A gradient boosting approach with diversity promoting measures for the ensemble of surrogates in engineering

Nestor V. Queipo and Efrain Nava

Applied Computing Institute, Faculty of Engineering

University of Zulia, Venezuela

Abstract

Ensemble of surrogates have shown to be effective in the modeling of a variety of engineering problems, from aerospace to automotive and oil and gas industries, among many others. The ensemble performance though can be negatively affected by the lack of diversity (correlated errors) among the members of the ensemble, i.e., similar errors throughout the input space. During the last decade ensemble research efforts in engineering have mostly focused on the so-called ensemble integration (combining prediction) phase with, for example, linear weighted averages of surrogates predictions based on performance (error) measures of both local and global nature. On the ensemble generation side (members of the ensemble), the emphasis has been on promoting diversity through so-called heterogeneous ensembles, that is the use of different learners (surrogates), such as, linear regression, radial-basis functions, kriging, support vector regression and neural networks. Because of the lack of control on diversity in heterogeneous ensembles, potentially correlated errors among the learners remains a critical issue.

This paper presents an alternative approach based on homogeneous ensembles (single learning algorithm) that promotes diversity both implicitly and explicitly, by randomly sampling the training data, and deterministically changing the data supplied to each surrogate model in the ensemble generation process. Specifically, it includes decision trees as learners, subsampling of the training set and training on random selections of the input features, i.e., random subspace. The ensemble is sequentially constructed as an additive model where at each stage a simple surrogate is fit to the negative gradient of the loss function at the training data of the latest ensemble (gradient boosting). In addition, as a byproduct of the modeling process, the proposed approach provides importance measures to preliminary rank input features. Statistically significant prediction improvements over heterogeneous ensembles were observed in a variety of well-known analytical test functions and industrial case studies in the areas of petroleum engineering-net present value of oil production in a mature reservoir and structural analysis-FSAE Brake pedal von Mises stress and buckling load factor.

1. Introduction

Ensemble of surrogates (e.g., Bishop 1995, Zerpa et al. 2005, Goel et al. 2007) refers to methods that generate several models which are combined to make a prediction in regression problems. In particular, they can be shown to be: i) statistically more accurate and robust than any single surrogate (hypothesis), and, ii) less effective when the models in the ensemble exhibit significantly correlated errors (similar errors throughout the input space). The former observation have led to applications in a variety of engineering domains, and, ensembles becoming a fixture among the winner approaches of data science competitions (e.g., www.kaggle.com). The latter (ii) highlights the need to promote diversity during the ensemble generation process (Bishop 1995; Brown et al. 2005a; Reeve and Brown 2018). Successful application domains include, but are not limited to, aerospace (Tenne 2013), automotive (Hamza and Saitou 2012; Fang et al. 2014), and oil and gas (Chen et al. 2007; Sanchez et al. 2008).

During the last decade significant progress has been made in the ensemble of surrogates for engineering, in particular, in the so-called area of *ensemble integration* (combining predictions), which has focused on linear weighted averages of surrogates predictions based on performance (error) measures that are either, static/global (Goel et al. 2007; Acar and Rais-Rohani 2009), dynamic/local (Zerpa et al. 2005; Sanchez et al. 2008; Zhang et al. 2012, Yin et al. 2018), or hybrid (Chen et al. 2018). In all instances, the surrogates in the ensembles were designed to have low bias (accurate), and to be diverse by adopting different learning algorithms (*heterogeneous ensembles*). For example, considering static integration, Acar and Rais-Rohani (2009) used ensembles of five different surrogates, namely, polynomial regression (PRS), radial basis functions (RBF), Kriging (KR), Gaussian process (GP), and support vector regression (SVR). Audet et al. 2018, on the other hand, introduced an order-based error tailored to surrogate-based search, considering PRS, RBF, and Kernel Smoothing (KS) models. In contrast, Yin et al. (2018) and Chen et al. (2018) reported ensembles with dynamic integration combining typical metamodels, namely, PRS, RBF and KR. Heterogeneous ensembles, tough, lack control of diversity with previous studies of ensembles in engineering not explicitly testing the level of correlation among the surrogates in the *ensemble generation* phase. To avoid this shortcoming, a variety of approaches have been proposed, most notably, the so-called overproduce-and-choose with pruning process (Rooney et al. 2004; Mendes-Moreira et al. 2012) and as discussed by Bishop (1995) by accounting for the correlation among the members of the ensemble (Viana et al. 2014).

Alternatively, in homogeneous ensembles (where all the surrogates share the same learning algorithm) diversity can be more systematically controlled by randomly sampling the training data (implicit methods) or deterministically manipulating the ensemble generation process (explicit methods). Randomly sampling the data so that the surrogate models are trained with different sets of data can be very effective by, for example, sampling with replacement, i.e., bagging (Breiman 1996), and the subsampling of the training set or training on random samples of input features, i.e., random subspace (Ho 1998). On the other hand, diversity can be promoted explicitly by sequentially (Rosen 1996) or simultaneously (Liu and Yao 1999) generating models which deliberately penalize the correlation between members of the ensemble in the loss function. In addition, an ensemble can be constructed sequentially as an additive model where at each stage a simple surrogate (boost) is fit to the negative gradient of the loss function at the training data such that a particular cost function is minimized; for example, for a quadratic loss function the cited gradient can be shown to be the residual of the latest ensemble. Note that the contribution (weight) of each base learner to the prediction is always dynamically established, that is, it varies with location. This latest approach is known as gradient boosting (Shapire 1990; Freund 1995), a form of functional gradient descent (Breiman 1999; Friedman 2001).

Furthermore, when using ensembles and, in general, in surrogate-modeling, it is often of interest to address the so called feature selection problem. The feature selection problem makes reference to the removal of irrelevant or redundant (non predictive information) inputs in the training data for faster and more effective learning. More precisely, features with higher ranking or importance scores may be chosen for further investigation or for building a more parsimonious model. A frequently adopted approach for feature selection is the Sobol's method (Sobol 1993), a variance-based technique used with a trained model (as a separate task) that determines the contribution of each feature and their interactions to the overall variance of the target variable. Notably, a particular form of surrogates known as decision trees can provide importance measures to preliminary rank input variables features as a byproduct of the modeling process (Karagiannopoulos et al. 2007; Kazemitabar et al., 2017).

This paper presents: i) the bias/variance/covariance formulation to raise awareness of the need to address potential correlation among surrogates in heterogeneous ensembles, ii) an ensemble approach based on gradient boosting and decision trees with diversity promoting measures for solving engineering modeling problems, and iii) the relative performance of the proposed approach with respect to heterogeneous ensembles on selected case studies. The remainder of this paper is structured as follows: problem statement (Section 2), bias/variance/covariance formulation (Section 3), heterogeneous ensembles and correlation among surrogates (Section 4), decision trees for regression (Section 5), a gradient boosting approach for the ensemble of surrogates (Section 6), case studies (Section 7), results and discussion (Section 8), and conclusions and recommendations (Section 9).

2. Problem statement

The problem of interest is one of typical regression, that is, given a training set D consisting of n instances, such as $\mathrm{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$, where x_i is an instance of a vector of features and y_i is a target, the task is to learn an approximate function $\hat{f}: \mathrm{X} -> \mathrm{R}$ of the true function f from D. The approximate function $\hat{f}$ can be a single surrogate f_i or the result of a linearly weighted sum of M individual surrogates F. The function $\hat{f}$ is estimated by optimizing a loss function typically mean square error considering out-of-fold predictions.

3. Bias, variance and covariance decomposition

For an ensemble F (with constant weights) it can be shown (Ueda and Nakano, 1996) that the expected value of the MSE-mean squared error (generalization error) can be expressed as Equation (1):

$$E\{(F-y)^2\} = \overline{bias}^2 + \frac{1}{M}\,\overline{var} + (1-\frac{1}{M})\overline{covar} \tag{1}$$

where,

$$\overline{bias} = \frac{1}{M}\sum_{i}\left(E\{f_i\} - y\right)^{2} \tag{2}$$

$$\overline{var} = \frac{1}{M}\sum_{i} E\left\{(f_i - E\{f_i\})^2\right\} \tag{3}$$

$$\overline{covar} = \frac{1}{M(M-1)}\sum_{i}\sum_{j\neq i} E\left\{(f_i - E\{f_i\})(f_j - E\{f_j\})\right\} \tag{4}$$

where, var denotes the variance of the individual surrogates in the ensemble, and $covar$ the averaged covariance of the surrogates in the ensemble; the estimates of all the components are computed considering several random trials (training sets). Note that in this context diversity between surrogates (differences in the predictor outputs) is measured by their pair-wise covariance. This decomposition tells us that the generalization error depends not only on the bias and variance of the individual surrogates but also on the covariance between the surrogates. The sought ensemble (optimum generalization error) should account for all components, deliberately promoting diversity among surrogates while minimizing bias.

Figure 1 reports the generalization error and bias, variance and covariance decomposition for an heterogeneous ensemble of surrogates designed to approximate the well-known Rosenbrock 6D test function (Section 6), considering sixty (60) training data points and an one hundred and fifty (150) random trials. Details of the ensemble approximation are available in the next section (Section 4). As expected, the ensemble provides a significantly lower generalization error than the average of the individual surrogates through a reduction of the error variance. Most notably, the often ignored covariance component is shown as a prominent contributor to the generalization error, only second to the bias error.

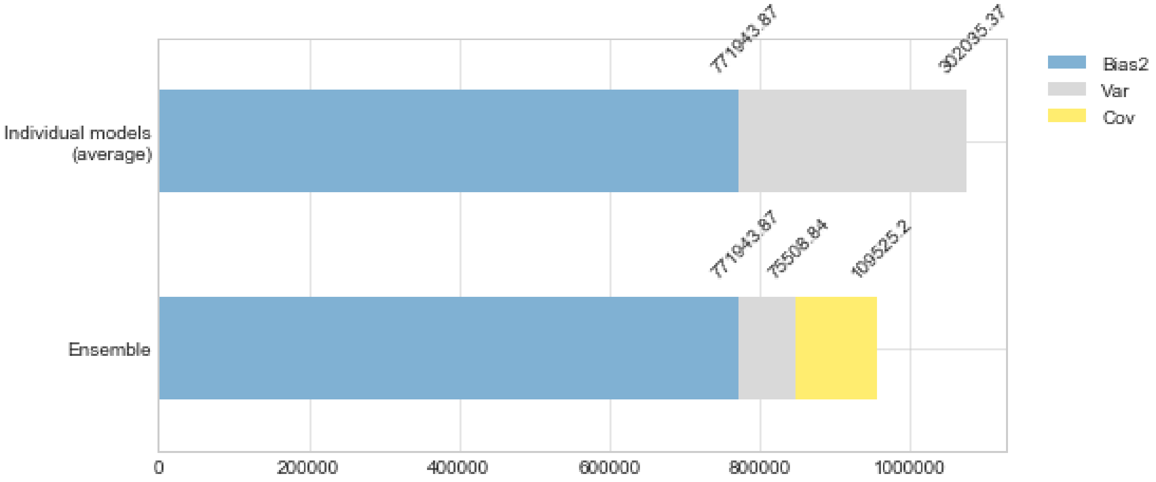


Figure 1. Squared error bias (squared bias – Bias2), variance (Var) and covariance (Cov) decomposition for an heterogeneous ensemble of surrogates – Rosenbrock 6D test function.

