [171].Moreover, ensemble concepts can arise in a single deep learning model, such as the concatenation of the filter bank [159] to achieve over-complete representations, parameter or connection permutation in training process to mimic the bagging concept [168]–[170].

H. Ensemble Regression Converted to Ensemble Classification

Multi-class classification deals with discrete output space whereas regression deals with continuous output space. If the regression output does not require high resolution, it is feasible to approximate it with multiple discrete values, thereby resulting in a multi-class classification problem. Hence, a classification algorithm can be used to solve the regression problem.

In [175], the authors proposed an ensemble regression from classification conversion method called extreme randomized discretization (ERD) to convert continuous values to discrete values. The ERD grouped the continuous values into bins where the boundaries were created randomly in order to create ensembles. The authors evaluated the proposed method with several regression datasets and they concluded that the ensemble method outperformed the ensemble regression converted to classification method with equal-width discretization bins. They also found that a greater number of bins yielded better performance.

IV. Conclusion

This paper has reviewed the state-of-the-art on ensemble classification and ensemble regression. The theories relating to ensemble classification and regression have been discussed including bias-variance decomposition and the diversity issue. Conventional ensemble methods have been introduced and recent improvements to the conventional ensemble methods have also been discussed . State-of-the-art ensemble methods such as multi-objective optimization based ensembles, multiple kernel learning, and fuzzy ensembles have also been reviewed. The early stage development of ensemble incorporating deep learning has also been surveyed. Some divide-and-conquer based ensemble methods specialized for time series forecasting have also been presented.

The paper has also reviewed some methods that convert regression problems into multiple-class classification problems and that apply ensemble classification to regression problems.

V. Future Work

Although there are numerous recent ensemble methods for solving classification and regression problems reported in the literature, there is still ample room for improvement. There are several promising research directions for ensemble classification and regression as discussed below.

"Big Data" [176] has attracted considerable attention recently. It is worthy to investigate the benefits of ensemble learning for solving big data problems. We can also investigate the benefits of ensemble methods for solving other machine learning tasks such as clustering [177], [178].

There are learning algorithms which are unstable [13], [25], [179], especially randomized learning approaches such as random vector functional link (RVFL) neural network. According to a recent comprehensive evaluation [180], there is a large gap in performance between randomized learning algorithms and rank 1 methods (Random Forest). On the other hand, according to [25], highly unstable methods are naturally suitable for ensembles. That is, the variance of ensemble can be reduced significantly. Therefore, it is worthwhile to investigate the performance of ensemble methods with fast randomized base learners such as RVFL.

There are few ensemble classification and regression methods containing fuzzy systems as base predictors. Multi-objective optimization based ensemble classification and regression methods are also under-researched in the literature. Further research on fuzzy ensemble and multi-objective optimization based ensemble methods is recommended. A possible direction is to combine fuzzy systems and optimization algorithms because ensemble methods with fuzzy systems require much more parameter tuning. Hence, employing optimization algorithms will accelerate the process. For multi-objective optimization based ensemble methods, we would like to highlight the possibility of applying optimization to different base predictors other than feed-forward neural networks. We can also investigate the applicability of ensemble methods for evolutionary algorithms [181], [182].

In Section III-G, we reviewed some deep learning based ensemble classification and regression methods such as CNN

ensembles and deep SVM ensembles. Current deep learning approaches aim to generate deep representations for the data. However, it is common sense that deep neural networks are very difficult to train because of the vanishing gradient. Recent research remedies this by either initializing the network with some unsupervised training approach such as restricted Boltzmann machine or reducing the number of parameters by 'sharing weights' such as CNN. However, it remains an open question how well these approaches can regularize the network. Ensemble methods, which can be regarded as complementary to the current research, boost the performance of neural networks by reducing the variance significantly as we mentioned in Section II. In the future, researchers should further investigate deep learning based ensemble methods. Several questions may be addressed: Is it necessary to pre-train the 'base deep network' in an unsupervised manner in the ensemble? Is it possible to increase the diversity of the 'base deep network' without losing too much accuracy in the ensemble? How can one develop advanced ensemble methods which are suitable for deep learning?

Ensemble regressors for approximating fitness landscapes in the context of evolutionary and swarm algorithms are also under-researched [183], [184]. Moreover, deep learning ensembles are a future research direction. Although certain deep learning ensemble classification methods can be easily transformed/migrated to deal with regression problems, we need further development on deep learning ensemble regression algorithms. Deep RVFL ensembles are a promising approach due to their diversity and fast training.

Deep and complex models are much more difficult to train than shallow and simple models because they need a large training data set to tune a large number of parameters. Deep learning approaches are becoming more feasible because sufficiently large data sets are becoming available [185], [186]. However, there exist few efforts in the literature to investigate the performance of ensemble methods for big data. Hence, this is a another area where further research is required.

