2019

Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems

Mohsen Shahhosseini
Iowa State University, mohsen@iastate.edu

Guiping Hu
Iowa State University, gphu@iastate.edu

Hieu Pham
Iowa State University

Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems

Abstract
Aggregating multiple learners through an ensemble of models aims to make better predictions by capturing the underlying distribution more accurately. Different ensembling methods, such as bagging, boosting and stacking/blending, have been studied and adopted extensively in research and practice. While bagging and boosting intend to reduce variance and bias, respectively, blending approaches target both by finding the optimal way to combine base learners to find the best trade-off between bias and variance. In blending, ensembles are created from weighted averages of multiple base learners. In this study, a systematic approach is proposed to find the optimal weights to create these ensembles for bias-variance tradeoff using cross-validation for regression problems (Cross-validated Optimal Weighted Ensemble (COWE)). Furthermore, it is known that tuning hyperparameters of each base learner inside the ensemble weight optimization process can produce better performing ensembles. To this end, a nested algorithm based on bi-level optimization that considers tuning hyperparameters as well as finding the optimal weights to combine ensembles (Cross-validated Optimal Weighted Ensemble with Internally Tuned Hyperparameters (COWE-ITH)) was proposed. The algorithm is shown to be generalizable to real data sets though analyses with ten publicly available data sets. The prediction accuracies of COWE-ITH and COWE have been compared to base learners and the state-of-art ensemble methods. The results show that COWE-ITH outperforms other benchmarks as well as base learners in 9 out of 10 data sets.

Keywords
Ensemble, Blending, Bi-level Optimization, Bias-Variance tradeoff, Hyperparameters

Disciplines
Statistical Models | Systems Engineering

Comments
This is a pre-print of the article Shahhosseini, Mohsen, Guiping Hu, and Hieu Pham. "Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems." *arXiv preprint arXiv:1908.05287* (2019). Posted with permission.

Optimizing Ensemble Weights and Hyperparameters of Machine Learning Models for Regression Problems

Mohsen Shahhosseini[1], Guiping Hu[1*], Hieu Pham[1]

[1] Department of Industrial and Manufacturing Systems Engineering, Iowa State University, Ames, Iowa, 50011, USA
* Corresponding author: Tel.: (515) 294-8638; Fax: (515) 294-3524; E-mail: gphu@iastate.edu

Abstract

Aggregating multiple learners through an ensemble of models aims to make better predictions by capturing the underlying distribution more accurately. Different ensembling methods, such as bagging, boosting and stacking/blending, have been studied and adopted extensively in research and practice. While bagging and boosting intend to reduce variance and bias, respectively, blending approaches target both by finding the optimal way to combine base learners to find the best trade-off between bias and variance. In blending, ensembles are created from weighted averages of multiple base learners. In this study, a systematic approach is proposed to find the optimal weights to create these ensembles for bias-variance tradeoff using cross-validation for regression problems (Cross-validated Optimal Weighted Ensemble (COWE)). Furthermore, it is known that tuning hyperparameters of each base learner inside the ensemble weight optimization process can produce better performing ensembles. To this end, a nested algorithm based on bi-level optimization that considers tuning hyperparameters as well as finding the optimal weights to combine ensembles (Cross-validated Optimal Weighted Ensemble with Internally Tuned Hyperparameters (COWE-ITH)) was proposed. The algorithm is shown to be generalizable to real data sets though analyses with ten publicly available data sets. The prediction accuracies of COWE-ITH and COWE have been compared to base learners and the state-of-art ensemble methods. The results show that COWE-ITH outperforms other benchmarks as well as base learners in 9 out of 10 data sets.

Keywords: *Ensemble, Blending, Bi-level Optimization, Bias-Variance tradeoff, Hyperparameters*

1. Introduction

A learning program is given data in the form $D = \{(X_i, y_i): X_i \in \mathbb{R}^{m\,x\,n}, y_i \in \mathbb{R}\}$ with some unknown underlying function $y = f(x)$ where the x_is are predictor variables and the y_is are the responses with *m* instances and *n* predictor variables. Given a subset S of D, a predictive learner is constructed on S, and given new values of X and Y not in S, predictions will be made for a corresponding Y. These predictions can be computed from any machine learning method or statistical model such as linear regression, trees or neural networks (Large et al. 2019). In the case where Y is discrete, the learning program is a classification problem. If Y is continuous, the learning program is a regression problem. The focus of this paper is on regression where the goal is to accurately predict continuous responses.

Many predictions can be based on a single model such as a single decision tree, but there is strong evidence that a single model can be outperformed by an ensemble of models, that is, a collection of individual models that can be combined to reduce bias, variance, or both (Dietterich 2000). A single model is unlikely to capture the entire underlying structure of the data to achieve optimal predictions. This is where integrating multiple models can improve prediction accuracy significantly. By aggregating multiple base learners (individual models), more information can be captured on the underlying structure of the data (Brown et al. 2005). The popularity of ensemble modeling can be seen in various practical applications such as the Netflix Prize, the data mining world cup, and Kaggle competitions (Töscher and Jahrer 2008; Niculescu-Mizil et al. 2009; Koren 2009; Yu et al. 2010; Taieb and Hyndman 2014; Hoch 2015; Sutton et al. 2018; Kechyn et al. 2018).

Although ensembling models in data analytics are well-motivated, not all ensembles are created equal. Specifically, different types of ensembling include bagging, boosting, and stacking/blending (Breiman 1996; Freund 1995; Wolpert 1992). Bagging forms an ensemble with sampling from training data with replacement (bootstrap) and averaging or voting over class labels (Breiman 1996); boosting constructs ensemble by combining weak learners with the expectation that subsequent models would compensate for errors made by earlier models (Brown 2017); and stacking takes the output of the base learners on the training data and applies another learning algorithm on them to predict the response values (Large et al. 2019). Each method has its strengths and weaknesses. Bagging tends to reduce variance and does not work well with relatively simple models; boosting aims at reducing bias by sequentially combining weak learners but is sensitive to noisy data and outliers and is susceptible of overfitting; while stacking tries to reduce bias, that is, to fix the errors that base learners made by fitting one or more meta-models on the predictions made by base learners. (Brown 2017; Large et al. 2019). In this study, we focus on blending, that is, stacking type of creating ensembles, in which based learners are integrated with a weighted average. Although seemingly straightforward, the procedure for creating an ensemble is a scientific process. For an ensemble to outperform any of its individual components, the individual learners must be accurate and diverse enough to effectively capture the structure of the data (Hansen and Salamon 1990). However, determining the diversities of models to include is one challenging part of constructing an

optimal ensemble. For the 2017 KDD cup, the winning team utilized an ensemble of 13 models which included trees, neural networks and linear models (Hu et al. 2017). This diversity in models is where the strength of an ensemble lies. Specifically, trees and neural networks are nonlinear models, where they partition the data space differently than linear models (Khaki et al. 2019a; Khaki et al. 2019b). As such, these models represent different features of the data, and once combined, can collectively represent the entire data space better than they would individually. However, in addition to determining the base models to be included, which we will not discuss in this study, there are two additional components that must be addressed. The first is how to tune the hyperparameters of each base model and the second is how to weight the base models to make the final predictions.