4. Heterogeneous ensembles and correlation between surrogates

These ensembles combine a set of heterogeneous surrogates, i.e., generated by different learning algorithms L_i on a single dataset D; each of the surrogates have their hyper parameters that must be chosen among a given set of candidates (hyper-parameters grid), for example, the number of neurons in the hidden layer in a multilayer perceptron neural network (MLP). In this study, we focus on ensembles based on linear weighted averages of surrogate predictions considering static/global performance measures (e.g., MSE, R-square). In abstract terms. their basic construction strategy can be described using the following steps:

1. Specify the set of possible hyper-parameters considering all the members of the ensemble
2. Split dataset D into k folds, with training sets D^{-i} corresponding to D – fold i, and $i = 1,2, \dots, k$
3. For each of the elements in the set of hyper-parameters:
    a. Generate a set of M base-level learners (surrogates) $f_1, \dots, f_M$ using (inner) cross-validation
    b. Make out-of-fold predictions with the M trained base-level learners (surrogates)
    c. Train (identify the weights) a meta-level linear regressor using the outputs (predictions) from step 3.b and the corresponding target values
4. Select the meta-level linear regressor among those obtained in 3.c corresponding to the lowest (outer) cross-validation error computed on D^i

Figure 2 illustrates the training process for the ensemble using (inner) cross-validation. The features of the training set for the meta-regressor are P_1, P_2,.., P_M, that is, the out-of-fold predictions (P_i) of the base-level learners (f_i). The Appendix: Metamodels presents a brief description of each of the surrogates under consideration, a combination of parametric and non-parametric models, namely: linear regression (13.1), multilayer perceptron neural networks (13.2), support vector regression-RBF (13.3) and Gaussian processes (13.4).

In general, though, there has been no explicit evaluation of the correlation among the individual surrogates which, as previously discussed may compromise the generalization errors of heterogeneous ensembles. An illustration of how members of the ensembles can in fact be correlated follows.

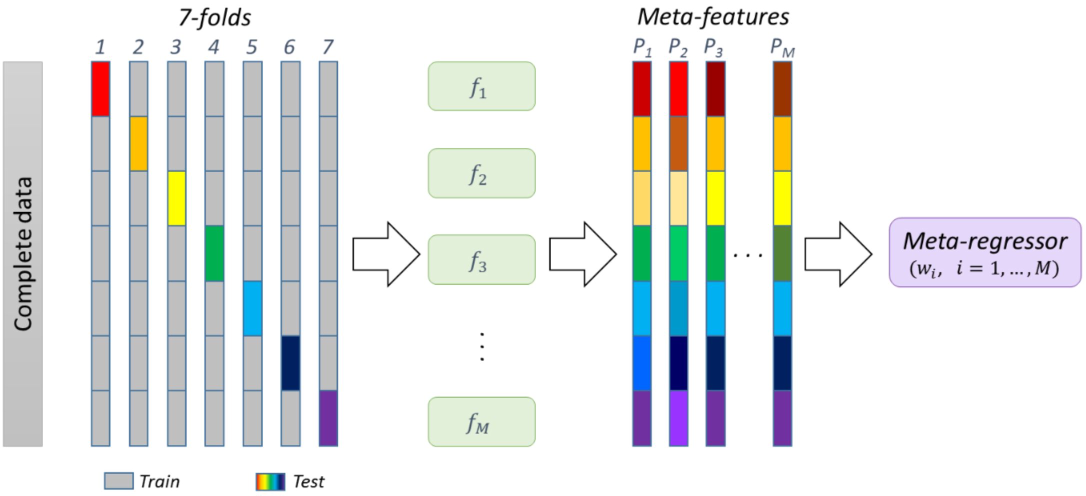


Figure 2. Heterogeneous ensemble generation process illustrated with 7-fold cross-validation and M surrogates to construct a meta-level data (out-of-fold) set of predictions. Once the meta-level data set is available a linear regressor can be fitted; the regression coefficients represent the relative contribution of the individual surrogates to the ensemble prediction.

The correlation between surrogates f_i and f_j can be estimated through the expression in Equation (5).

$$corr\ (\boldsymbol{x}, y, f_i, f_j) = \frac{E\{(y - f_i)(y - f_j)\}}{\sigma(f_i)\sigma(f_j)} \tag{5}$$

Using the heterogeneous ensemble cited in the previous section (Rosenbrock 6D test function), Figure 3 (a) displays the empirical distribution (box-plots) of correlation values for different pairs of surrogates in one hundred and fifty (150) random trials. Note that two (SVR-Gaussian process and Linear regression-Multilayer perceptron regression) out of the six possible pairs exhibit considerably high median correlation values (over 0.75). Details of the empirical distribution observed between Linear regression - Multilayer perceptron regression are available in Figure 3 (b). It further illustrates how different learning algorithms in heterogeneous ensembles do not necessary lead to the desired outcome – negative or non-correlated models.

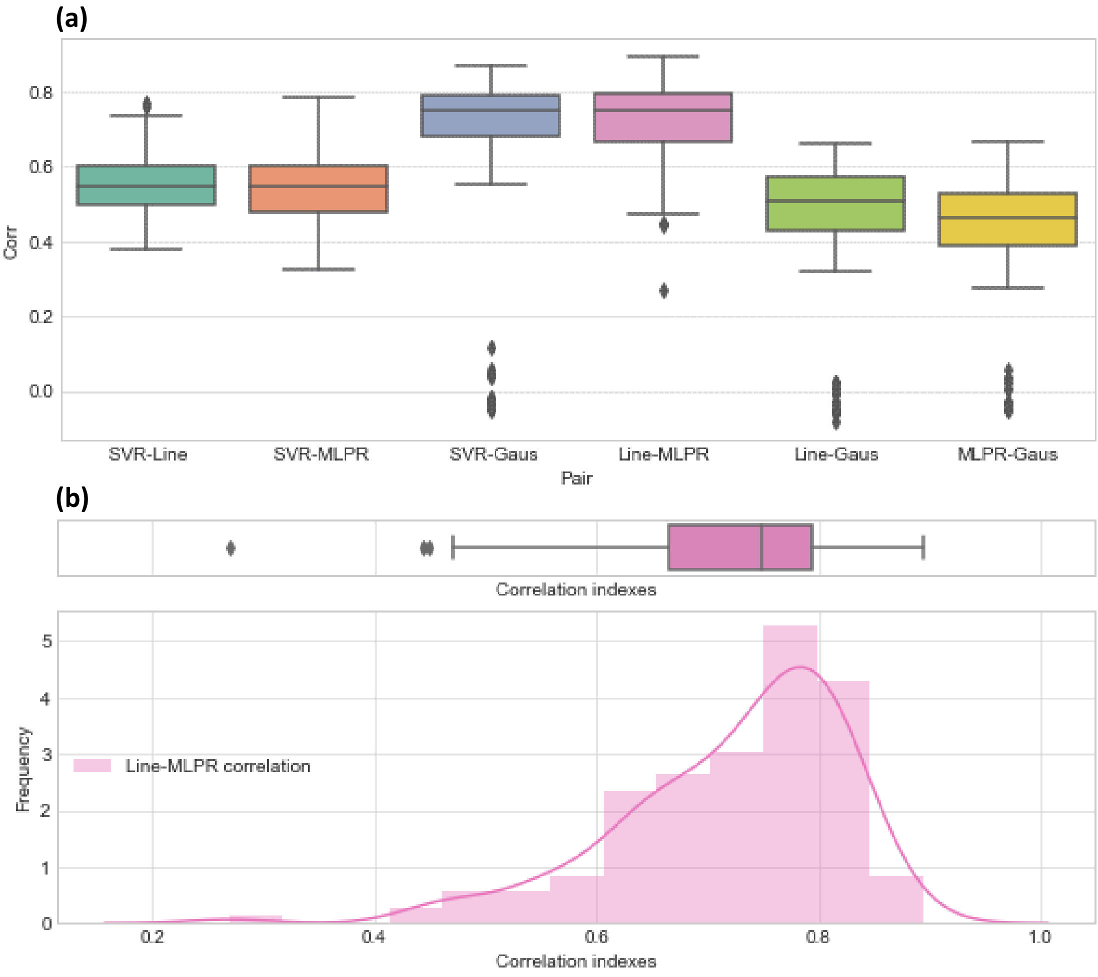


Figure 3. Correlations among models of the ensemble (a) and correlations of LinearRegression-MLPR, the two most correlated models (b) – Rosenbrock 6D test function. In this context, the labels Line, MLPR and Gauss, refer to linear, multilayer perceptron and Gaussian process regressors, respectively.

5. Decision trees for regression

A decision tree learner, a non-parametric supervised method, builds regression models in the form of a tree structure (Breiman et al. 1984). The goal: predict the value of a target variable by learning simple decision rules inferred from the data features. Starting from the *root* (with the whole training data), datasets are broken down (split) at *decision nodes* into smaller and smaller subsets considering at each split the feature with maximal reduction of a weighted (with the number of samples) variability measure (e.g., mean square error). After a termination criterion is met, e.g., a pre-specified maximum depth is reached, the result is a tree with decision and leaf nodes. A decision node has two (binary) branches, each associated with a set of values for the tested feature, while a leaf node represents a prediction (average value of the samples at the node) of the numerical target, i.e., function output. Figure 4 illustrates a decision tree with root, decision and leaf nodes for a single variable (feature) function; note that at each decision node a splitting led to a maximum reduction in the weighted standard deviation of the data.