REFERENCES

[1] T. G. Dietterich, "Ensemble methods in machine learning," in *Multiple Classifier Systems*. Springer, 2000, pp. 1–15.

[2] R. Kohavi and D. H. Wolpert, "Bias plus variance decomposition for zero-one loss functions," in *Proc. International Conference on Machine Learning (ICML'96)*, 1996, pp. 275–283.

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

[4] Y. Zhao, J. Gao, and X. Yang, "A survey of neural network ensembles," in *Proc. International Conference on Neural Networks and Brain (ICNNB '05)*, vol. 1, Oct. 2005, pp. 438–442.

[5] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[6] D. Gopika and B. Azhagusundari, "An analysis on ensemble methods in classification tasks," *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 3, no. 7, pp. 7423–7427, 2014.

[7] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010. [Online]. Available: http://dx.doi.org/10.1007/s10462-009-9124-7

[8] P. Yang, Y. Hwa Yang, B. B. Zhou, and A. Y. Zomaya, "A review of ensemble methods in bioinformatics," *Current Bioinformatics*, vol. 5, no. 4, pp. 296–308, Dec. 2010.

[9] J. M. Moreira, C. Soares, A. M. Jorge, and J. F. de Sousa, "Ensemble approaches for regression: A survey," *ACM Computing Surveys*, vol. 45, no. 1, pp. 1–10, 2012.

[10] Y. Ren, P. N. Suganthan, and N. Srikanth, "Ensemble methods for wind and solar power forecasting: A state-of-the-art review," *Renewable Sustain. Energy Rev.*, vol. 50, pp. 82–91, Oct. 2015.

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

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

[13] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, no. 1, pp. 1–58, 1992.

[14] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *The Journal of Machine Learning Research*, vol. 6, pp. 1621–1650, 2005.

[15] P. Domingos, "A unified bias-variance decomposition," in *Proc. International Conference on Machine Learning (ICML'00)*, 2000, pp. 231–238.

[16] E. Kleinberg, "Stochastic discrimination," *Annals of Mathematics and Artificial intelligence*, vol. 1, no. 1, pp. 207–239, 1990.

[17] R. E. Schapire and Y. Freund, "Boosting the margin: A new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, pp. 322–330, 1998.

[18] V. Pisetta, "New insights into decision tree ensembles," Ph.D. dissertation, Lyon 2.

[19] E. B. Kong and T. G. Dietterich, "Error-correcting output coding corrects bias and variance." in *Proc. International Conference on Machine Learning (ICML'95)*, 1995, pp. 313–321.

[20] J. H. Friedman, "On bias, variance, 0/1–loss, and the curse-of-dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[21] L. Breiman, "Arcing classifier (with discussion and a rejoinder by the author)," *The Annals of Statistics*, vol. 26, no. 3, pp. 801–849, 1998.

[22] G. M. James, "Variance and bias for general loss functions," *Machine Learning*, vol. 51, no. 2, pp. 115–135, 2003.

[23] L. Zhang, Y. Ren, and P. N. Suganthan, "Towards generating random forest with extremely randomized trees," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN'14)*, Beijing, China, Jul. 2014.

[24] L. Zhang and P. N. Suganthan, "Oblique decision tree ensemble via multisurface proximal support vector machine," *IEEE Trans. Cybern.*, pp. 2168–2267, Nov. 2014.

[25] L. Breiman, "Bias, variance, and arcing classifiers," University of California, Berkeley, CA, Tech. Rep. 460, 1996.

[26] C.-X. Zhang and J.-S. Zhang, "RotBoost: A technique for combining rotation forest and adaboost," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1524–1536, 2008.

[27] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. International Conference on Machine Learning (ICML.96)*, vol. 96, 1996, pp. 148–156.

[28] E. K. Tang, P. N. Suganthan, and X. Yao, "An analysis of diversity measures," *Machine Learning*, vol. 65, no. 1, pp. 247–271, 2006.

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

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

[31] C. Tan, M. Li, and X. Qin, "Random subspace regression ensemble for near-infrared spectroscopic calibration of tobacco samples," *Analytical Sciences*, vol. 24, no. 5, pp. 647–654, 2008.

[32] L. Breiman, "Randomizing outputs to increase prediction accuracy," *Machine Learning*, vol. 40, no. 3, pp. 229–242, 2000.

[33] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–196, 1999.

[34] M. Gönen and E. Alpaydın, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[35] Y. Ren and P. N. Suganthan, "A comparative study of empirical mode decomposition based short-term wind speed forecasting methods," *IEEE Trans. Sustain. Energy*, vol. 6, no. 1, pp. 236–244, 2015.