As previously stated, the construction of an ensemble model is a systematic process of combining many diverse base predictive learners. However, one area that has not been given much attention is *how* to combine the models to obtain an optimal learner. When aggregating predictive learners, there is always the question of how to weight each model as well as how to tune the parameters of the individual learners. The most straightforward approach is simply to average the pre-tuned base models, that is, all base models are given equal weight. However, numerous studies have shown that a simple average of models is not always the best and that a weighted ensemble can provide superior prediction results (Bhasuran 2016; Ekbal and Saha 2013; Winham et al. 2013; Shahhosseini et al. 2019). Moreover, the hyperparameter tuning process for each base model is often carried out separately as an independent procedure when in fact it should be part of the entire machine learning framework. That is, implementations of a weighted ensemble consider the tuning of hyperparameters and weighting of models as two independent steps instead of as an integrated process. These gaps in the machine learning model ensemble serve as the major motivations for this study.

In this paper, we propose an admissible framework for creating an optimal ensemble by considering the tuning of hyperparameters and weighting of models concurrently. The ensemble weighting scheme was firstly introduced in a conference paper (Shahhosseini et al. 2019), where the optimization model was proposed to find the optimal ensemble weights. However, the tuning of hyperparameters was conducted separately, only classical average ensemble benchmark was used to compare, and only Boston housing dataset was analyzed. The decision making framework model is formulated as and solved with a non-linear convex optimization approach.

To evaluate the proposed algorithm, numerical experiments on several data sets from different areas have been conducted to demonstrate the generalizability of the proposed scheme.

The main questions that we want to address in this paper are:

1) Does the proposed method introduce improvements over base learners?
2) How does the proposed method compare to conventional ensembling techniques?
3) What is the effect of tuning hyperparameters as part of finding optimized ensembles on the quality of predictions?
4) Can the results be generalized to multiple data sets?

The remainder of this paper is organized as follows. Section 2 reviews the literature in the related fields; mathematics and concepts of the optimization model is presented in Section 3; the proposed scheme (COWE-ITH) is introduced in Section 4; the results of comparing the proposed method with benchmarks are presented and discussed in Section 5; and finally, Section 6 concludes the paper with major findings and discussions.

2. Background

There have been extensive studies on weighted ensembles in the literature. Shen and Kong (2004) proposed a weighted ensemble of neural networks for regression problems using the natural idea that higher training accuracy results in higher weight for a model. Kim et al. (2019) proposed a weighted ensemble approach using the least squares method to estimate weights, thus showing that their weighted ensemble is an improvement to the classical ensemble. Pham and Olafsson (2019b) proposed using the method of Cesaro averages for their weighting scheme essentially following a weighting pattern in line with Riemann zeta function with another generalization in Pham and Olafsson (2019a).

Moreover, the applications areas in which ensemble approaches are used span a variety of areas. Belayneh et al. (2016) constructed an ensemble of bootstrapped artificial neural networks to predict drought conditions of a river basis in Ethiopia, whereas Martelli et al. (2003) constructed an ensemble of neural networks to predict membrane protein achieving superior results than previous methods. Aside from neural networks, Van Rijn et al. (2018) investigated the use of heterogeneous ensembles for data streams and introduced an online estimation framework to dynamically update the prediction weights of base learners. Zhang and Mahadevan (2019) constructed an ensemble of support vector machines to model the incident rates in aviation. Conroy et al. (2016) proposed a dynamic ensemble approach for imputing missing data in classification problems and compared the results of their proposed method with other common missing data approaches. Multi-target regression task was addressed in a study by Breskvar et al. (2018) where ensembles of generalized decision trees with added randomization were used. In a study similar to the present paper, Large et al. (2019) introduced a probabilistic ensemble weighting scheme based on cross-validation for classification problems. The similarity of this study with the present paper is using cross-validated predictions for finding the optimal ensemble weights, whereas the differences/improvements are considering bias-variance tradeoff of the optimal ensembles, proposing a novel bi-level optimization approach that accounts for optimizing hyperparameters and ensemble weights in different levels, and using random search and Bayesian search to find the hyperparameters instead of grid search. As can be seen, aside from data science competitions, constructing an ensemble of models has many real-world applications due to the potential to achieve superior performance to that of a single model.

It can be observed that the existing ensembling studies all consider the base model construction and the weighted averaging to be independent steps. Intuition tells us that considering the tuning of model parameters in conjunction with the weighted average should produce a superior ensemble. This intuition can be thought of in terms of the bias-variance tradeoff (Yu et al. 2006). Namely, if each base model is optimally tuned individually, then by definition they will have low bias but will have high variance. Therefore, by further combining these optimally tuned models we will create an ensemble which ultimately has low bias and high variance. However, by considering the model tuning and weighting as two concurrent processes (as opposed to independent), then we can balance both bias and variance to obtain an optimal ensemble – the goal of this paper. In this study, we proposed a method that integrates the parameter tuning of the individual models and the ensemble weights design where the bias and variance trade-off is considered altogether in one decision making framework.

To the best of our knowledge, there have not been studies that combing the model hyperparameter tuning and the model weights aggregation for optimal ensemble design. Motivated by this gap in the literature, we propose a novel bi-level optimization approach that accounts for optimizing hyperparameters and ensemble weights in different levels to address this issue. We formulated our model with the objective to minimize bias and variance and account for the model hyperparameters and aggregate weights for each predictive learner with a bi-level nonlinear, convex program where it is guaranteed to find the optimal solution to the objective function.