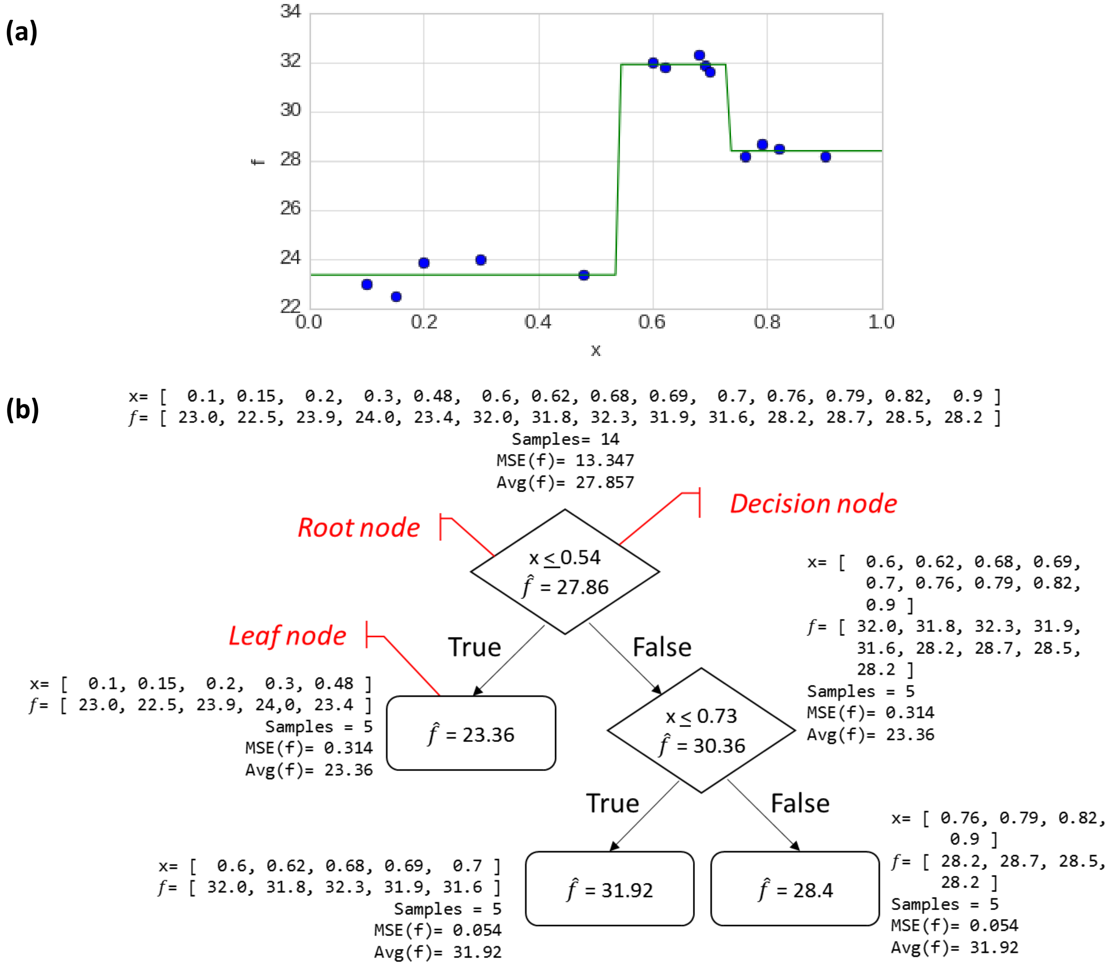

(a)
34
32
30
28
26
24
22
f
0.0
0.2
0.4
0.6
0.8
1.0
x
(b)
x= [0.1, 0.15, 0.2, 0.3, 0.48, 0.6, 0.62, 0.68, 0.69, 0.7, 0.76, 0.79, 0.82, 0.9]
f= [23.0, 22.5, 23.9, 24.0, 23.4, 32.0, 31.8, 32.3, 31.9, 31.6, 28.2, 28.7, 28.5, 28.2]
Samples= 14
MSE(f)= 13.347
Avg(f)= 27.857
Root node
Decision node
x ≤ 0.54
f̂ = 27.86
x= [0.6, 0.62, 0.68, 0.69, 0.7, 0.76, 0.79, 0.82, 0.9]
f= [32.0, 31.8, 32.3, 31.9, 31.6, 28.2, 28.7, 28.5, 28.2]
Samples = 5
MSE(f)= 0.314
Avg(f)= 23.36
Leaf node
True
False
x= [0.1, 0.15, 0.2, 0.3, 0.48]
f= [23.0, 22.5, 23.9, 24,0, 23.4]
Samples = 5
MSE(f)= 0.314
Avg(f)= 23.36
f̂ = 23.36
x ≤ 0.73
f̂ = 30.36
True
False
x= [0.76, 0.79, 0.82, 0.9]
f= [28.2, 28.7, 28.5, 28.2]
Samples = 5
MSE(f)= 0.054
Avg(f)= 31.92
x= [0.6, 0.62, 0.68, 0.69, 0.7]
f= [32.0, 31.8, 32.3, 31.9, 31.6]
Samples = 5
MSE(f)= 0.054
Avg(f)= 31.92
f̂ = 31.92
f̂ = 28.4

Figure 4. A decision tree estimation from training sample (a); the corresponding decision tree is also shown (b). Note root, decision and leaf nodes (predictions)

Algorithm

A decision tree for regression can be obtained through the following steps (with mean square error adopted as a variability measure):

I. The whole training data set is considered as the root; compute the mean square error of the outputs in the training data set.

II. The data set is split considering the different attributes into child nodes. At this step, for each of the features under consideration, the data set is split and the mean square error for the data in the child nodes is calculated and weighted with the number of samples in each node. The resulting mean square error—weighted sum of MSE among the two branches, is subtracted from the one before the split, and hence the MSE reduction is obtained. The standard candidate split points are those half-way between consecutive pairs of the sorted input data; see Fayyad and Irani (1992) for a more efficient approach.

III. Feature selection for decision nodes. Considering the features with the largest MSE reduction are associated with decision nodes, a measure of the importance of a particular feature can then be calculated as the sum of the MSE reductions over all nodes where a split was made on that feature; note that the MSE reductions are weighted to account for the size (number of samples) of the node. Important features can appear at different depths of the trees since a particular splitting variable may or not be good in terms of overall accuracy.

IV. Recursively process (II-III) non-leaf nodes until a termination criterion is met. Each branch in the previous step leads to another decision node (further splitting will be conducted) or a leaf node (e.g., MSE below a threshold value). In practice a termination criterion is needed; for example, when the MSE for a branch is a small fraction (e.g., 5%) of the MSE associated with the training data (root node) or the number of samples in the leaf nodes reach a minimum value (e.g., 3). When there is more than one sample in a leaf node, the prediction for the output is the average of the sample values.

Decision tress have several advantages; for example, it allows for both discrete and continuous variables, naturally couples model training with preliminary feature importance, and represents a white-box model easy to understand and interpret. Unfortunately, they can create over-complex models that do not generalise the data well (overfitting), are based on a greedy algorithm where locally optimal decisions are made at each node which can result in far from optimal trees, and small variations in the data might result in very different outcomes. Figure 5 shows the Branin and Hoo 2D original test function (a), the decision tree-weak learner approximation (b), and the corresponding decision tree-weak learner (c) obtained using a training sample size of twenty (20) and a maximum depth of three (*weak* learner). The shortcoming of decision trees are addressed by sequentially training multiple (weak) trees in an ensemble learner; details of such an approach are next.

6. A gradient boosting approach for the ensemble of surrogates

The proposed approach, labeled as GBwDM –Gradient boosting with diversity measures–, is designed to promote diversity among base learners, hence reducing the covariance component of the generalization error and potentially improving prediction estimates, be robust to overfitting and, as a byproduct of the modeling process, generate preliminary importance measures of the feature variables. We claim these objectives can be achieved by adopting a homogeneous ensemble approach based on decision trees and gradient boosting considering subsampling and random subspace strategies. The architecture (components and interactions) of the approach is described below.

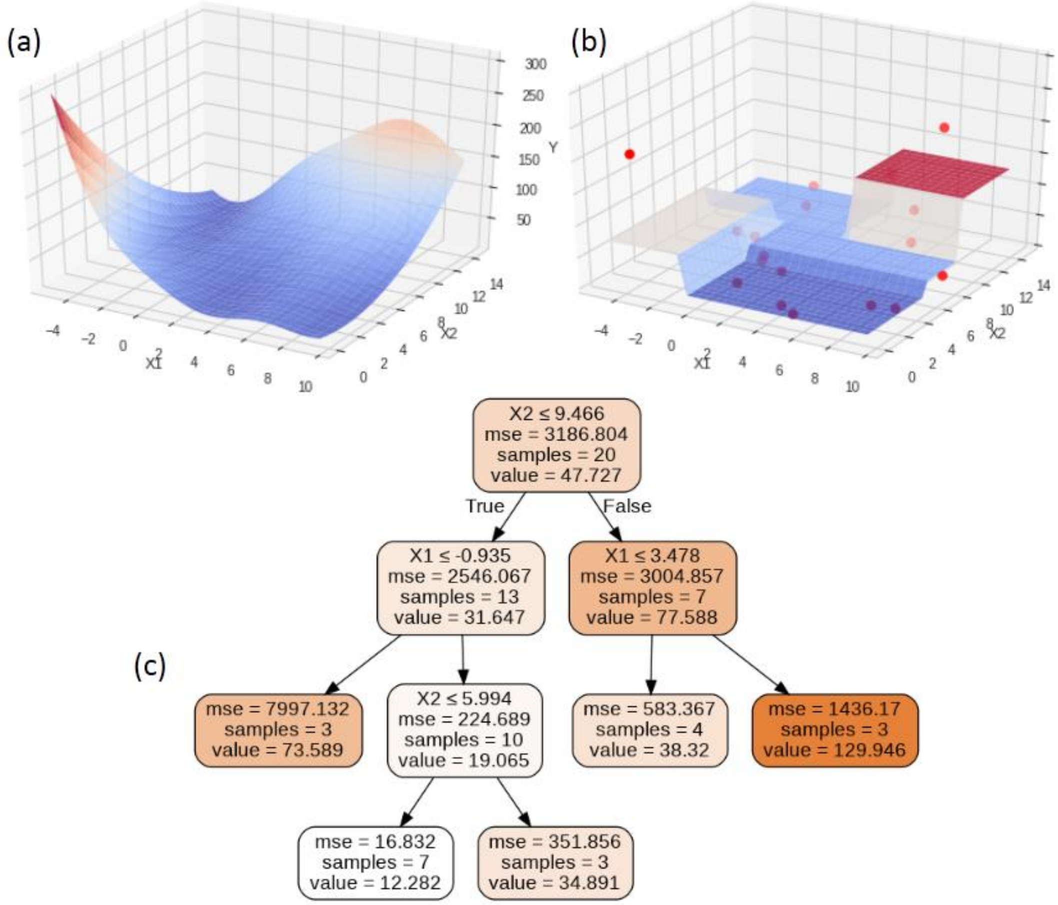


Figure 5. Branin and Hoo 2D modeling using a decision tree: (a) original test function, (b) decision tree-weak learner approximation, and (c) corresponding decision tree-weak learner

I. Hyper-parameters setting. In this stage a set of hyper-parameters for the ensemble of surrogates is defined. In the case of ensembles of decision trees using gradient boosting, hyper-parameters are, for example: number of trees, maximum depth in trees, subsample and random subspace values, number of estimators and learning rate. In the spirit of implicitly promoting diversity, subsampling makes reference to fitting the base learners with a fraction (subsample value) of the training data and the random subspace value specifies the number of features to consider when looking for the best split. The hyper-parameters are selected from a grid of pre-specified values (grid search) using k-fold cross-validation.

II. Ensemble generation. Figure 6 shows the basic steps of using the gradient boosting algorithm with a quadratic loss function. Note it builds the ensemble by sequentially fitting a base learner (each function increment is called a boost), e.g., a *weak* decision tree, to the residuals of the latest ensemble $\left(y_i - F_{m-1}(x_i)\right)$, that is, the negative gradient of the loss function at the training data (explicitly promoting diversity). *Weak* decision trees, also called *stumps,* refer to models where a rough approximation is sought and often translates to limiting the trees depth to a very low number (often 2 to 3).

Algorithm – Simple boosting algorithm for regression

```
Inputs:   Training set = $\{(x_1, y_1), \dots, (x_n, y_n)\}$
          $M$ = number of sub-models (base learners) of the  ensemble
Output:   $F_M$ = the ensemble
let $h_0$ be a constant model
let $F_0$ be an ensemble just consisting of $h_0$
for $m = 1, \dots, M$
    for each pair $(x_i, y_i)$ in the training set
        compute the residual $R$ (negative gradient)
        $R(y_i, F_{m-1}(x_i))$
    end for
    train a regression sub-model $h_m$ on the residuals
    add $h_m$ to the ensemble and find the optimal
    parameter $\alpha$ for each of leaf of $h_m$;
    $F_m = F_{m-1}(x) + \alpha_m \cdot h_m(x)$
end for
return the ensemble $F_M$
```

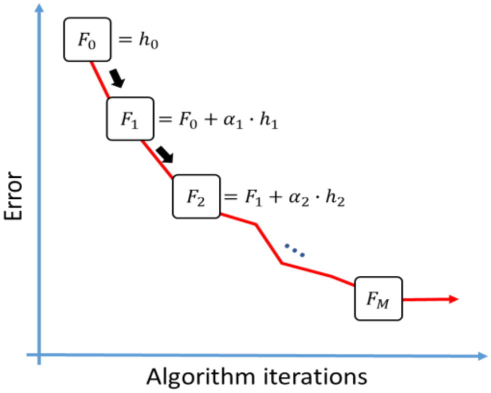

F_0 $= h_0$
F_1 $= F_0 + \alpha_1 \cdot h_1$
F_2 $= F_1 + \alpha_2 \cdot h_2$
F_M
Error
Algorithm iterations

Figure 6. Basic boosting algorithm for building ensembles assuming a quadratic loss function. The initial constant model F_0 can be, for example, the mean of the training target values.