[36] A. Chandra and X. Yao, "Multi-objective ensemble construction, learning and evolution," in *Proc. PPSN Workshop on Multi-objective Problem Solving from Nature (part of the 9th International Conference on Parallel Problem Solving from Nature: PPSN-IX)*. Citeseer, 2006, pp. 9–13.

[37] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[38] L. I. Kuncheva, *Fuzzy classifier design*. Springer Science & Business Media, 2000, vol. 49.

[39] ——, "Fuzzy classifiers," *Scholarpedia*, vol. 3, no. 1, p. 2925, 2008, revision 133818.

[40] H. Ishibuchi, T. Morisawa, and T. Nakashima, "Voting schemes for fuzzy-rule-based classification systems," in *Proc. IEEE International Conference on Fuzzy Systems (FuzzIEEE'96)*, vol. 1. IEEE, 1996, pp. 614–620.

[41] H. Ishibuchi, T. Nakashima, and T. Morisawa, "Voting in fuzzy rule-based systems for pattern classification problems," *Fuzzy sets and systems*, vol. 103, no. 2, pp. 223–238, 1999.

[42] P. Bonissone, J. M. Cadenas, M. C. Garrido, and R. A. Díaz-Valladares, "A fuzzy random forest," *International Journal of Approximate Reasoning*, vol. 51, no. 7, pp. 729–747, 2010.

[43] M. Korytkowski, R. Nowicki, R. Scherer, and L. Rutkowski, "Ensemble of rough-neuro-fuzzy systems for classification with missing features," in *Proc. IEEE International Conference on Fuzzy Systems (FuzzIEEE'08)*. IEEE, 2008, pp. 1745–1750.

[44] N. C. Oza, "Online bagging and boosting," in *Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC'05)*, vol. 3, 2005, pp. 2340–2345.

[45] T. Hothorn and B. Lausen, "Double-bagging: Combining classifiers by bootstrap aggregation," *Pattern Recognition*, vol. 36, no. 6, pp. 1303–1309, 2003.

[46] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine Learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[47] D.-p. Niu, F.-l. Wang, L.-l. Zhang, D.-k. He, and M.-x. Jia, "Neural network ensemble modeling for nosiheptide fermentation process based on partial least squares regression," *Chemometrics and Intelligent Laboratory Systems*, vol. 105, no. 1, pp. 125–130, 2011.

[48] W. Kim, J. Park, J. Yoo, H. Kim, and C. G. Park, "Target localization using ensemble support vector regression in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 43, no. 4, pp. 1189–1198, Aug. 2013.

[49] S. B. Kotsiantis, D. Kanellopoulos, and I. D. Zaharakis, "Bagged averaging of regression models," in *Artificial Intelligence Applications and Innovations*, 2006, pp. 53–60.

[50] A. Scherbart and T. W. Nattkemper, "The diversity of regression ensembles combining bagging and random subspace method," in *Proc. Advances in Neuro-Information Processing (NIPS'09)*, 2009, pp. 911–918.

[51] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[52] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," *Machine Learning*, vol. 42, no. 3, pp. 287–320, 2001.

[53] A. Vezhnevets and V. Vezhnevets, "Modest Adaboost-Teaching Adaboost to generalize better," in *Graphicon*, vol. 12, no. 5, 2005, pp. 987–997.

[54] C. Domingo and O. Watanabe, "MadaBoost: A modification of AdaBoost," in *Proc. Annual Conference on Computational Learning Theory (COLT'00)*, 2000, pp. 180–189.

[55] S. Avidan, "SpatialBoost: Adding spatial reasoning to adaboost," in *Proc. Computer Vision (ECCV'06)*, 2006, pp. 386–396.

[56] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[57] L. Breiman, "Pasting small votes for classification in large databases and on-line," *Machine Learning*, vol. 36, no. 1-2, pp. 85–103, 1999.

[58] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[59] J. R. Lloyd, "GEFCom2012 hierarchical load forecasting: Gradient boosting machines and gaussian processes," *International Journal of Forecasting*, vol. 30, no. 2, pp. 369–374, Apr. 2014.

[60] R. Feely, "Predicting stock market volatility using neural networks," Master's thesis, Trinity College Dublin, 2000, B.A dissertation.

[61] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, pp. 119–139, 1997.

[62] D. Shrestha and D. Solomatine, "Experiments with AdaBoost. RT, an improved boosting scheme for regression," *Neural Computation*, vol. 18, no. 7, pp. 1678–1710, 2006.

[63] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. International Conference on Machine Learning (ICML'96)*, vol. 96, 1996, pp. 148–156.

[64] P. Kankanala, S. Das, and A. Pahwa, "Adaboost$^+$: An ensemble learning approach for estimating weather-related outages in distribution systems," *IEEE Trans. Power Syst.*, vol. 29, no. 1, pp. 359–367, Jan. 2014.