3. Materials and methods

Ensemble learning has been shown to outperform individual base models in various studies (Brown 2017), but as mentioned previously, designing a systematic method to combine base models is of great importance. Based on many data science competitions, the winners are the ones who achieved superior performance by finding the best way to integrate the merits of different models (Puurula et al. 2014; Hong et al. 2014; Hoch 2015; Wang et al. 2015; Zou et al. 2017). It has been shown that the optimal choice of weights aims to obtain the best prediction error by designing the ensembles for the best bias and variance balance (Shahhosseini et al. 2019).

Prediction error of a model includes errors from two components: bias and variance. Both are determined by the interactions between the data and model choice. Bias is a model's understanding of the underlying relationship between features and target outputs; whereas, variance is the sensitivity to perturbations in training data. For a given data set $D = \{(X_i, y_i): X_i \in \mathbb{R}^{m\,x\,n}, y_i \in \mathbb{R}\}$, we assume there exists a function $f: \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ with noise ϵ such that $y = f(x_i) + \epsilon$ where $\epsilon \sim N(0,1)$.

Assuming the prediction of a base learner for the underlying function $f(x)$ to be $\hat{f}(x)$, We define bias and variance as follows.

$$Bias\,[\hat{f}(x)] = E\big[\hat{f}(x)\big] - f(x) \tag{1}$$

$$Var[\hat{f}(x)] = E\big[\hat{f}(x)^2\big] - E\big[\hat{f}(x)\big]^2 \tag{2}$$

Based on bias-variance decomposition (Hastie et al. 2005) the above definitions for bias and variance can be aggregated to the following:

$$E\left[\left(f(x) - \hat{f}(x)\right)^2\right] = \left(Bias\,[\hat{f}(x)]\right)^2 + Var\big[\hat{f}(x)\big] + Var(\epsilon) \tag{3}$$

The third term, $Var(\epsilon)$, in Equation (3) is called irreducible error, which is the variance of the noise term in the true underlying function ($f(x)$) and cannot be reduced by any model (Hastie et al. 2005).

The learning objective of every prediction task is to approximate the true underlying function with a predictive model which has low bias and low variance, but this is not always accessible. Common approaches to reduce variance are cross-validation and bagging (bootstrapped aggregated ensemble). On the other hand, reducing bias is done commonly with boosting. Although each of these approaches have its own merits and shortcomings, finding the optimal balance between them is the main challenge (Zhang and Ma 2012).

To find the optimal way to combine base learners, a mathematical optimization approach is used that is able to find ensemble optimal weights. We consider regression problems which have continuous targets to predict in this article. Taking both bias and variance into account, and knowing that mean squared error (MSE) is defined as the expected prediction error ($E[(f(x) - f\hat{}(x))\hat{}2\,]$) (Hastie et al. 2005), the objective function in the mathematical model for optimizing ensemble weights is chosen to be MSE (Shahhosseini et al. 2019).

The following optimization model which we call Cross-validation Optimal Weighted Ensemble (COWE) intends to find the best way to combine predictions of base learners by finding the optimal weight to aggregate them in a way that the created ensemble minimizes the total expected prediction error (MSE). Note that the predictions of each base learner ($\hat{Y}_i$) are the predictions of trained base learners on the hold-out set of an *n*-fold cross-validation.

$$\begin{aligned} Min\ \ & MSE\big(w_1\hat{Y}_1 + w_2\hat{Y}_2 + \cdots + w_k\hat{Y}_k\ ,\ Y\big) \\ & s.t. \\ & \textstyle\sum_{j=1}^{k} w_j = 1, \\ & w_j \geq 0, \quad \forall j = 1, \ldots, k. \end{aligned} \tag{4}$$

where w_j is the weights corresponding to base model *j* ($j = 1, \ldots, k$), $\hat{Y}_j$ represents the vector of predictions of base model *j* on the validation instances of cross-validation, and Y is the vector of true response values. Assuming n is the total number of instances, y_i as the true value of observation i, and $\hat{y}_{ij}$ as the prediction of observation i by base model j, the optimization model is as follows.

$$Min \ \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{k} w_j \hat{y}_{ij}\right)^2 \quad (5)$$
$$s.t.$$
$$\sum_{j=1}^{k} w_j = 1,$$
$$w_j \geq 0, \quad \forall j = 1, \dots, k.$$

The above formulation is a nonlinear convex optimization problem. As the constraints are linear, computing the Hessian matrix will demonstrate the convexity of the objective function. Hence, since a local optimum of a convex function (objective function) on a convex feasible region (feasible region of the above formulation) is guaranteed to be a global optimum, the optimal solution of this problem is proved to be the global optimal solution (Boyd and Vandenberghe 2004).

The COWE algorithm is displayed in Fig.1.

Inputs: Data set $D = \{(\boldsymbol{x}, y): \boldsymbol{x} \in \mathbb{R}^{n\times m}, y \in \mathbb{R}^n\}$;
Base learning algorithm L;
For $j = 1, \dots, k$:
$\hat{Y}_j = L(D)$ % Train a base learner j
Compute w_j from optimization problem (5)
Combine base learners $1, \dots, j$ with weights $w_1, \dots, w_j$.
End
Outputs: Optimal objective value (MSE^*)
Optimal ensemble weights ($w_1^*, \dots, w_j^*$)
Prediction vector of the ensemble with optimal weights ($\hat{Y}^*$)

Fig.1 The COWE algorithm

The optimization model presented in this section (COWE) assumes hyperparameters of each base learner is tuned before conducting the learning task. For example, if one of the base learners is the random forest, its hyperparameters are tuned with one of the many common tuning approaches and the predictions made with the tuned model act as the inputs of the optimization model to find the optimal ensemble weights. One of the main questions of this study is whether the best performing ensemble results from the set of tuned hyperparameters. To answer this question, an algorithm is proposed which is based on bi-level optimization. This algorithm makes it possible to find the best set of hyperparameters that results in the best ensemble.

4. Cross-validated Optimal Weighted Ensembles with Internally Tuned hyperparameters (COWE-ITH)

A nonlinear optimization model was proposed in section 3 to find the optimal weights of combining different base learner predictions. In this section, we want to find the effect of tuning hyperparameters of each base learner on the optimal ensemble weights. A common approach in creating ensembles is tuning hyperparameters of each base model with different searching methods

like grid search, random search, Bayesian optimization, etc., independently and then combine the predictions of those tuned base learners by some weights. We claim here that the ensemble with the best prediction accuracy (the least mean squared error) may not be created from hyperparameters tuned individually. To this end, we propose two bi-level optimization based nested algorithms with regard to two different search methods that aim to find the best combination of hyperparameters that result in the least prediction error. .2 demonstrates a flow chart of traditional weighted ensemble creation and COWE – ITH, respectively.