While in this study the base learners are decision trees and the loss function is quadratic, gradient boosting is not limited to a particular surrogate or loss function. More generally, at each iteration the base learners h_m are trained on so called pseudo-residuals, which represents the negative gradient -Equation (6)- of the loss function at training points,

$$\nabla L = \frac{\partial L\big(y_i,\ F_{m-1}(x_i)\big)}{\partial F_{m-1}(x_i)} \tag{6}$$

That is, the loss reduction for each training point if the predicted value $-F_{m-1}(x_i)-$ were to become one unit closer to the target value. While the base learner loses accuracy at the training points, it provides an approximation of the cited gradient throughout the input space $-h_m(x)-$ which is used to implement a greedy stagewise strategy for calculating the latest ensemble –$F_m(x)$–, i.e., $F_m(x) = F_{m-1}(x) + \alpha_m \cdot h_m(x)$. Finally, at each leaf, the optimal parameter α is sought so that $L\big(y_i,\ F_m(x_i)\big)$ is minimized.

While the number of base learners are a parameter to be set, the gradient boosting ensembles have been shown to be resistant to overfitting so it can be specified quite robustly without jeopardizing predictive performance. An efficient and scalable implementation of the gradient boosting framework known as eXtreme Gradient Boosting, aka xgboost is also available (Chen and Guestrin 2016).

III. Ensemble selection. Considering the set of hyper-parameters in (I), the best ensemble is the one with the lowest score in holdout or nested cross-validation (Cawley and Talbot 2010). When holdout (data split intro training and test sets) is not used, nested cross-validation should be implemented so that the same data is not used to both tune the ensemble hyper-parameters and evaluate the ensemble performance – leading to over-fitting and underestimation of the prediction errors. Nested CV effectively uses a series of train/validation/test set splits, where in the inner loop, ensembles are fitted to training data and the best set of hyper-parameters is the one associated with the ensemble with the lowest score over the validation set. In the outer loop, the best ensemble is identified as the one with the lowest (average) estimated generalization error over test sets. The particular approach to ensemble selection should be identified depending on the data size, and model stability, among other factors.

IV. Features importance. Feature importance refers to the reduction in model accuracy when a variable is removed, and it is critical for variable selection and the building of parsimonious and interpretable models. As previously mentioned, decision trees can naturally couple the training process with preliminary feature importance identification at no additional computational expense. In this study, the overall importance score for a particular feature is the average of the ones obtained in the base learners (stumps) in the ensemble.

7. Case studies

The proposed approach is evaluated using three analytical test functions, namely, Branin & Hoo 2D (Dixon and Szegö 1978), Rosenbrock 6D (Rosenbrock 1960) and Dixon & Price 12D (Dixon and Price 1989) and two industrial case studies in the areas of engineering modeling, i.e., the von Mises stress and buckling load factor in a Formula SAE[1] brake pedal (structure analysis) as discussed in Romero and Queipo (2017) and the net present value of the twenty-year oil production of a mature oil reservoir considering alternative new well locations (reservoir engineering).

7.1. General considerations

Table 1 list the ensembles and individual surrogate models (base learners) under consideration and the corresponding computer implementations (libraries). The ensembles are identified as Gradient Boosting with Diversity Measures (GBwDM-proposed approach) and heterogenous ensemble (HEA); the individual members of the heterogeneous ensemble are also indicated, i.e., linear regression, SVR – radial basis function, neural networks, and Gaussian process. All the results were obtained using the cited libraries within Python programs in Google's Colaboratory environment – a free Jupyter notebook environment than runs entirely in the Cloud. Jupyter is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text (www.jupyter.org).

[1] Formula SAE is a student design competition organized by SAE International, previously known as the Society of Automotive Engineers, SAE.

Table 1. Modeling techniques (including the proposed approach) under consideration and the corresponding computer implementations (libraries)

Model	Implementation
Ensembles	
GBwDM (proposed approach)	Scikit-learn http://scikit-learn.org/stable/
Heterogeneous ensemble approach (HEA)	MLxtend https://rasbt.github.io/mlxtend/
Base learners	
Linear regression	Scikit-learn http://scikit-learn.org/stable/
SVR – Radial basis function	
Multi-layer perceptron	
Gaussian Process	

All models were trained/tested using the holdout cross-validation approach; the general hyper-parameters (Table 2) of all the models were identified using 5-fold cross-validation within the training set, and their values were the result of a preliminary study and adopting common practices. For all case studies the size of the training sets was specified as 10 times the number of dimensions and seeking sample-independent results, one hundred and fifty (150) random trials of the training/test sets were conducted.

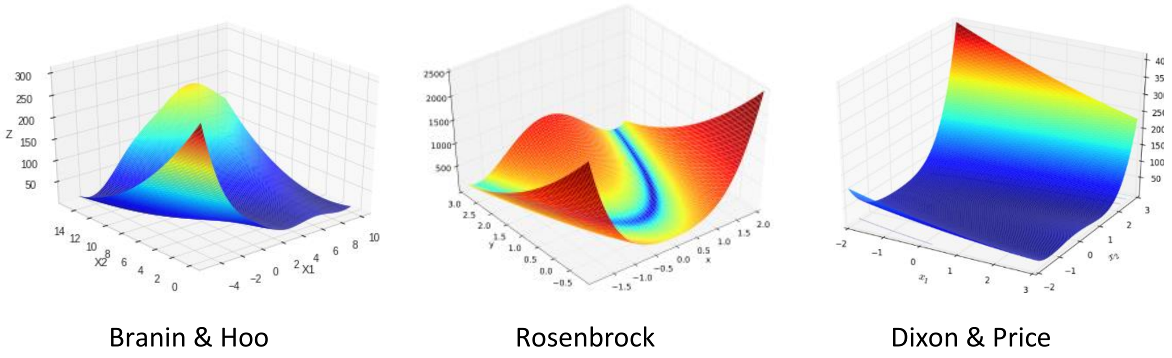
Table 2. Hyper-parameters in the ensembles throughout the case studies

Ensemble	Hyper-parameter	Values
GBwDM	Learning rate	0.05, 0.10
	Number of estimators	50, 75, 100
	Subsampling	0.3, 0.5, 1.0
	Feature sampling	$\sqrt{\#features}, \#features$
	Tree's max depth	2, 3, 4
HEA	Base learners	Linear regressor
		SVR RBF
		ML perceptron
		Gaussian process

The results of the proposed approach and those of heterogeneous ensembles are compared considering median of the mean square error in the test data set; the number of times each particular type of ensemble exhibited a lower mean square error (winner) is also reported.

7.1. Analytical test functions

The selection of these functions aimed to provide a variety of function behavior, number of features, and ranges; Branin & Hoo (2D), Rosenbrock (6D), and Dixon & Price (12D), all frequently used as modeling and optimization case studies (Table 3). Figure 7 illustrates the two-features behavior of the analytical test functions under consideration.



Branin & Hoo

Rosenbrock

Dixon & Price

Figure 7. Two-feature representations of the selected analytical test functions

Table 3. Formulas for the selected analytical test functions with number of features and range

Function	No. dimensions	Formula	Range
Branin & Hoo Dixon & Szegö 1978	*2D*	$\left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cos(x_1) + 10$ (7)	$5 \leq x_1 \leq 10$ $0 \leq x_2 \leq 15$
Rosenbrock 6D Rosenbrock 1960	*6D*	$\sum_{i=1}^{N/2}[100(x_{2i-1}^2 - x_{2i})^2 + (x_{2i} - 1)^2]$ (8)	$-2 \leq x_i \leq 2$
Dixon & Price 12D Dixon & Price 1989	*12D*	$(x_1 - 1)^2 + \sum_{i=2}^{12}(2x_i^2 - x_{i-1})^2$ (9)	$-10 \leq x_i \leq 10$ $i = 1, 2, \ldots, 12$

Training and test data sets

For all the analytical case studies data is generated using a latin-hypercube design for both, training and test sets, with sizes 10x and 100x the corresponding input domain dimension, respectively. For example, for the Rosenbrock 6D case study, the training and test sets are of size 60 and 600, respectively. On a separate note, Table 4 shows the hyper-parameters grid under consideration for the heterogeneous ensemble.

7.1. Industrial test cases

These cases differ in the nature of the simulation-based models, i.e., multi-phase flow in porous media, and structural analysis and the number and type of features under consideration, that is, 6D discrete and 9D continuous input variables, respectively.

Table 4. Base learners hyper-parameters grid – Analytical test functions

Model	Hyper-parameter	Values		
		Branin & Hoo	***Rosenbrock***	***Dixon & Price***
Linear regression	-	-	-	-
SVR – Radial basis function	C*	55, 110, 220	900, 1800, 3600	382, 764, 1528 (*x* 10^{3})
	Gamma†	0.01, 0.10, 0.20	0.100, 0.125, 0.150	0.1, 0.2, 0.5 (*x* 10^{-3})
Multi-layer perceptron	Hidden layer size**	3, 5, 6	4, 6, 8	4, 6, 9
	Activation function	relu, logistic	relu, logistic	relu, logistic
Gaussian process	-	-	-	-

* The C values under consideration are 0.5, 1.0 and 2.0 times the reference value calculated as proposed by Cherkassky and Ma (2004)

† The Gamma values were the result of a preliminary study

** Settled by limiting the number of parameters of the model (weights) to be a fraction (50%, 75%, 100%) of the total number of data points available during the learning process

7.1.1. Net present value of the oil production in a mature reservoir considering alternative new well locations

Most of the current oil production comes from mature reservoirs, that is, reservoirs whose production have reached their peaks and have started to decline. At the same time, the rate of replacement of the produced reserves by new discoveries of conventional resources have been steadily declining in the last decade, with a trend that will continue for the foreseeable future (Syed 2012, Brown et al. 2017). Hence, the modeling and optimization of oil recovery processes from aging resources are rapidly becoming a pressing issue for both energy policy and oil producers.

The problem of interest is to estimate the net present value (NPV) of the oil production in a mature reservoir (in western Venezuela) during a twenty-year horizon considering the variables LOC_1i, LOC_1j, LOC_2i, LOC_2j, LOC_3i, LOC_3j, that is, the potential areal location of three (3) new wells. The NPV (target variable) formulae and the discrete range of potential well locations are available in Table 5; note the NPV is a function of the reservoir production (oil+gas), water injection and operating costs under a specific oil price profile (Table 6). The case study corresponds to a real reservoir whose main characteristics are summarized in Table 7.