[65] Y. Lin and Y. Jeon, "Random forests and adaptive nearest neighbors," *Journal of the American Statistical Association*, vol. 101, no. 474, pp. 578–590, 2006.

[66] H. Jiang, Y. Deng, H.-S. Chen, L. Tao, Q. Sha, J. Chen, C.-J. Tsai, and S. Zhang, "Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes," *BMC Bioinformatics*, vol. 5, no. 1, p. 81, 2004.

[67] K.-Q. Shen, C.-J. Ong, X.-P. Li, Z. Hui, and E. P. Wilder-Smith, "A feature selection method for multilevel mental fatigue EEG classification," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 7, pp. 1231–1237, 2007.

[68] Y. Ye, Q. Wu, J. Zhexue Huang, M. K. Ng, and X. Li, "Stratified sampling for feature subspace selection in random forests for high dimensional data," *Pattern Recognition*, vol. 46, no. 3, pp. 769–787, 2013.

[69] L. Zhang, Y. Ren, and P. N. Suganthan, "Instance based random forest with rotated feature space," in *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'13)*, 2013, pp. 31–35.

[70] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[71] B. H. Menze, B. M. Kelm, D. N. Splitthoff, U. Koethe, and F. A. Hamprecht, "On oblique random forests," in *Machine Learning and Knowledge Discovery in Databases*, 2011, pp. 453–469.

[72] L. Zhang and P. N. Suganthan, "Random forests with ensemble of feature spaces," *Pattern Recognition*, vol. 47, no. 10, pp. 3429–3437, Oct. 2014.

[73] O. L. Mangasarian and E. W. Wild, "Multisurface proximal support vector machine classification via generalized eigenvalues," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 1, pp. 69–74, 2006.

[74] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, 2006.

[75] L. I. Kuncheva and J. J. Rodríguez, "An experimental study on rotation forest ensembles," in *Multiple Classifier Systems*, 2007, pp. 459–468.

[76] K.-H. Liu and D.-S. Huang, "Cancer classification using rotation forest," *Computers in Biology and Medicine*, vol. 38, no. 5, pp. 601–610, 2008.

[77] A. Ozcift, "SVM feature selection based rotation forest ensemble classifiers to improve computer-aided diagnosis of parkinson disease," *Journal of Medical Systems*, vol. 36, no. 4, pp. 2141–2147, 2012.

[78] J.-F. Xia, K. Han, and D.-S. Huang, "Sequence-based prediction of protein-protein interactions by means of rotation forest and autocorrelation descriptor," *Protein and Peptide Letters*, vol. 17, no. 1, pp. 137–145, 2010.

[79] G. Stiglic and P. Kokol, "Effectiveness of rotation forest in meta-learning based gene expression classification," in *Proc. IEEE International Symposium on Computer-Based Medical Systems (CBMS'07)*, 2007, pp. 243–250.

[80] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[81] L. Bo, L. Xinjun, and Z. Zhiyan, "Novel algorithm for constructing support vector machine regression ensemble," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 541–545, 2006.

[82] K. Lu and L. Wang, "A novel nonlinear combination model based on support vector machine for rainfall prediction," in *Proc. IEEE International Joint Conference on Computational Sciences and Optimization (CSO'11)*, 2011, pp. 1343–1346.

[83] L. Wang and J. Wu, "Application of hybrid RBF neural network ensemble model based on wavelet support vector machine regression in rainfall time series forecasting," in *Proc. IEEE International Joint Conference on Computational Sciences and Optimization (CSO'12)*, 2012, pp. 867–871.

[84] J.-S. Chou and A.-D. Pham, "Enhanced artificial intelligence for ensemble approach to predicting high performance concrete compressive strength," *Construction and Building Materials*, vol. 49, pp. 554–563, Dec. 2013.

[85] J. Wu, "A novel artificial neural network ensemble model based on k-nearest neighbor nonparametric estimation of regression function and its application for rainfall forecasting," in *Proc. International Joint Conference on Computational Sciences and Optimization (CSO'09)*, vol. 2, 2009, pp. 44–48.

[86] S. Wang, L. Yu, L. Tang, and S. Wang, "A novel seasonal decomposition based least squares support vector regression ensemble learning approach for hydropower consumption forecasting in China," *Energy*, vol. 36, no. 11, pp. 6542–6554, 2011.

[87] H. Liu, H.-Q. Tian, C. Chen, and Y.-F. Li, "A hybrid statistical method to predict wind speed and wind power," *Renewable Energy*, vol. 35, pp. 1857–1861, 2010.

[88] J. Catalão, H. Pousinho, and V. Mendes, "Hybrid Wavelet-PSO-ANFIS approach for short-term wind power forecasting in Portugal," *IEEE Trans. Sustain. Energy*, vol. 2, no. 1, pp. 50–59, Jan. 2011.