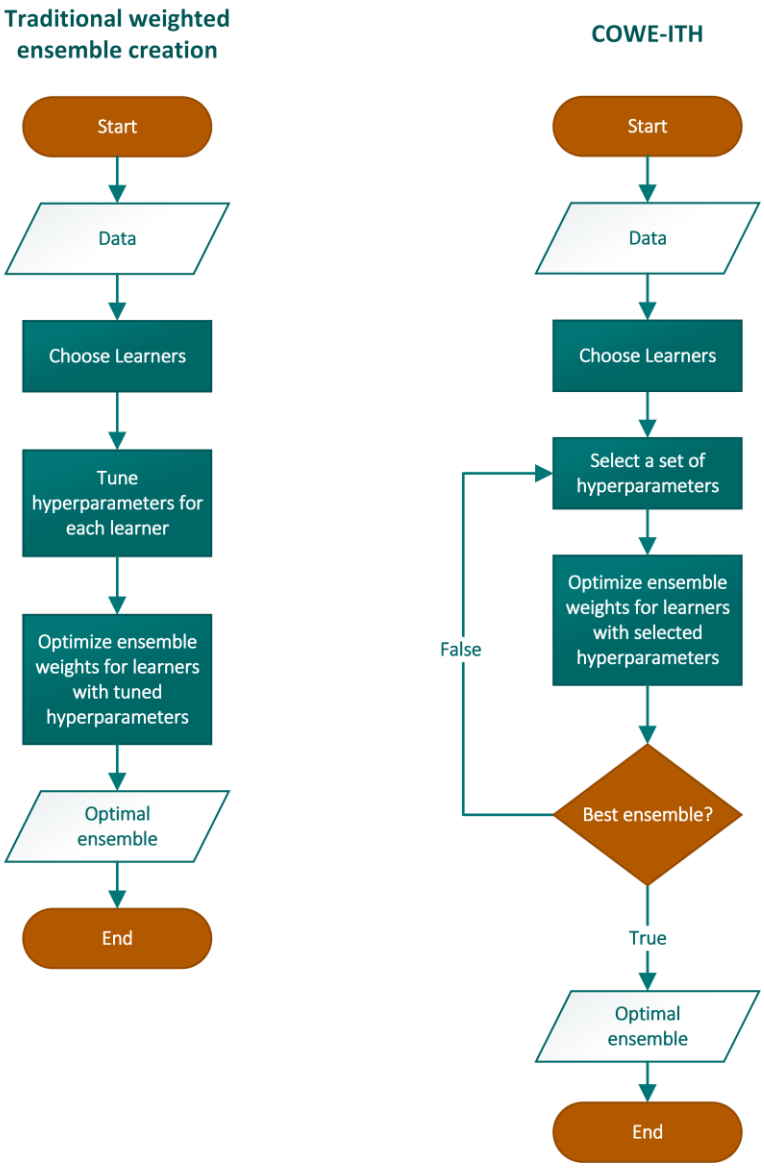

Fig.2 traditional weighted ensemble creation flowchart vs. COWE-ITH flowchart

4.1. COWE-ITH with random search

Random search addresses the exhaustive enumeration of grid search and replaces it with selecting random subsets of hyperparameter combinations. This method can be applied on discrete, continuous, or mixed spaces and it is able to outperform grid search in terms of solution quality and computational time. Furthermore, prior knowledge can be used to specify the distribution of hyperparameters when setting them (Bergstra and Bengio 2012).

In this article, we consider two settings of random search: discrete and mixed settings. In the discrete setting, n_d combinations of all the hyperparameter combinations are chosen randomly and a nested algorithm finds the best weights to create ensembles with respect to the hyperparameter settings.

Inputs: Data set $D = \{(\boldsymbol{x}, y): \boldsymbol{x} \in \mathbb{R}^{n \times m}, y \in \mathbb{R}^n\}$;
Base learning algorithm L;
Hyperparameters $h_1 = \alpha_1, \dots, \alpha_t$,
$\dots$,
$h_p = \varphi_1, \dots, \varphi_v$;

Randomly choose n_d combinations of all h_1, h_2, …, and h_p combinations
$h_1 = \alpha_1, \dots, \alpha_{t'}$,
$\dots$, % n_d combinations of hyperparameters
$h_p = \varphi_1, \dots, \varphi_{v'}$;
For $j = 1, \dots, k$:
  For $h_1 = \alpha_1, \dots, \alpha_{t'}$:
    For $h_2 = \beta_1, \dots, \beta_{u'}$:
      …
        For $h_p = \varphi_1, \dots, \varphi_{v'}$:
          $\hat{Y}_{jh_1,\dots,h_l} = L(D)$ % Train base learner j with hyperparameters $h_1, h_2, \dots, h_p$
          Compute $w_{jh_1 \dots h_p}$ from optimization problem (5)
          Combine base learners $1, \dots, j$ with weights $w_{1h_1,\dots,h_p}, \dots, w_{jh_1,\dots,h_p}$.
        End.
    End.
  End.
End.
Find the minimum of objective values in optimization problem (5) for all found weights.
Find the weights $w^*_{1h^*_1,\dots,h^*_p}, \dots, w^*_{jh^*_1,\dots,h^*_p}$ corresponding to the minimum objective value in optimization problem (5).

Outputs: Optimal objective value (MSE^*)
Optimal combination of hyperparameters $h^*_1, h^*_2, \dots, h^*_p$.
Optimal ensemble weights $w^*_{1h_1,\dots,h_p}, \dots, w^*_{jh_1,\dots,h_p}$
Prediction vector of ensemble with optimal weights ($\hat{Y}^*$)

Fig.3 The COWE-ITH algorithm with random search (discrete)

On the other hand, in the mixed setting, which is a combination of discrete and continuous hyperparameter values, n_m sets of hyperparameters are randomly chosen to find the optimal ensemble weights. Assuming d and c represent the number of discrete and continuous hyperparameters, respectively, the proposed algorithm is similar to Fig.3, with the difference that n_m of hyperparameters sets are randomly chosen from d discrete and c continuous hyperparameters.

4.2. COWE-ITH with Bayesian optimization

Bayesian optimization aims to approximate the unknown function with surrogate models like Gaussian process. The main difference between Bayesian optimization and other search methods is incorporating prior belief about the underlying function and updating it with new observations. Bayesian optimization tries to gather observations with the highest information in each iteration by making a balance between exploration (exploring uncertain hyperparameters) and exploitation (gathering observations from hyperparameters close to the optimum) (Snoek et al. 2012).