Table 5. NPV formulae (target function) with economic parameters and the discrete range of potential well locations

Net Present Value [Bs.]	$$\sum_{k=1}^{K} \frac{\left(q_o(k)P_o + q_g(k)P_g - q_w(k)C_w - q_i(k)C_i - OC_k\right)\Delta t_k}{(1+\beta)^{\frac{k\Delta t_k}{365}}} \qquad (10)$$			
	where		*Value*	*Unit*
	$q_{\{o,g,w\}}$	daily production of oil, gas and water	-	[bbl]
	q_i	water injection rate	-	[bbl]
	P_o	net income due to oil production	*See Table 6*	[Bs/bbl]
	P_g	net income due to gas production	4 030.65	[Bs/MMBTU]
	C_w	cost of handling the water production	0	[Bs/bbl]
	C_i	water injection cost	40.10	[Bs/bbl]
	OC_k	Operating costs(e.g., drilling) at the *k*th instant	-	[Bs]
	k	sampling instant	-	[days]
	Δt_k	is the time step size in days	-	[days]
	β	annual discount factor	10	[%]
	K	production horizon (20 years) in days	20*365	[days]
	Bbl: barrels; Bs.: Bolivars, the Venezuelan currency			

Range of potential well locations

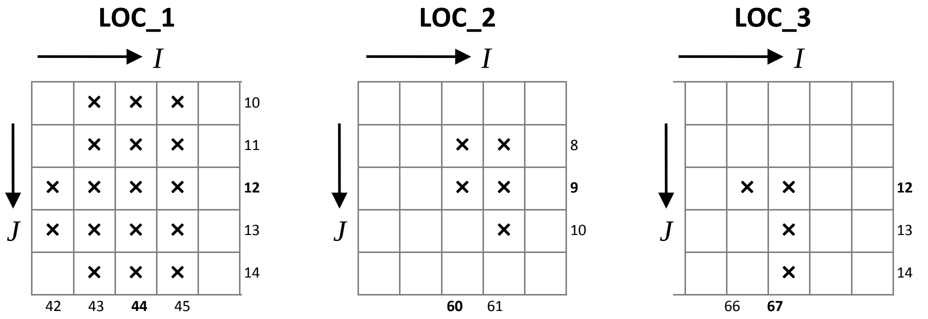


Table 6. Oil price profile used in the Net Present Value calculation

Year	**2016**	**2017**	**2018**	**2019**	**2020**	**2021**	**2022**	**2023**	**2024**	**2025**	**2026**	**2027**
Price	41,63	51,56	56,39	60,19	64,00	65,74	68,54	69,49	71,49	73,49	74,49	75,49

Year	**2028**	**2029**	**2030**	**2031**	**2032**	**2033**	**2034**	**2035**	**2036**	**2037**	**2038**
Price	76,49	77,49	78,49	79,49	80,49	81,49	82,49	83,49	84,49	85,49	86,49

The reservoil numerical model has a grid of 135x75x78 for a total of 789,785 cells, with an average cell size of 100m x 100m x 3m (Figure 8). This model corresponds to a mature reservoir with more than 80 wells and 20-years exploitation plan that includes among other elements three (3) new wells to be drilled at specific times. The wells are assumed to be vertical and the NPV of interest is associated with alternative aereal locations for the new wells.

Table 7. General mature reservoir characteristics – NPV of mature oil reservoir case study

Property	Value
Initial pressure, psia	5 550
Buble pressure (Pb), psia	4 215
Average porosity, fraction	0.18
Average permeability, mD	294.9
Average rock shale (Vsh), fraction	0.25
Initial water saturation (Swi), fracción	0.28
API gravity	39.5
Oil volumetric factor (Boi), bbl/STB	1.7668
Initial solution gas-oil ratio (Rsi), SCF/STB	1 436
Oil viscosity @ Pb, cps	0.251

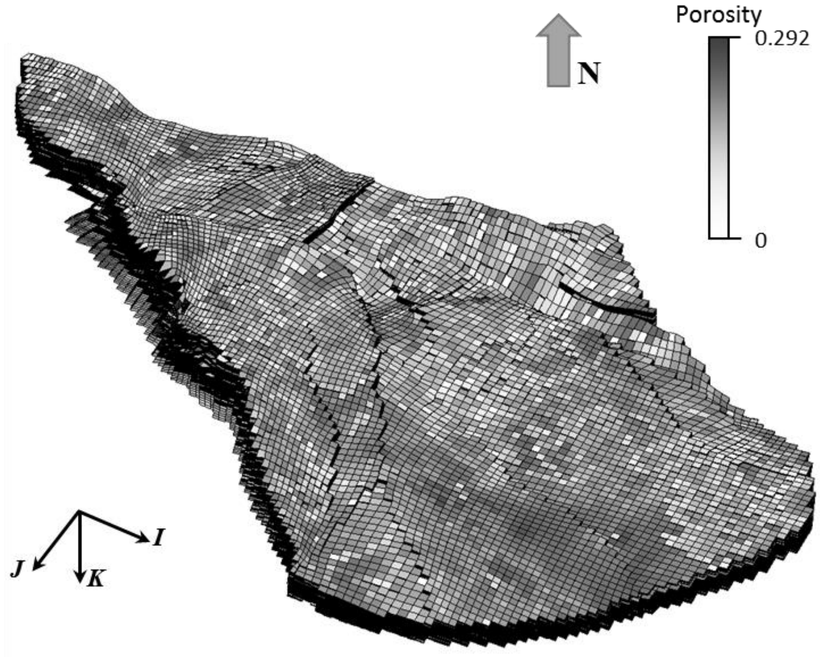


Figure 8. Numerical grid of mature reservoir simulation model - Oil production NPV case study

Training and test data sets

The data consists of 178 samples randomly divided in training and test data sets of size 60 and 118, respectively. Table 8 shows the hyper-parameters grid for the heterogeneous ensemble.

Table 8. Hyper-parameters grid for the heterogeneous ensemble – Oil production NPV case study.

Model	Hyper-parameter	Values
Linear regression	-	-
SVR – Radial basis function	C*	135x10^6, 270x10^6, 540x10^6
	Gamma†	0.10, 0.15, 0.20
Multi-layer perceptron	Hidden layer size**	4, 6, 8
	Activation function	relu, logistic
Gaussian process	-	-

* The C values under consideration are 0.5, 1.0 and 2.0 times the reference value calculated as proposed by Cherkassky and Ma (2004)

† The Gamma values were the result of a preliminary study

** Settled by limiting the number of parameters of the model (weights) to be a fraction (50%, 75%, 100%) of the total number of data points available to the learning process

7.2. FSAE brake pedal

The Formula SAE project (SAE international 2016b, c) with the annual participation of 2470 students from 120 university teams, promotes careers and excellence in engineering in the automotive industry including research, design, manufacturing, testing, developing, marketing, management and finances. In particular, each component in a single seater FSAE vehicle must be safe and lightweight in order to decrease the vehicle mass, and take the most advantage of the engine power. The brake pedal design of FSAE vehicles represents an easy to understand structural problem among engineering students and practitioners around the world yet relevant to aerospace and automotive industrial environments.

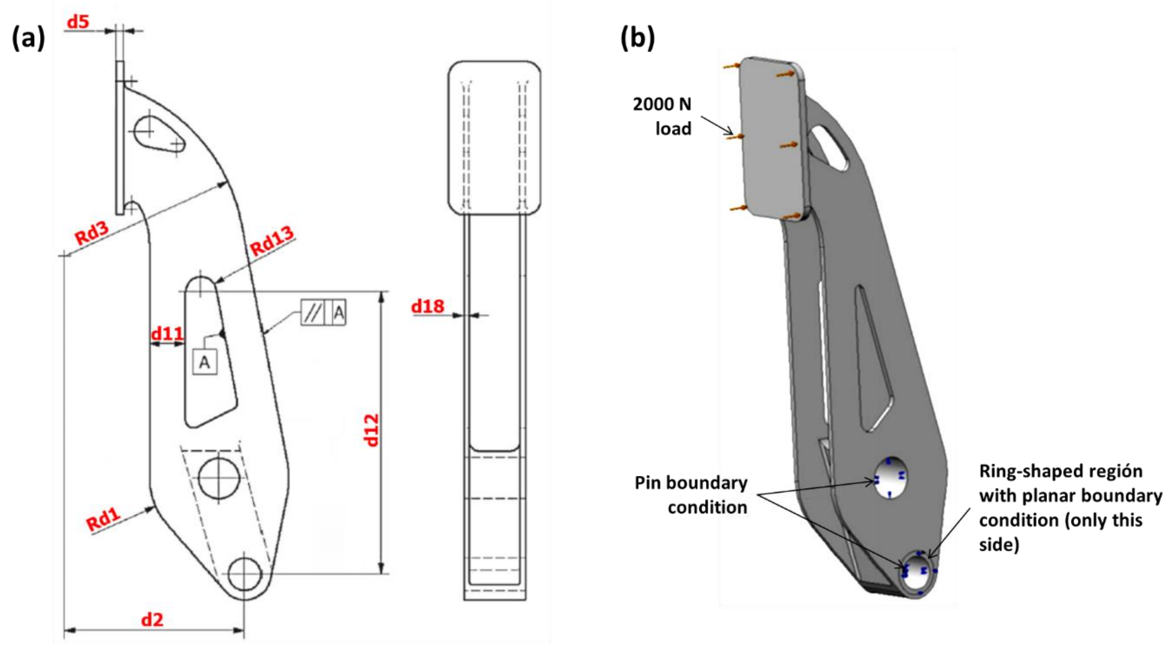


Figure 9. Geometric variables (a), loads and boundary conditions (b) – FSAE brake pedal case study

With reference to Figure 9 (a), the problem of interest is to estimate the von Mises stress (ε_S) and buckling load factor (ε_B) in a FSAE brake pedal considering as features eight (8) geometric variables (in circles), and the Young modulus –E of the material. The loads and boundary conditions of the brake pedal are also shown in Figure 9 (b) with the range of the features specified in Table 9. Please see Romero and Queipo (2017) for a detailed discussion.

Training and test data sets

The data consists of 1320 samples randomly divided in training and test data sets of size 90 and 1230, respectively.

Table 10 shows the hyper-parameters grid for the heterogeneous ensemble.

Table 9. Feature probability distributions - FSAE brake pedal case study

Feature	Distribution	Parameters
d1	*Uniform*	[27, 33]
d2		[70, 80]
d3		[70, 87]
d5		[2, 4]
d11		[8, 18]
d12		[80, 115]
d13		[2, 8]
d18		[2, 6]
E	*LogNormal*	μ: 11.059, σ: 0.0709

Table 10. Hyper-parameters grid for the heterogeneous ensemble – FSAE brake pedal case study

Base learner	Hyper-parameter	Values	
		von Mises stress	***Buckling load factor***
Linear regression	-	-	-
SVR – Radial basis function	C*	60, 120, 240	10, 20, 40
	Gamma†	0.10, 0.15, 0.20	0.10, 0.15, 0.20
Multi-layer perceptron	Hidden layer size**	4, 6, 9	4, 6, 9
	Activation function	Relu, logistic	Relu, logistic
Gaussian process	-	-	-

* The C values under consideration are 0.5, 1.0 and 2.0 times the reference value calculated as proposed by Cherkassky and Ma (2004)

† The Gamma values were the result of a preliminary study

** Settled by limiting the number of parameters of the model (weights) to be a fraction (50%, 75%, 100%) of the total number of data points available to the learning process

8. Results and discussion

The proposed approach (GBwDM) exhibited:

- Lowest median (statistically significant) of test set mean square errors and most winners for analytical (except the 2D) and industrial case studies, namely, oil production NPV optimization, FSAE – von Mises stress and FSAE – buckling load factor
- Strong resistance to overfitting for all case studies
- Considerable speedup with respect to heterogeneous ensemble learning
- Diversity measures that proved to be effective for improving the prediction performance
- A preliminary assessment of feature importance that is statistically aligned with those reported by the frequently used Sobol's method

Lowest median of test set mean square errors for all analytical (except the 2D) case studies. Based on one hundred and fifty (150) random trials, GBwDM outperformed the heterogeneous ensemble approach (HEA) with 22.6% (6D), and 5.7% (12D) lower median MSE (Figure 10; Table 11); the error distribution associated with the training data is shown in Figure 11. A Pearson's chi-squared test on the difference of the medians confirmed the statistical significance of the results. When considering the most meaningful analytical test cases (6D and 12D) and each of the trials, GBwDM was the winner (exhibited the lowest mean square error) in proportions to 140:9 (14:1) and 86:64 (3:2), respectively.