[89] A. U. Haque, P. Mandal, J. Meng, A. K. Srivastava, T.-L. Tseng, and T. Senjyu, "A novel hybrid approach based on wavelet transform and fuzzy ARTMAP network for predicting wind farm power production," in *Proc. IEEE Industry Applications Society Annual Meeting (IAS'12)*, Las Vegas, NV, Oct. 2012, pp. 1–8.

[90] S. H. Cao and J. C. Cao, "Forecast of solar irradiance using recurrent neural networks combined with wavelet analysis," *Applied Thermal Engineering*, vol. 25, pp. 161–172, 2005.

[91] N. Huang, Z. Shen, S. Long, M. Wu, H. Shih, Q. Zheng, N. Yen, C. Tung, and H. Liu, "The empirical mode decomposition and Hilbert spectrum for nonlinear and nonstationary time series analysis," *Proc. Royal Society London A*, vol. 454, pp. 903–995, 1998.

[92] X. Wang and H. Li, "One-month ahead prediction of wind speed and output power based on EMD and LSSVM," in *Proc. International Conference on Energy and Environment Technology (ICEET'09)*, vol. 3, 2009, pp. 439–442.

[93] Y. Zhang, U. J. Lu, Y. Meng, H. Yan, and H. Li, "Wind power short-term forecasting based on empirical mode decomposition and chaotic phase space reconstruction," *Automation of Electric Power Systems*, vol. 36, no. 5, pp. 24–28, 2012.

[94] X. Fan and Y. Zhu, "The application of empirical mode decomposition and gene expression programming to short-term load forecasting," in *Proc. International Conference on Natural Computation (ICNC'10)*, vol. 8, Yantai, China, 10-12 Aug. 2010, pp. 4331–4334.

[95] Y. Ren and P. N. Suganthan, "Empirical mode decomposition – k nearest neighbor models for wind speed forecasting," *Journal of Power and Energy Engineering*, vol. 2, no. 4, pp. 176–185, 2014.

[96] Y. Ren, P. N. Suganthan, and N. Srikanth, "A novel empirical mode decomposition with support vector regression for wind speed forecasting," *IEEE Trans. Neural Netw. Learn. Syst.*, no. 99, 2014.

[97] Y. Ren, X. Qiu, and P. N. Suganthan, "Empirical mode decomposition based adaboost-backpropagation neural network method for wind speed forecasting," in *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'14)*, Orlando, US, Dec. 2014.

[98] J. Shi, J. Guo, and S. Zheng, "Evaluation of hybrid forecasting approaches for wind speed and power generation time series," *Renewable and Sustainable Energy Reviews*, vol. 16, pp. 3471–3480, 2012.

[99] J. Wu and C. C. Keong, "Prediction of hourly solar radiation using a novel hybrid model of ARMA and TDNN," *Solar Energy*, vol. 85, no. 5, pp. 808–817, 2011.

[100] Y. Ren, P. N. Suganthan, N. Srikanth, and S. Sarkar, "A hybrid ARIMA-DENFIS method for wind speed forecasting," in *Proc. IEEE International Conference on Fuzzy Systems (FUZZ'13)*, 2013.

[101] M. Nguyen, H. Abbass, and R. McKay, "A novel mixture of experts model based on cooperative coevolution," *Neurocomputing*, vol. 1–3, pp. 155–163, 2006.

[102] ——, "Analysis of CCME: Coevolutionary dynamics, automatic problem decomposition and regularization," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 38, no. 1, pp. 100–109, 2008.

[103] D. L. Gonzalez, Z. Chen, I. K. Tetteh, T. Pansombut, F. Semazzi, V. Kumar, A. Melechko, and N. F. Samatova, "Hierarchical classifier-regression ensemble for multi-phase non-linear dynamic system response prediction: Application to climate analysis," in *Proc. IEEE International Conference on Data Mining Workshops (ICDMW'12)*. IEEE, 2012, pp. 781–788.

[104] S. Kohli, S. Prakash, and P. Gupta, "Hierarchical age estimation with dissimilarity-based classification," *Neurocomputing*, vol. 120, pp. 164–176, 23 Nov. 2013.

[105] Y. Liu and X. Yao, "Ensemble learning via negative correlation," *Neural Networks*, vol. 12, no. 10, pp. 1399–1404, 1999.

[106] G. Brown, J. L. Wyatt, and P. Tiňo, "Managing diversity in regression ensembles," *Journal of Machine Learning Research*, vol. 6, pp. 1621–1650, 2005.

[107] M. Alhamdoosh and D. Wang, "Fast decorrelated neural network ensembles with random weights," *Information Sciences*, vol. 264, pp. 104–117, 2014.