Given b iterations of Bayesian optimization, b combinations of hyperparameters have been identified and treated as the inputs to the optimization problem (5) in a proposed nested algorithm. The algorithm is similar to Fig.3 with the change of n_d random combinations of hyperparameters to b combinations of hyperparameters chosen by running Bayesian optimization.

5. Results and discussion

To evaluate the proposed algorithm, numerical experiments on several data sets downloaded from UCI Machine Learning Repository[1] (Dua and Graff 2019) and Kaggle Data sets[2] from different areas have been conducted to demonstrate the generalizability of the proposed scheme. Details of these data sets are shown in Table.1 (Harrison et al. 1978; Brooks et al. 1989; Quinlan 1993; Yeh 1998; Efron et al. 2004; Cortez and Morais 2007; Cortez et al. 2009; Tsanas and Xifara 2012; Acharya et al. 2019).

[1] https://archive.ics.uci.edu/ml/index.php
[2] https://www.kaggle.com/datasets

Table.1 data sets chosen to evaluate COWE-ITH

	Data sets	Number of Instances	Number of Attributes	Area
1	Airfoil Self-Noise	1503	6	Physical
2	Auto MPG	398	8	Automobiles
3	Boston Housing	506	14	Housing
4	Concrete Compressive Strength	1030	9	Physical
5	Diabetes Data	442	10	Life
6	Energy efficiency	768	8	Computer
7	Forest Fires	517	13	Physical
8	Graduate Admissions	500	9	Education
9	Wine Quality	4898	12	Business
10	Yacht Hydrodynamics	308	7	Physical

Four machine learning algorithms with minimal pre-processing tasks were designed for each data set separately and the proposed algorithm is applied to them using similar hyperparameters settings. 5-fold cross-validation was used as the validation method for all designed ML models and the entire process was repeated 5 times. In addition, 20% of each data set was held out for testing and the training and optimizing procedure was done on the remaining 80%.

The only parameter of the proposed method (COWE-ITH) that may result in different quality of outcomes is the number of iterations for search methods, or in other words, the number of selected hyperparameter sets. This parameter is closely related to the bias-variance tradeoff. The experiments showed that choosing large values for this parameter results in decreasing the bias with the cost of increasing variance, that is overfitting on training subset and lack of generalizability. On the flip side, experiments demonstrated that selecting small values for this parameter results in increasing the bias, or underfitting. To this end, the number of iterations for random search and Bayesian search methods is selected to be *5* and *RandomizedSearchCV* package from *Scikit-learn* library (Pedregosa et al. 2011) and *hyperopt* package (Bergstra et al. 2013) were used to implement random search and Bayesian search methods in Python 3, respectively. Also, Sequential Least Squares Programming algorithm (SLSQP) from Python's SciPy optimization library were used to solve optimization problems (Jones et al. 2001)

Table.2 presents the details of ML models and their hyperparameters settings (All other hyperparameters are set to their default values).

Table.2 ML models and their hyperparameters discrete and mixed settings

ML Models	Hyperparameter Settings		
	Parameter	Discrete	Mixed
LASSO	*alpha*	10^range(-5, 0.5, 0.5)[3]	Uniform(10^(-5) , 1)[4]
Random Forest	*n_estimators*	{100, 200, 500}	{100, 200, 500}
	max_depth	{4, 5, 6, 7, 8, 9, 10}	{4, 5, 6, 7, 8, 9, 10}
XGBoost	*gamma*	{5, 10}	Uniform(5 , 11)
	learning_rate	{0.1, 0.3, 0.5}	Uniform(0.1 , 0.6)
	n_estimators	{50, 100, 150}	{50, 100, 150}
	max_depth	{3, 6, 9}	{3, 6, 9}
SVM (rbf kernel)	*C*	linspace(0.01, 5, 20)[5]	Uniform(0.01, 5)
	gamma	range(0.01, 0.55, 0.05)	Uniform(0.01, 0.55)

Apart from the optimized ensemble method introduced in Section 3 (COWE) and in Shahhosseini et al. (2019), two other benchmarks have been used to compare the results of proposed learning methodology with them. First is the ensembles constructed with averaging the input base models which we call classical ensembles, and the second one is stacked ensembles with linear regression which we will call stacked regression. The latter benchmark has been widely used as one of the most effective methods to create ensembles and is created with fitting a linear regression model on the predictions made by different base learners.

Table.3 shows the average results of COWE-ITH based off of three search methods along with mean squared error of predictions made by each base learner and benchmarks. The superiority of ensemble techniques can be seen by comparing their prediction errors with base learners. This answers the first question asked in the Introduction section.

Table.3 The average results of applying ML models and created ensembles on ten public data sets

Data set	Search Method	Objective value on test set (MSE)							
		LASSO	Random Forest	SVM	XGBoost	Classical Ensemble	Stacked Regression	COWE	COWE-ITH
Airfoil Self-Noise	Bayesian	23.4289	4.2168	9.9587	4.1317	6.8706	3.6702	3.8070	3.3253
	Random (Mixed)	24.4285	4.3094	11.4062	3.5947	7.1981	3.3305	3.4781	3.1526
	Random (Discrete)	24.4767	4.3094	13.3493	3.9158	7.7875	3.4916	3.6823	3.2451
Auto MPG	Bayesian	10.8118	8.0348	9.1152	8.5998	7.6166	7.6596	7.6488	6.9482
	Random (Mixed)	10.8251	6.7550	8.2207	7.4734	6.8291	6.8746	6.7564	6.4186
	Random (Discrete)	10.8386	6.7550	9.0742	7.3809	7.0607	6.7861	6.7817	6.4943
Boston Housing	Bayesian	24.7792	11.5002	27.2316	11.3105	14.0182	11.2944	11.1304	10.5815
	Random (Mixed)	27.9904	14.4803	30.6102	12.2747	15.0038	13.2463	12.8253	11.7065
	Random (Discrete)	28.0146	14.4803	29.5564	11.9878	15.3097	12.8914	12.4633	11.1301
Concrete Compressive Strength	Bayesian	110.9536	31.1148	90.8375	26.4231	42.8134	26.0185	26.3733	24.1065
	Random (Mixed)	102.3046	28.3509	63.4611	23.3345	33.9224	22.2093	22.2911	18.6243
	Random (Discrete)	102.3781	28.3509	82.1069	24.7829	38.9924	23.5297	23.4782	21.0347