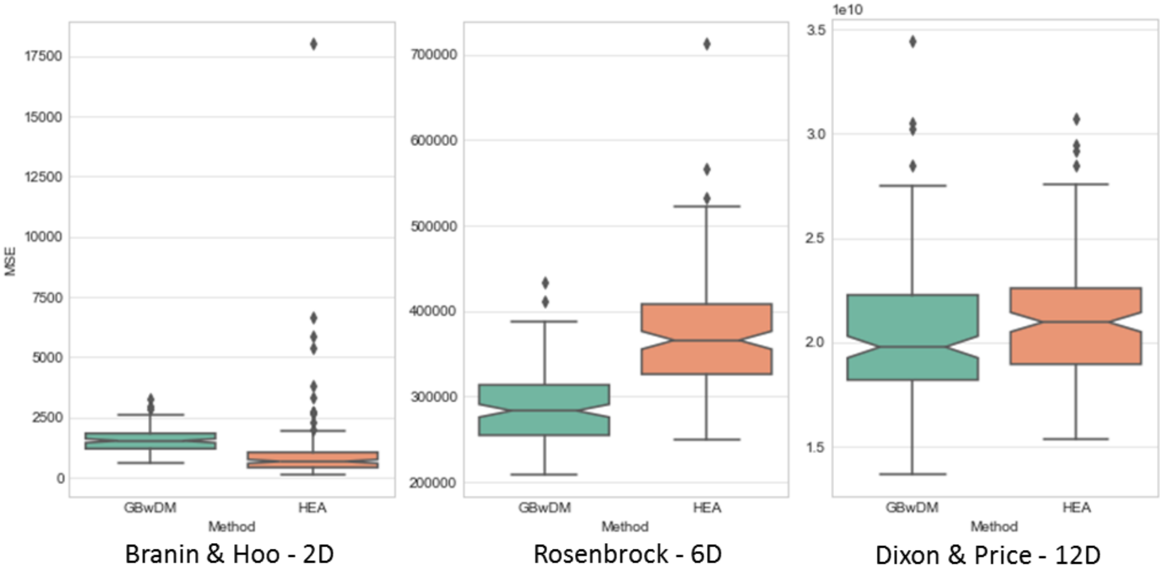
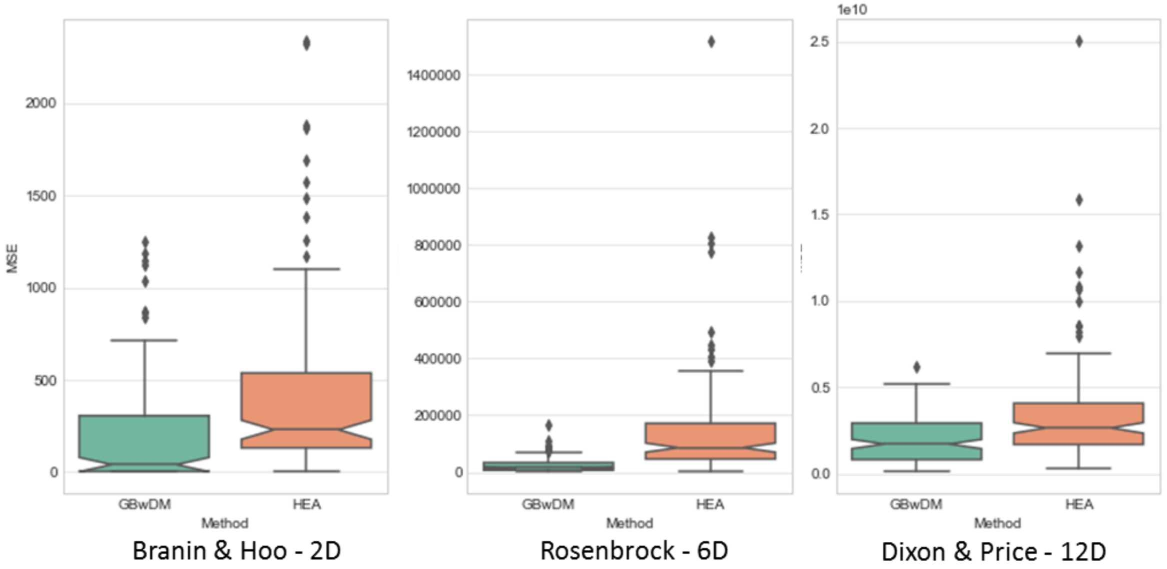


Figure 10. Test set MSE distributions for the analytical case studies, i.e., Branin & Hoo – 2D, Rosenbrock – 6D, and Dixon & Price – 12D

Table 11. Median of test set MSE for the analytical case studies, i.e., Branin & Hoo – 2D, Rosenbrock – 6D, and Dixon & Price – 12D

MSE (Test set)	**Branin & Hoo 2D**	**Rosenbrock 6D**	**Dixon & Price 12D**
GBwDM	1538.59	283234.06	19.77×10^9
HEA	681.21	366008.06	20.96×10^9
Pearson's chi-squared test (p-value)	<< 0.05	<< 0.05	<< 0.05

As expected, the proposed approach outperformed the average error of the individual surrogates. For example, in the highest dimensional analytical case study – Dixon 12D –, in the training set the median of the MSE of the proposed approach is 13x lower than the average of the medians of the individual surrogates (consistent with the ambiguity principle, see Brown et al. 2005b) trained using 5-fold cross-validation. It is worth noting that the individual surrogate with the best performance (GPR) can be shown to overfit the data (highest test error). The second best individual surrogate (RBF) does in fact perform only slightly worse than the proposed approach, but no criterion is available to select which individual surrogate will generalize the best based on training performance. Finally, MLP and LR exhibited similar performance, significantly worse than GPR and RBF. On the other hand, when considering the test set the median of the MSE of the proposed approach is 6x lower than the average of the medians of the individual surrogates. Among the individual surrogates, the best performance was provided by RBF, followed by MLP and LR (approx. 3x higher than RBF) with GPR exhibiting the worst performance.



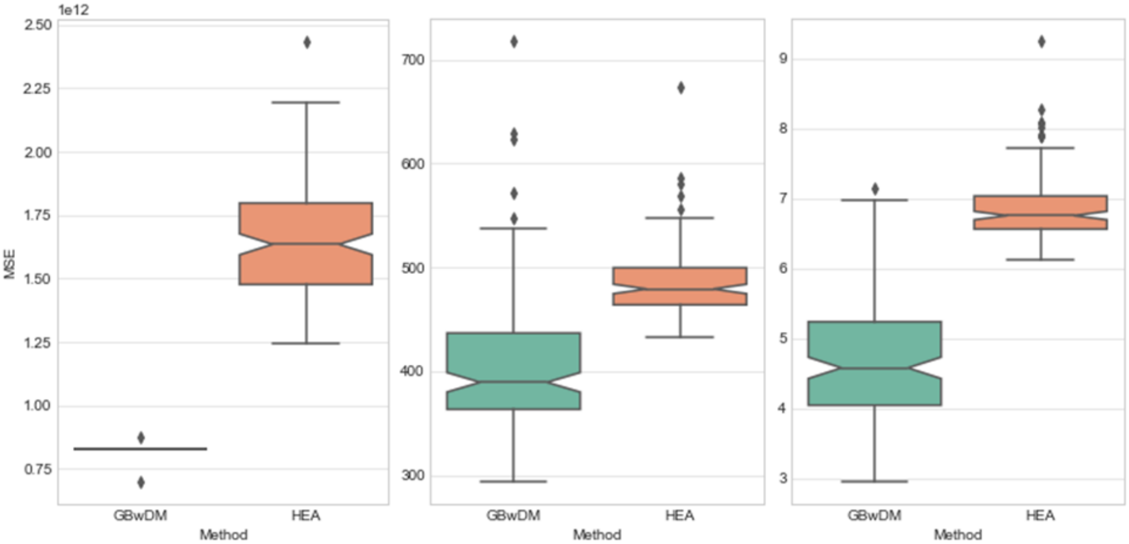
Branin & Hoo - 2D

Rosenbrock - 6D

Dixon & Price - 12D

Figure 11. Training MSE distributions for the analytical case studies, i.e., Branin & Hoo – 2D, Rosenbrock – 6D, and Dixon & Price – 12D

GBwDM also exhibited the lowest median MSE in all industrial case studies, namely, oil production NPV, FSAE – von Mises stress and FSAE – buckling load factor. Its application resulted in median MSE, 49.5%, 18.8% and 32.3% lower than those associated with the HEA (Figure 12, Table 12). Figure 13 shows the mean square errors obtained during the training phase for the industrial test cases. As with the analytical case studies, a Pearson's chi-squared test on the difference of the medians confirmed the statistical significance of the cited test results. Furthermore, GBwDM was the winner (exhibited the lowest mean square error) in proportions to 150:0 (15:1), 134:16 (8:1), and 149:1 (15:1), respectively.

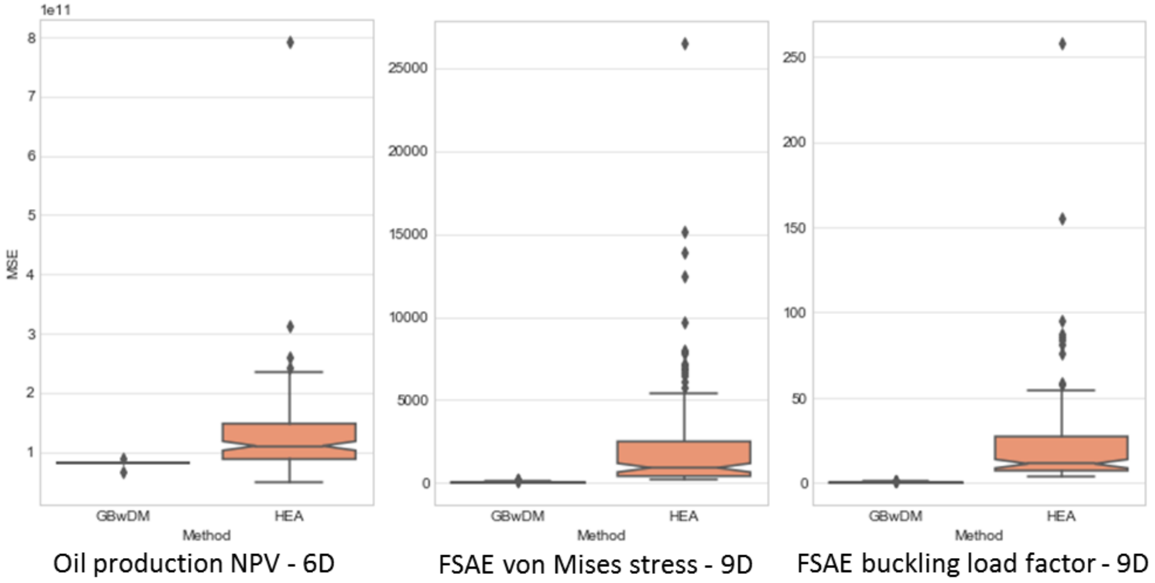


Oil production NPV - 6D   FSAE von Mises stress - 9D   FSAE buckling load factor - 9D