[108] H. Dam, H. Abbass, C. Lokan, and X. Yao, "Neural based learning classifier systems," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 26–39, Jan. 2008.

[109] F. Fernández-Navarro, P. Antonio Gutiérrez, C. Hervás-Martinez, and X. Yao, "Negative correlation ensemble learning for ordinal regression," *IEEE Trans. Neural Netw. Learn. Syst*, vol. 24, no. 11, pp. 1836–1849, Nov. 2013.

[110] Y. Jin, T. Okabe, and B. Sendhoff, "Neural network regularization and ensembling using multi-objective evolutionary algorithms," in *Proc. Congress on Evolutionary Computation (CEC'04)*, Portland, OR, 2004, pp. 1–8.

[111] A. Chandra and X. Yao, "DIVACE: Diverse and accurate ensemble learning algorithm," in *Intelligent Data Engineering and Automated Learning–IDEAL 2004*. Springer, 2004, pp. 619–625.

[112] ——, "Ensemble learning using multi-objective evolutionary algorithms," *Journal of Mathematical Modelling and Algorithms*, vol. 5, no. 4, pp. 417–445, 2006.

[113] H. Abbass, "Pareto neuro-ensembles," in *AI 2003: Advances in Artificial Intelligence*, T. Gedeon and L. Fung, Eds. Springer Berlin Heidelberg, 2003, vol. 2903, pp. 554–566.

[114] H. A. Abbass, "Pareto neuro-evolution: Constructing ensemble of neural networks using multi-objective optimization," in *Proc. IEEE Congress on Evolutionary Computation (CEC'03)*, vol. 3. IEEE, 2003, pp. 2074–2080.

[115] H. A. Abbass, R. Sarker, and C. Newton, "PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proc. IEEE Congress on Evolutionary Computation (CEC'01)*, vol. 2. IEEE, 2001, pp. 971–978.

[116] S. Gu and Y. Jin, "Generating diverse and accurate classifier ensembles using multi-objective optimization," in *Proc. IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM'14)*. IEEE, 2014, pp. 9–15.

[117] A. Chandra and X. Yao, "Evolving hybrid ensembles of learning machines for better generalisation," *Neurocomputing*, vol. 69, no. 7, pp. 686–700, 2006.

[118] D. W. Opitz and J. W. Shavlik, "Generating accurate and diverse members of a neural-network ensemble," *Advances in Neural Information Processing Systems*, vol. 8, pp. 535–541, 1996.

[119] H. Chen and X. Yao, "Evolutionary multiobjective ensemble learning based on bayesian feature selection," in *Proc. IEEE Congress on Evolutionary Computation (CEC'06)*. IEEE, 2006, pp. 267–274.

[120] ——, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Trans. Knowl. Data Eng*, vol. 22, no. 12, pp. 1738–1751, 2010.

[121] W.-C. Chen, L.-Y. Tseng, and C.-S. Wu, "A unified evolutionary training scheme for single and ensemble of feedforward neural network," *Neurocomputing*, vol. 143, pp. 347–361, 2014.

[122] C. J. Tan, C. P. Lim, and Y.-N. Cheah, "A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models," *Neurocomputing*, vol. 125, pp. 217–228, 2014.

[123] U. Bhowan, M. Johnston, and M. Zhang, "Ensemble learning and pruning in multi-objective genetic programming for classification with unbalanced data," in *AI 2011: Advances in Artificial Intelligence*. Springer, 2011, pp. 192–202.

[124] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, *Ensembles of multi-objective decision trees*. Springer, 2007.

[125] J.-C. Lévesque, A. Durand, C. Gagné, and R. Sabourin, "Multi-objective evolutionary optimization for generating ensembles of classifiers in the ROC space," in *Proc. Genetic and Evolutionary Computation (GECCO'12)*. ACM, 2012, pp. 879–886.

[126] N. Kondo, T. Hatanaka, and K. Uosaki, "RBF networks ensemble construction based on evolutionary multi-objective optimization," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, vol. 12, no. 3, pp. 297–303, 2008.

[127] F. Gullo, A. Talukder, S. Luke, C. Domeniconi, and A. Tagarelli, "Multiobjective optimization of co-clustering ensembles," in *Proc. annual conference companion on Genetic and evolutionary computation*. ACM, 2012, pp. 1495–1496.

[128] L. L. Minku and X. Yao, "An analysis of multi-objective evolutionary algorithms for training ensemble models based on different performance measures in software effort estimation," in *Proc. International Conference on Predictive Models in Software Engineering*. ACM, 2013, p. 8.

[129] C. Smith and Y. Jin, "Evolutionary multi-objective generation of recurrent neural network ensembles for time series prediction," *Neurocomputing*, vol. 143, pp. 302–311, 2014.