[3] Numbers between 10^(-5) and 10^(0.5) with 0.5 step size
[4] Uniformly distributed numbers between 10^(-5) and 1
[5] 20 linearly spaced numbers between 0.01 and 5

Diabetes	Bayesian	2909.4917	3206.2951	4952.6782	3285.8783	3078.2827	2958.5136	2897.8583	2924.5858
	Random (Mixed)	2891.8013	3160.0031	5317.0768	3947.5332	3129.8194	2909.1599	2876.8613	2876.8485
	Random (Discrete)	2910.7529	3160.0031	5459.3905	3577.4024	3089.5494	2959.6912	2887.6833	2883.9718
Energy Efficiency	Bayesian	10.6303	3.3720	8.4001	2.1630	3.8987	2.1771	2.1738	1.7518
	Random (Mixed)	10.1312	2.8514	7.3845	1.3832	3.5220	1.3601	1.3832	1.2768
	Random (Discrete)	10.3757	2.8514	9.2307	1.6349	3.9489	1.5526	1.6349	1.5410
Forest Fires	Bayesian	5321.3113	5608.6035	5435.9285	7811.9329	5562.1396	5522.5083	5497.3752	5416.3224
	Random (Mixed)	6752.3485	7153.8120	6972.7771	9626.7345	7011.1712	6993.6214	6990.0076	6910.4169
	Random (Discrete)	6756.3504	7153.8120	6983.2544	9615.4962	7018.6002	6918.2962	6924.6925	6941.7147
Graduate Admissions	Bayesian	0.0048	0.0045	0.0053	0.0211	0.0061	0.0039	0.0043	0.0042
	Random (Mixed)	0.0039	0.0046	0.0060	0.0217	0.0059	0.0040	0.0040	0.0039
	Random (Discrete)	0.0068	0.0046	0.0061	0.0217	0.0070	0.0043	0.0046	0.0047
Wine Quality	Bayesian	0.4308	0.3711	0.4158	0.4209	0.3893	0.3698	0.3708	0.3526
	Random (Mixed)	0.4446	0.3868	0.4432	0.4254	0.4047	0.3885	0.3891	0.3695
	Random (Discrete)	0.4497	0.3868	0.4292	0.4327	0.4032	0.3903	0.3894	0.3718
Yacht Hydrodynamics	Bayesian	82.0460	1.5203	126.7806	1.5179	24.2880	1.4419	1.4686	0.9393
	Random (Mixed)	94.5628	1.6062	123.6544	1.3003	25.9503	1.3182	1.3279	0.8464
	Random (Discrete)	95.0700	1.6062	178.2943	1.4304	32.1412	1.2611	1.4066	1.1169

Comparing the results of three different search methods from Table.3, we observe that their results are very close to each other, and there is no clear dominating advantage in term of prediction error. The performance is data set specific.

Table.4 Comparing optimal hyperparameters of COWE and COWE-ITH for Airfoil Self-Noise data set

Hyperparameter	Ensemble Method	Random Search (Discrete)	Random Search (Mixed)	Bayesian Search
LASSO (*alpha*)	COWE	1.00E-05	0.042137	0.023439
	COWE-ITH	0.1	0.243908	0.053135
Random Forest (n_estimators)	COWE	200	200	200
	COWE-ITH	200	200	100
Random Forest (*max_depth*)	COWE	10	10	10
	COWE-ITH	10	10	10
XGBoost (*gamma*)	COWE	5	7.682878	10.65204
	COWE-ITH	5	7.682878	5.928025
XGBoost (learning_rate)	COWE	0.3	0.322892	0.269256
	COWE-ITH	0.3	0.322892	0.51186
XGBoost (n_ estimators)	COWE	150	50	150
	COWE-ITH	150	50	100
XGBoost (*max_depth*)	COWE	9	9	9
	COWE-ITH	9	9	9
SVM (*C*)	COWE	5	2.002395	4.120363
	COWE-ITH	0.272632	0.889269	1.430838
SVM (*gamma*)	COWE	0.26	0.240383	0.500007
	COWE-ITH	0.16	0.103484	0.038688

Table.4 demonstrates the different choices of hyperparameters as the optimal selections for creating optimal ensembles from COWE and COWE-IT for Airfoil Self-Noise data set (the same was observed for other data sets, but they are not shown here). Comparing the tuned hyperparameters

found by three mentioned search methods before creating ensembles, with the ones found by COWE-ITH, the main claim of this paper is proved to be true. The hyperparameters found to be optimal by COWE-ITH method which resulted in the best ensembles are different from the hyperparameters tuned separately (COWE). This means that in order to create better performing ensembles, the hyperparameters should not necessarily be the ones that are proved to be optimal independently. This addresses the third question from questions raised in the introduction section.