Figure 12. Test set MSE distribution for the industrial case studies, i.e., oil production NPV, FSAE – von Mises stress and FSAE – buckling load factor

Table 12. Median of test set MSE for the industrial case studies, i.e., oil production NPV, FSAE – von Mises stress and FSAE – buckling load factor

MSE (Test set)	**Oil production NPV (5D)**	**FSAE – von Mises stress (9D)**	**FSAE – buckling load factor (9D)**
GBwDM	0.827×10^{11}	389.34	4.58
HEA	1.638×10^{11}	479.25	6.75
Pearson's chi-squared test (p-value)	<< 0.05	<< 0.05	<< 0.05



Oil production NPV - 6D

FSAE von Mises stress - 9D

FSAE buckling load factor - 9D

Figure 13. Training MSE distribution for the industrial case studies, i.e., oil production NPV, FSAE – von Mises stress and FSAE – buckling load factor

As expected, in the HEA no single surrogate prevailed throughout all the case studies with support vector regression (SVR) and linear regression (LR) being the best performers and Gaussian Process (GP) and Multilayer Perceptron (MLP) exhibiting significantly less contributions to the ensemble outputs. Specifically, SVR was the best performer in all the analytical and the oil production NPV case studies, while LR outperformed the rest of the base learners in the FSAE – von Mises stress and FSAE – buckling load factor industrial case studies (Figure 14). Note that the weights (coefficients) in the ensemble can be negative and correspond to the implementation of heterogeneous ensembles in one of the most recognized machine learning libraries (scikit-learn.org).

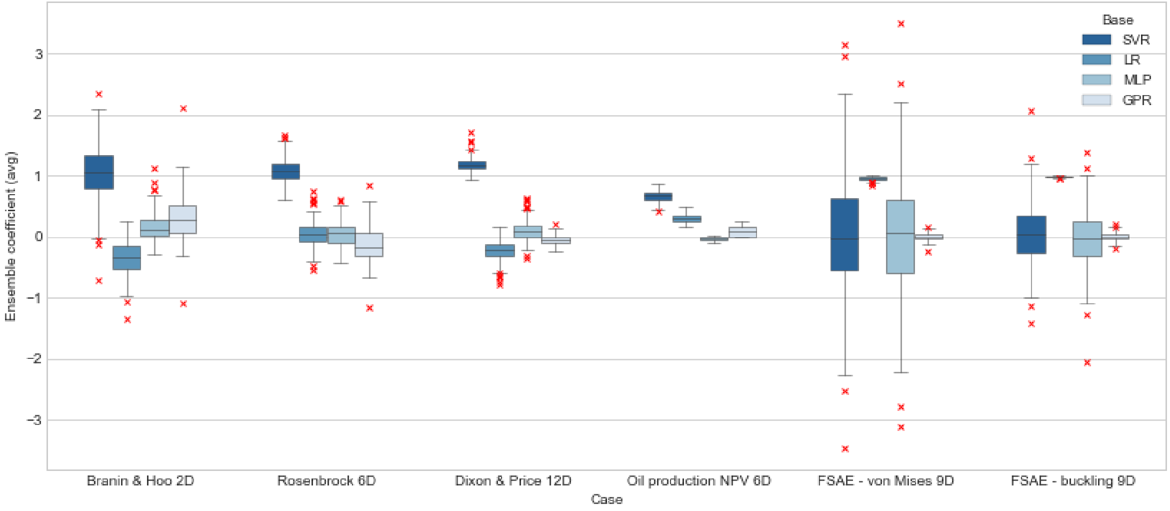


Figure 14. Base learner coeficients (weights) distribution for the analytical and industrial case studies

GBwDM resistance to overfitting. Increasing the numbers of estimators in GBwDM invariably reduced the training error and did not lead to a significant increase in the test set error; this adds to the empirical evidence that gradient boosting can be quite robust to overfitting. This is an attractive feature considering that you can grossly specify the number of members of the ensemble and still deliver a strong prediction performance. Figure 15 and Figure 16 depict the monotonically decreasing behavior of the training and test set error versus the number of estimators in the ensemble for all the analytical and industrial case studies, respectively.

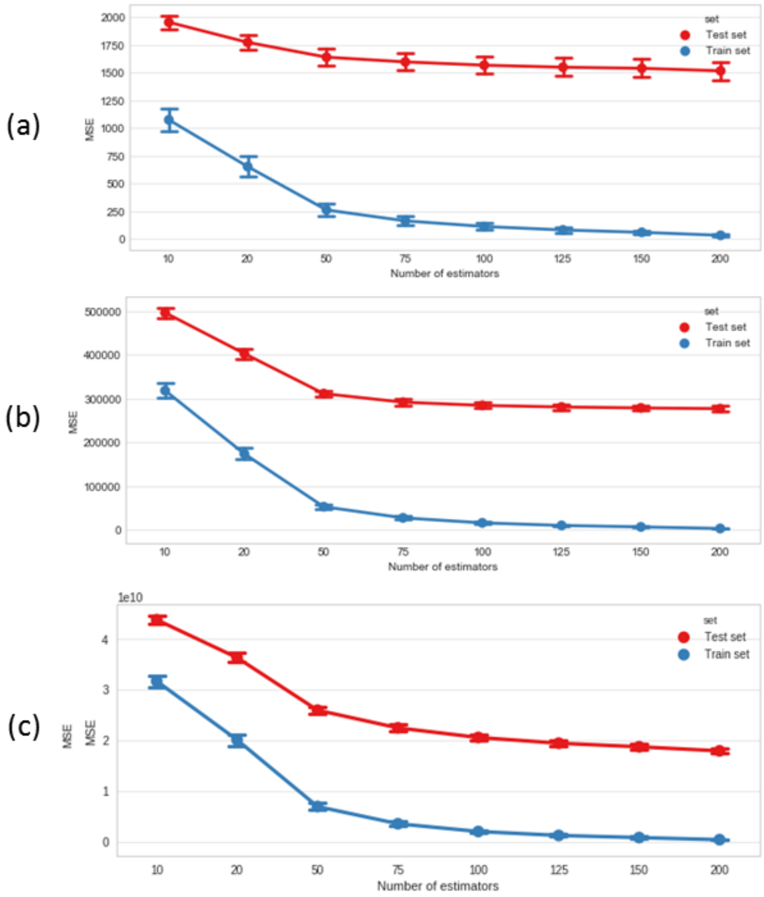


Figure 15. Median of training and test set MSE for the analytical case studies, i.e., Branin & Hoo – 2D (a), Rosenbrock – 6D (b), and Dixon & Price – 12D (c)

Effectivity of diversity measures. GBwDM incorporates two promoting diversity measures in the ensemble, namely, random subspace and subsampling. Results show that in the 6D and 12D analytical case studies, a significant amount of subsampling (either 0.3 or 0.5) was adopted by most of the winners in each of the trials, approximately 91% and 97%, respectively. Similarly, random subspace (square root of the number of features) was present in a considerable amount of the winners, i.e., 33% and 47%, respectively. On the other hand, in the industrial test cases, the most effective diversity measure was subsampling. Subsampling values of either 0.3 or 0.5 were present in 100%, 92% and 96% of the predictor winners in the Oil production NPV, FSAE – von Mises stress, and FSAE buckling load factor, respectively.

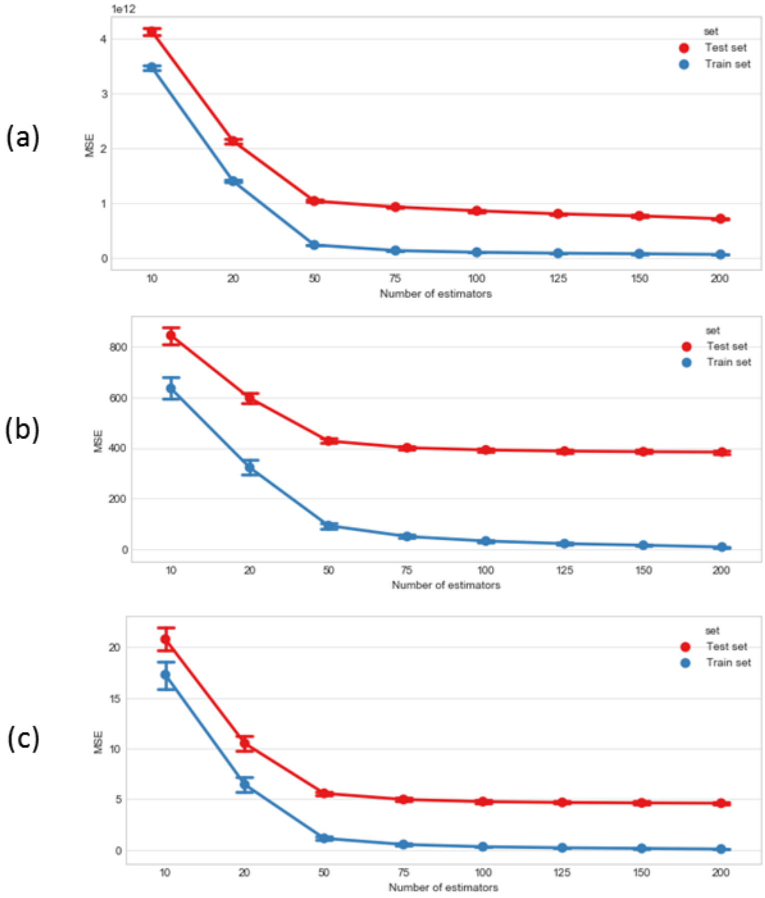


Figure 16. Median of training and test set MSE for the industrial case studies, i.e., oil production NPV – 6D (a), FSAE – von Mises stress– 9D (b), and FSAE buckling load factor – 9D (c)

Computational efficiency. The median computational execution times of the proposed approach, in general, was significantly lower than those of the heterogeneous counterpart. For example, when considering the highest dimensional analytical test case (Dixon 12D), the computational time of the proposed approach was approx. 18 seconds versus 458 seconds of the heterogeneous ensemble approach, that is, 25x faster when using the cloud computing platform Colaboratory by Google. The main reason for the speedup can be attributed to the fact that being resistant to overfitting is a natural feature of GBwDM (stagewise parameter setting, see Hastie et al. 2009), while in the HEA the quest for such a feature relies on a costly nested cross-validation procedure (Cawley and Talbot 2010) as implemented in MLxtend (https://rasbt.github.io/mlxtend/).

Preliminary feature selection as a by product of the modeling process. The potential of GBwDM for this task is illustrated with the FSAE Brake pedal case studies. Figure 17 (a) and Figure 18 (a) display the ranking of the feature variables for a random trial of the training set in the FSAE-von Mises stress and FSAE-bucking load factor studies, respectively. Note that for the former target variable the three (3) most important features (with reference to Figure 17 (a)) are d18 (23.2%), d1

(15.1%) and d12 (12.8%), while for the latter (Figure 18 (a)), those are d18 (49.8%), the module of elasticity E (8.2%), and d12 (7.8%); in parenthesis the reduction percentage of the total MSE. Considering the learning algorithm for decision trees greedily select the splitting variables for maximum reduction of a variability measure (e.g., MSE), one would assume that the feature ranking should be related to that of the Sobol's method. That indeed seems to be the case. For one hundred and fifty (150) random trials of the training set, Figure 17(b) and Figure 18(b) show the empirical distribution of a measure of the rank correlation-similarity (Kendall's tau statistic) of the ordering of the most important features obtained from GBwDM and the Sobol´s method for both target variables of interest. The median of the corresponding p-values confirm the statistical dependence of the feature rankings from GBwDM and the Sobol´s method Figure 17(c), Figure 18(c). Specifically, for the target variables of interest, the medians of the p-values are in both cases approximately 0.02, significantly lower than the threshold of 0.05 for rejecting the null-hypothesis (the two ranking features are independent).