[130] C. Smith, J. Doherty, and Y. Jin, "Multi-objective evolutionary recurrent neural network ensemble for prediction of computational fluid dynamic simulations," in *Proc. IEEE Congress on Evolutionary Computation (CEC'14)*. IEEE, 2014, pp. 2609–2616.

[131] W. Du, S. Y. S. Leung, and C. K. Kwong, "Time series forecasting by neural networks: A knee point-based multiobjective evolutionary algorithm approach," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8049–8061, 2014.

[132] L. I. Kuncheva, ""Fuzzy" versus "nonfuzzy" in combining classifiers designed by boosting," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 729–741, 2003.

[133] R. Scherer, *Multiple Fuzzy Classification Systems*. Springer Publishing Company, Incorporated, 2014.

[134] A.-M. Yang, Y.-M. Zhou, and M. Tang, "A classifier ensemble method for fuzzy classifiers," in *Fuzzy Systems and Knowledge Discovery*. Springer, 2006, pp. 784–793.

[135] O. Cordón, A. Quirin, and L. Sánchez, "A first study on bagging fuzzy rule-based classification systems with multicriteria genetic selection of the component classifiers," in *Proc. International Workshop on Genetic and Evolving Systems (GEFS'08)*, 2008, pp. 11–16.

[136] Z. Fu, D. Zhang, L. Wang, and X. Li, "Ensemble learning algorithm in classification of fuzzy-classes*," *Journal of Computational Information Systems*, vol. 9, no. 22, pp. 8929–8938, 2013.

[137] Y. Nojima and H. Ishibuchi, "Designing fuzzy ensemble classifiers by evolutionary multiobjective optimization with an entropy-based diversity criterion," in *Proc. International Conference on Hybrid Intelligent Systems (HIS'06)*, 2006, pp. 59–59.

[138] H. Ishibuchi and T. Yamamoto, "Evolutionary multiobjective optimization for generating an ensemble of fuzzy rule-based classifiers," in *Proc. Genetic and Evolutionary Computation (GECCO'03)*, 2003, pp. 1077–1088.

[139] J. Cao, H. Wang, S. Kwong, and K. Li, "Combining interpretable fuzzy rule-based classifiers via multi-objective hierarchical evolutionary algorithm," in *Proc. IEEE International Conference on Systems, Man, and Cybernetics (SMC'11)*, 2011, pp. 1771–1776.

[140] Y. Zhang, H. Yang, S. Prasad, E. Pasolli, J. Jung, and M. Crawford, "Ensemble multiple kernel active learning for classification of multisource remote sensing data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 2, pp. 845–858, Feb 2015.

[141] X. Wang, X. Liu, N. Japkowicz, and S. Matwin, "Ensemble of multiple kernel SVM classifiers," in *Advances in Artificial Intelligence*. Springer, 2014, pp. 239–250.

[142] N. Shawe-Taylor and A. Kandola, "On kernel target alignment," *Advances in Neural Information Processing Systems*, vol. 14, p. 367, 2002.

[143] G. R. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *The Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[144] S. Sonnenburg, G. Rätsch, and C. Schäfer, "A general and efficient multiple kernel learning algorithm," in *Proc. Neural Information Processing Systems (NIPS'05)*, Dec. 2005.

[145] C. Cortes, M. Mohri, and A. Rostamizadeh, "Two-stage learning kernel algorithms," in *Proc. International Conference on Machine Learning (ICML'10)*, 2010, pp. 239–246.

[146] C. Cortes, M. Kloft, and M. Mohri, "Learning kernels using local rademacher complexity," in *Proc. Advances in Neural Information Processing Systems (NIPS'13)*, 2013, pp. 2760–2768.

[147] F. Orabona, L. Jie, and B. Caputo, "Multi kernel learning with online-batch optimization," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 227–253, 2012.

[148] X. Xu, I. W. Tsang, and D. Xu, "Soft margin multiple kernel learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 749–761, 2013.

[149] T. Suzuki and R. Tomioka, "SpicyMKL: A fast algorithm for multiple kernel learning with thousands of kernels," *Machine Learning*, vol. 85, no. 1-2, pp. 77–108, 2011.

[150] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien, "l_p-norm multiple kernel learning," *The Journal of Machine Learning Research*, vol. 12, pp. 953–997, 2011.

[151] J. Zhuang, I. W. Tsang, and S. Hoi, "Two-layer multiple kernel learning," in *International Conference on Artificial Intelligence and Statistics*, 2011, pp. 909–917.

[152] Z. Sun, N. Ampornpunt, M. Varma, and S. Vishwanathan, "Multiple kernel learning and the SMO algorithm," in *Proc. Advances in Neural Information Processing Systems (NIPS'10)*, 2010, pp. 2361–2369.