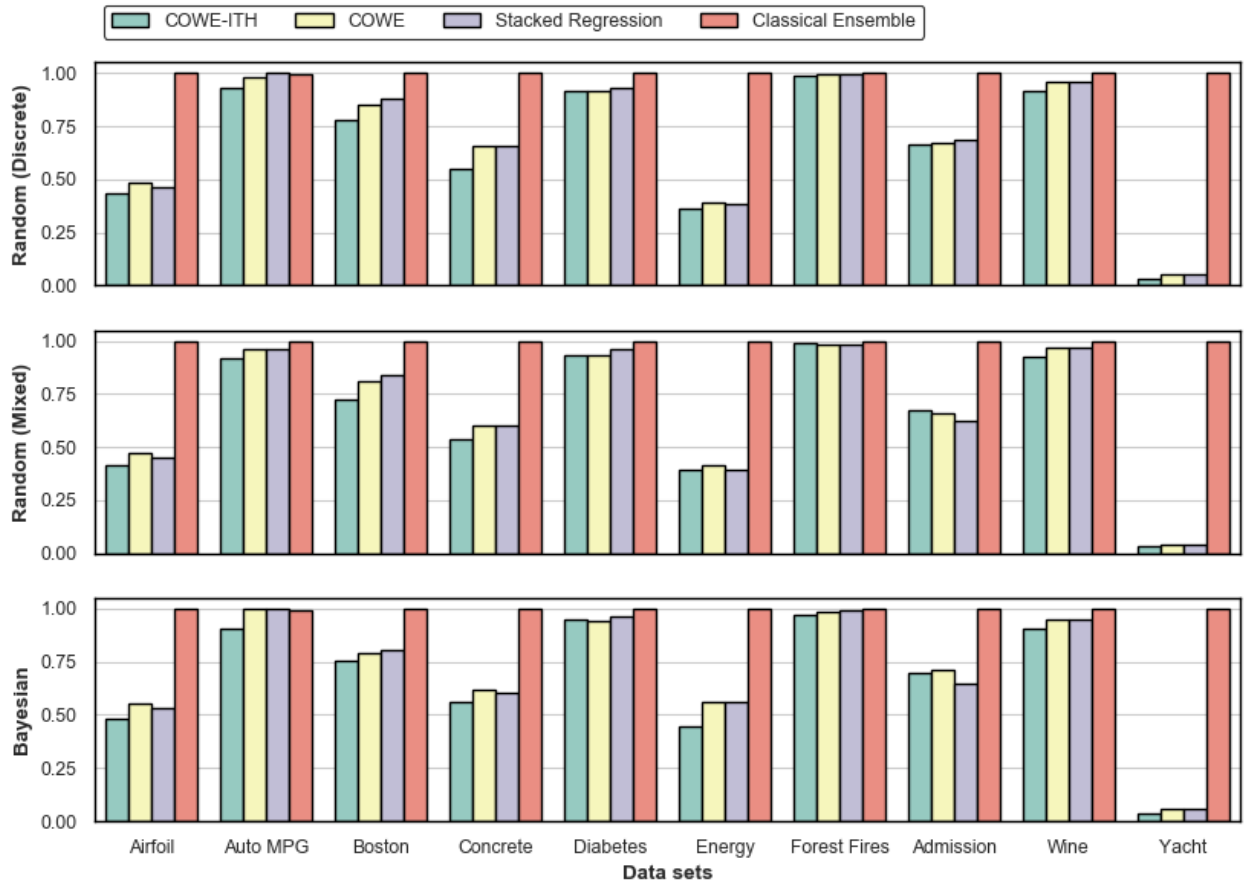

Fig.4 Comparison of the proposed method vs. benchmarks on 10 public data sets (average normalized test errors)

Fig.4 demonstrates the normalized error rates of data sets understudy for all ensemble models and search methods. It visualizes the comparison between COWE-ITH and the benchmarks. The figure shows almost complete dominance of COWE-ITH over the benchmarks addressing the second question raised in the introduction section. COWE-ITH has been the winner in 9 out of 10 public data sets. Furthermore, it can be seen that classical ensemble models with equal weights for the base learners are not performing as well. Their weakness is specifically evident in the results for the Yacht data set when the predictions of at least one of the base learners had not been helpful for ensembles. The classical ensemble model could not differentiate the base learners and assigned equal weights to all base learners, while COWE and COWE-ITH assigned zero weights as the optimal weight to the poorly predicted models.

Hence, it can be concluded that the proposed schemes (COWE and COWE-ITH) improve the prediction accuracy of each base learner. Comparing them to the common ensembling methods (classical ensembling and stacked regression), COWE-ITH could achieve better prediction accuracy among all, while introducing an improvement over successful COWE scheme. Therefore, this confirms the hypothesis that tuning hyperparameters of base learners inside optimal ensemble creating procedure will result in better prediction accuracy. These findings demonstrate the generalizability of COWE and COWE-ITH to real data sets since we have applied the methods on 10 publicly available data sets with diverse properties, which addresses the last question raised at the end of the Introduction section.

6. Conclusion

A systematic framework to build ensembles with optimized weights (COWE) for regression problems was introduced in this paper. The goal is to minimize bias and variance by combining different base learners using a nonlinear convex optimization model to find optimal weights. The model is capable of finding optimal ensemble weights that minimize both bias and variance of the predictions.

Moreover, in an attempt to observe the effect of tuning hyperparameters of base learners on the created ensembles, a bi-level nested algorithm that finds the optimal weights to combine base learners as well as the optimal set of hyperparameters for each of them (COWE-ITH) was designed in this study. The proposed methods were applied to ten public data sets and compared to other ensemble techniques. Based on the obtained results, it was shown that COWE-ITH is able to dominate other ensemble creation methods. Furthermore, it was demonstrated that the hyperparameters used in creating optimal ensembles are different when they are tuned internally with COWE-ITH algorithm, than when they are tuned independently.

This study is subject to a few limitations which suggest future research directions. Firstly, designing a bi-level nested algorithm for classification problems could expand the algorithm to classification problems and investigate its effectiveness on them. Secondly, applying a similar concept of hyperparameter tuning on other ensemble creating methods such as regularized stacking will more demonstrate the impact of hyperparameter tuning when creating ensembles. Lastly, trying to speed-up the ensemble creating process when considering hyperparameter tuning will create a competitive edge for the algorithm over competitions.

References

Acharya, M., Armaan, A., & Antony, A. (2019). A Comparison of Regression Models for Prediction of Graduate Admissions. IEEE International Conference on Computational Intelligence in Data Science 2019.

Belayneh, A., Adamowski, J., Khalil, B., & Quilty, J. (2016). Coupling machine learning methods with wavelet transforms and the bootstrap and boosting ensemble approaches for drought prediction. *Atmospheric research, 172*, 37-47.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research, 13*(Feb), 281-305.

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.

Bhasuran, B., Murugesan, G., Abdulkadhar, S., & Natarajan, J. (2016). Stacked ensemble combined with fuzzy matching for biomedical named entity recognition of diseases. *Journal of biomedical informatics, 64*, 1-9.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*: Cambridge university press.

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

Breskvar, M., Kocev, D., & Džeroski, S. (2018). Ensembles for multi-target regression with random output selections. [journal article]. *Machine Learning, 107*(11), 1673-1709.

Brooks, T. F., Pope, D. S., & Marcolini, M. A. (1989). Airfoil self-noise and prediction.

Brown, G. (2017). Ensemble Learning. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of Machine Learning and Data Mining* (pp. 393-402). Boston, MA: Springer US.

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

Conroy, B., Eshelman, L., Potes, C., & Xu-Wilson, M. (2016). A dynamic ensemble approach to robust classification in the presence of missing data. [journal article]. *Machine Learning, 102*(3), 443-463.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems, 47*(4), 547-553.

Cortez, P., & Morais, A. d. J. R. (2007). A data mining approach to predict forest fires using meteorological data.

Dietterich, T. G. (2000). *Ensemble methods in machine learning.* Paper presented at the International workshop on multiple classifier systems.

Dua, D., & Graff, C. (2017). UCI machine learning (2017). *URL http://archive.ics.uci.edu/ml*.

Efron, B., Hastie, T., Johnstone, I., & Tibshirani, R. (2004). Least angle regression. *The Annals of statistics, 32*(2), 407-499.

Ekbal, A., & Saha, S. (2013). Stacked ensemble coupled with feature selection for biomedical entity extraction. *Knowledge-Based Systems, 46*, 22-32.

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

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*(10), 993-1001.

Harrison Jr, D., & Rubinfeld, D. L. (1978). Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management, 5*(1), 81-102.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer, 27*(2), 83-85.

Hoch, T. (2015). *An Ensemble Learning Approach for the Kaggle Taxi Travel Time Prediction Challenge.* Paper presented at the DC@ PKDD/ECML.

Hong, T., Pinson, P., & Fan, S. (2014). Global energy forecasting competition 2012: Elsevier.

Hu, K., Huang, P., Chen, H., & Peng, Y. (2017). KDD CUP 2017 Travel Time Prediction Predicting Travel Time – The Winning Solution of KDD CUP 2017. KDD.

Jones, E., Oliphant, T., & Peterson, P. (2001). SciPy: Open source scientific tools for Python.

Kechyn, G., Yu, L., Zang, Y., & Kechyn, S. (2018). Sales forecasting using WaveNet within the framework of the Kaggle competition. *arXiv preprint arXiv:1803.04037*.

Khaki, S., & Khalilzadeh, Z. (2019). Classification of Crop Tolerance to Heat and Drought: A Deep Convolutional Neural Networks Approach. *arXiv preprint arXiv:1906.00454*.

Khaki, S., & Wang, L. (2019). Crop yield prediction using deep neural networks. *Frontiers in plant science, 10*.

Kim, D., Yu, H., Lee, H., Beighley, E., Durand, M., Alsdorf, D. E., et al. (2019). Ensemble learning regression for estimating river discharges using satellite altimetry data: Central Congo River as a Test-bed. *Remote sensing of environment, 221*, 741-755.

Koren, Y. (2009). The bellkor solution to the netflix grand prize. *Netflix prize documentation, 81*(2009), 1-10.

Large, J., Lines, J., & Bagnall, A. (2019). A probabilistic classifier ensemble weighting scheme based on cross-validated accuracy estimates. *Data Mining and Knowledge Discovery*, 1-36.

Martelli, P. L., Fariselli, P., & Casadio, R. (2003). An ENSEMBLE machine learning approach for the prediction of all-alpha membrane proteins. *Bioinformatics, 19*(suppl_1), i205-i211.

Niculescu-Mizil, A., Perlich, C., Swirszcz, G., Sindhwani, V., Liu, Y., Melville, P., et al. (2009). *Winning the KDD cup orange challenge with ensemble selection.* Paper presented at the KDD-Cup 2009 Competition.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research, 12*(Oct), 2825-2830.

Pham, H., & Olafsson, S. (2019). Bagged ensembles with tunable parameters. *Computational Intelligence, 35*(1), 184-203.

Pham, H., & Olafsson, S. (2019). On Cesaro averages for weighted trees in the random forest. *Journal of Classification*, 1-14.

Puurula, A., Read, J., & Bifet, A. (2014). Kaggle LSHTC4 winning solution. *arXiv preprint arXiv:1405.0546*.

Quinlan, J. R. (1993). *Combining instance-based and model-based learning.* Paper presented at the Proceedings of the tenth international conference on machine learning.

Shahhossseini, M., Hu, G., & Pham, H. (2019). Optimizing Ensemble Weights for Machine Learning Models: A Case Study for Housing Price Prediction. Nanjing, China: Proceedings of 2019 INFORMS International Conference on Service Science, Available at: http://works.bepress.com/mohsen-shahhosseini/1/.

Shen, Z.-Q., & Kong, F.-S. (2004). *Dynamically weighted ensemble neural networks for regression problems.* Paper presented at the Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826).

Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical bayesian optimization of machine learning algorithms.* Paper presented at the Advances in neural information processing systems.

Sutton, C., Ghiringhelli, L. M., Yamamoto, T., Lysogorskiy, Y., Blumenthal, L., Hammerschmidt, T., et al. (2018). NOMAD 2018 Kaggle Competition: Solving Materials Science Challenges Through Crowd Sourcing. *arXiv preprint arXiv:1812.00085*.

Taieb, S. B., & Hyndman, R. J. (2014). A gradient boosting approach to the Kaggle load forecasting competition. *International journal of forecasting, 30*(2), 382-394.

Tsanas, A., & Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings, 49*, 560-567.

Töscher, A., & Jahrer, M. (2008). The bigchaos solution to the netflix prize 2008. *Netflix Prize, Report*.

van Rijn, J. N., Holmes, G., Pfahringer, B., & Vanschoren, J. (2018). The online performance estimation framework: heterogeneous ensemble learning for data streams. [journal article]. *Machine Learning, 107*(1), 149-176.

Wang, H., Xu, Q., & Zhou, L. (2015). Large unbalanced credit scoring using Lasso-logistic regression ensemble. *PloS one, 10*(2), e0117844.

Winham, S. J., Freimuth, R. R., & Biernacka, J. M. (2013). A weighted random forests approach to improve predictive performance. *Statistical Analysis and Data Mining: The ASA Data Science Journal, 6*(6), 496-505.

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

Yeh, I.-C. (1998). Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete research, 28*(12), 1797-1808.

Yu, H.-F., Lo, H.-Y., Hsieh, H.-P., Lou, J.-K., McKenzie, T. G., Chou, J.-W., et al. (2010). *Feature engineering and classifier ensemble for KDD cup 2010.* Paper presented at the KDD Cup.

Yu, L., Lai, K. K., Wang, S., & Huang, W. (2006). *A bias-variance-complexity trade-off framework for complex system modeling.* Paper presented at the International Conference on Computational Science and Its Applications.

Zhang, C., & Ma, Y. (2012). *Ensemble machine learning: methods and applications*: Springer.

Zhang, X., & Mahadevan, S. (2019). Ensemble machine learning models for aviation incident risk prediction. *Decision Support Systems, 116*, 48-63.

Zou, H., Xu, K., Li, J., & Zhu, J. (2017). The Youtube-8M kaggle competition: challenges and methods. *arXiv preprint arXiv:1706.09274*.