[153] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. International Conference on Machine Learning (ICML'04)*, 2004, p. 6.

[154] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[155] A. Zien and C. S. Ong, "Multiclass multiple kernel learning," in *Proc. International Conference on Machine Learning (ICML'07)*, 2007, pp. 1191–1198.

[156] K. Crammer, J. Keshet, and Y. Singer, "Kernel design using boosting," in *Proc. Advances in Neural Information Processing Systems (NIPS'02)*, 2002, pp. 537–544.

[157] V. Pisetta, P.-E. Jouve, and D. A. Zighed, "Learning with ensembles of randomized trees: New insights," in *Machine Learning and Knowledge Discovery in Databases*, 2010, pp. 67–82.

[158] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[159] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[160] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'12)*, 2012, pp. 3642–3649.

[161] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Networks*, vol. 32, pp. 333–338, 2012.

[162] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[163] F. Agostinelli, M. R. Anderson, and H. Lee, "Adaptive multi-column deep neural networks with application to robust image denoising," in *Proc. Advances in Neural Information Processing Systems (NIPS'13)*, 2013, pp. 1493–1501.

[164] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *The Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.

[165] A. Abdullah, R. C. Veltkamp, and M. A. Wiering, "An ensemble of deep support vector machines for image categorization," in *Proc. IEEE International Conference of Soft Computing and Pattern Recognition (SOCPAR'09)*, 2009, pp. 301–306.

[166] D. M. Tax, R. P. Duin, and M. Van Breukelen, "Comparison between product and mean classifier combination rules," in *Proc. Workshop on Statistical Pattern Recognition*, 1997.

[167] A. Abdullah, R. C. Veltkamp, and M. A. Wiering, "Spatial pyramids and two-layer stacking SVM classifiers for image categorization: A comparative study," in *Proc. IEEE International Joint Conference on Neural Networks (IJCNN'09)*, 2009, pp. 5–12.

[168] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[169] J. Ba and B. Frey, "Adaptive dropout for training deep neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS'13)*, 2013, pp. 3084–3092.

[170] L. Wan, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, "Regularization of neural networks using dropconnect," in *Proc. International Conference on Machine Learning (ICML'13)*, 2013, pp. 1058–1066.

[171] A. Berg, J. Deng, and L. Fei-Fei, "Large scale visual recognition challenge 2010," 2010.

[172] E. Busseti, I. Osband, and S. Wong, "Deep learning for time series modeling," Stanford University, CA, Tech. Rep. CS 229, Dec. 2012.

[173] M. Wiering, M. Schutten, A. Millea, A. Meijster, and L. R. B. Schomaker, "Deep support vector machines for regression problems," in *Proc. International Workshop on Advances in Regularization, Optimization, Kernel Methods, and Support Vector Machines: Theory and Applications*, 2013.

[174] X. Qiu, L. Zhang, Y. Ren, P. N. Suganthan, and G. Amaratunga, "Ensemble deep learning for regression and timeseries forecasting," in *Proc. IEEE Symposium on Computational Intelligence and Ensemble Learning (CIEL'14)*, Orlando, US, Dec. 2014.

[175] S. M. Halawani, I. A. Albidewi, and A. Ahmad, "A novel ensemble method for regression via classification problems," *Journal of Computer Science*, vol. 7, no. 3, p. 387, 2011.

[176] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: Discussions from data analytics perspectives [discussion forum]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.

[177] S. Alam, G. Dobbie, Y. S. Koh, P. Riddle, and S. U. Rehman, "Research on particle swarm optimization based clustering: a systematic review of literature and techniques," *Swarm and Evolutionary Computation*, vol. 17, pp. 1–13, 2014.

[178] S. J. Nanda and G. Panda, "A survey on nature inspired metaheuristic algorithms for partitional clustering," *Swarm and Evolutionary Computation*, vol. 16, pp. 1–18, 2014.
[179] P. Mc Leod and B. Verma, "Variable hidden neuron ensemble for mass classification in digital mammograms [application notes]," *IEEE Computational Intelligence Magazine*, vol. 8, no. 1, pp. 68–76, 2013.
[180] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.
[181] R. Mallipeddi, S. Jeyadevi, P. N. Suganthan, and S. Baskar, "Efficient constraint handling for optimal reactive power dispatch problems," *Swarm and Evolutionary Computation*, vol. 5, pp. 28–36, 2012.
[182] R. Mallipeddi, P. N. Suganthan, Q.-K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696, 2011.
[183] Y. Chen, W. Xie, and X. Zou, "How can surrogates influence the convergence of evolutionary algorithms?" *Swarm and Evolutionary Computation*, vol. 12, pp. 18–23, 2013.
[184] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
[185] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 2009, pp. 248–255.
[186] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, MA, Tech. Rep. 07-49, 2007.