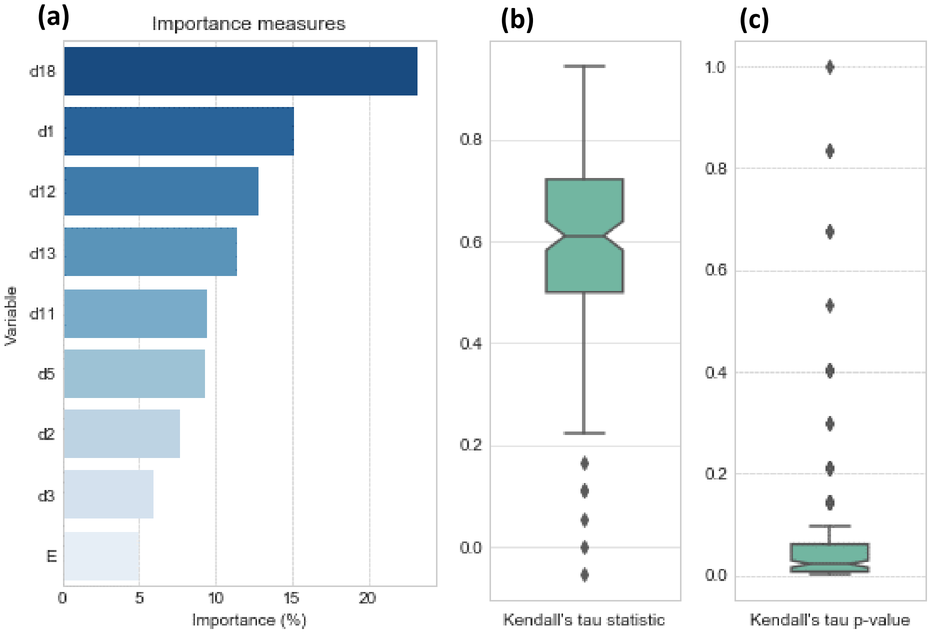


Figure 17. GBwDM feature importance for a selected random trial (a), a rank correlation-similarity (Kendall's tau) statistic distribution (b) and the corresponding p-value (c), for one hundred and fifty (150) random trials when compared with Sobol's feature importance ranking – FSAE von Mises stress case study

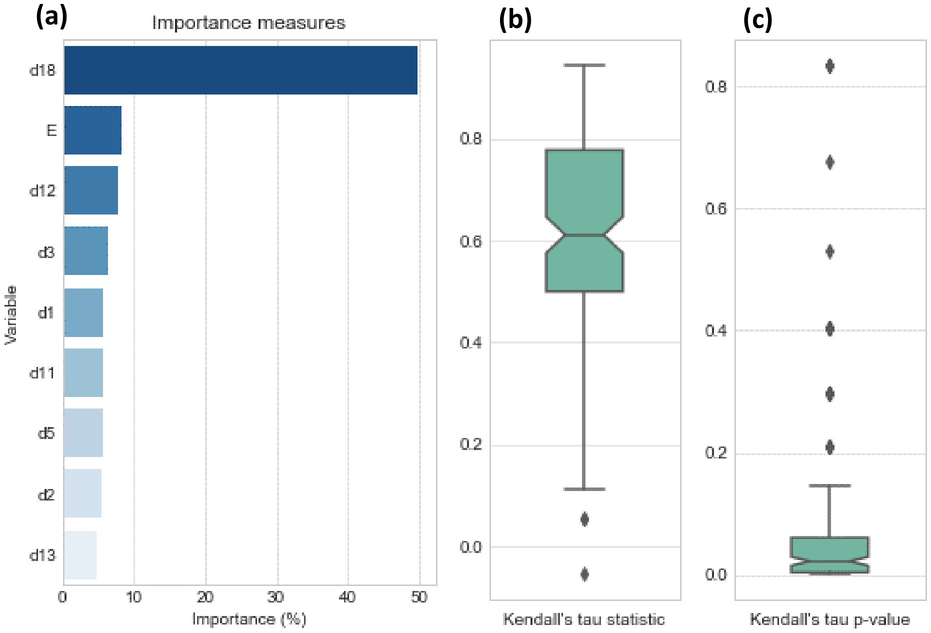


Figure 18. GBwDM feature importance for a selected random trial (a), a rank correlation-similarity (Kendall's tau) statistic distribution (b) and the corresponding p-value (c), for one hundred and fifty (150) random trials when compared with Sobol's feature importance ranking – FSAE buckling load factor case study

9. Conclusions and recommendations

This work presented an homogeneous ensemble approach based on gradient boosting and decision trees with diversity measures (subsampling and random subspace) for solving engineering modeling problems. The proposed approach is designed to improve prediction estimates by promoting diversity among the members of the ensemble (reducing the covariance component of the generalization error), be robust to overfitting and, as a byproduct of the modeling process, generate preliminary importance measures of the feature variables.

The effectiveness of the proposed approach was demonstrated in the modeling of a variety of analytical and industrial case studies with statistically significant gains over the classical heterogeneous ensemble approach and holds promise to be particularly useful in scenarios where surrogate selection and overfitting may be critical issues. Specifically, GBwDM exhibited the lowest median of test set mean square errors and most winners for all analytical (except the 2D) and industrial case studies. It also showed strong resistance to overfitting and diversity measures that proved to be effective for improving the prediction performance. All, while offering an assessment of feature importance that is statistically aligned with those reported by the frequently used Sobol's method and a significant speedup over heterogeneous ensembles.

The feature importance provided by the proposed approach is only a preliminary assessment and should not be interpreted as a substitute of the application of the Sobol's method, but as a byproduct of the ensemble learning process. While both are the result of a variance-based global sensitivity analysis, Sobol's method represents a more theoretical sound alternative. The latter can explicitly account for the contributions to variance not only of individual variables (like in the proposed approach), but also of their interactions. Note that if a detailed global sensitivity analysis is considered necessary, in the proposed approach the Sobol's method could be coupled with the final ensemble.

This work was designed with the perspective of more systematically controlling the diversity among the members of the ensemble. The strategy: randomly sampling the training data (random subspace and subsampling) and manipulating the ensemble generation process (gradient boosting). Considering the adopted perspective and strategy (independent of surrogates and cost function) and the promising results, the proposed approach and meaningful variants should be further evaluated. In particular, some potential research venues include its application to a broader set of industrial case studies, the use of alternative weak learners, cost functions, and diversity mechanisms (e.g., penalizing in the loss function the correlation among members of the ensemble).

10. Replication of results

Data and code (Jupyter notebooks) needed to replicate the results of selected case studies can be found at https://github.com/nqueipo/SMO-Manuscript-GBwDM. The data and code are released under Creative Commons CC BY-SA and GNU General Public (GPLv3) licences, respectively. Note for reviewers: If the manuscript is accepted for publication, a digital object identifier (DOI) will be assigned (using Zenodo.org or figshare.com) to the repository holding the final version of the data and code for readers' easier and global access.

11. Conflict of interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

12. References

Acar E, Rais-Rohani M (2009) Ensemble of metamodels with optimized weight factors. Struct. Multidiscipl. Optim 37(3):279–294

Audet C, Kokkolaras M, Le Digabel S, Talgorn B (2018) Order-based error for managing ensembles of surrogates in mesh adaptive direct search. Journal of Global Optimization 70(3):645-675. https://doi.org/10.1007/s10898-017-0574-1

Bishop C (1995) Neural networks for pattern recognition. Oxford University Press, Oxford

Breiman L, Friedman JH, Olshen RA, Stone CJ (1984). Classification and Regression Trees. The Wadsworth and Brooks-Cole statistics-probability series, Monterey

Breiman L (1996) Bagging Predictors. Machine Learning 24(2):123-140. https://doi.org/10.1023/A:1018054314350

Breiman L (1999) Prediction games and arcing algorithms. Neural Computation 11(7):1493–1517. https://doi.org/10.1162/089976699300016106

Brown G, Wyatt JL, Tiño P (2005a) Managing diversity in regression examples. Journal of Machine Learning Research 6:1621-1650

Brown G, Wyatt JL, Harris R, Yao X (2005b) Diversity creation methods: a survey and categorization. Information Fusion 6:5-20

Brown JB, Salehi A, Benhallam W, Matringe S (2017) Using data-driven technologies to accelerate the field development planning process for mature-field rejuvenation. SPE Western Regional Meeting, Bakersfield, California, USA, 23–27 April. SPE 185751.

Cawley, GC, Talbot NLC (2010). On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. Journal of Machine Learning Research 11:2079-2107.

Chen T, Guestrin C (2016) XGBoost: A Scalable Tree Boosting System. KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 785-794. https://doi.org/10.1145/2939672.2939785

Chen D, Zhong A, Gano J, Hamid S, De Jesus O, and Stephenson S (2007) Construction of surrogate model ensembles with sparse data. IEEE Congress on Evolutionary Computation, Singapore 244-251. doi: 10.1109/CEC.2007.4424478

Chen L, Qiu H, Jiang C, Cai X, Gao L (2018) Ensemble of surrogates with hybrid method using global and local measures for engineering design. Struct Multidisc Optim 57(4):1711-1729. https://doi.org/10.1007/s00158-017-1841-y

Cherkassky V, Ma Y (2004) Practical selection of SVM parameters and noise estimation for SVM regression. Neural Networks 17:113-26. https://doi.org/10.1016/S0893-6080(03)00169-2

Dixon LCW, Szego GP (1978). The global optimization problem: an introduction. Towards global optimization 2:1-15

Dixon LCW, Price RC (1989) The Truncated Newton Method for Sparse Unconstrained Optimisation Using Automatic Differentiation. Journal of Optimization Theory and Applications 60(2):261-275. https://doi.org/10.1007/BF00940007

Draper N, Smith H (1969) Applied regression analysis. Biom J 11(6): 427–427. https://doi.org/10.1002/bimj.19690110613

Fang J, Gao Y, Sun G. (2014) Fatigue optimization with combined ensembles of surrogate modeling for a truck cab. J Mech Sci Technol 28: 4641–4649

Fayyad UM, Irani KB (1992) On the handling of continuous-valued attributes in decision tree generation. Mach Learn (1992) 8:87-102. https://doi.org/10.1007/BF00994007

Friedman JH (2001) Greedy function approximation: a gradient boosting machine. Ann Statist 29(5):1189-1232

Freund Y (1995) Boosting a weak learning algorithm by majority. Information and Computation 121(2):256–285. https://doi.org/10.1006/inco.1995.1136

Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. Struct Multidisc Optim 33(3):199–216

Hamza K, Saitou K (2012) A Co-Evolutionary Approach for Design Optimization via Ensembles of Surrogates with Application to Vehicle Crashworthiness. Transactions of ASME Journal of Mechanical Design 134(1):011001-1–011001-10

Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Springer Series in Statistics, New York

Hecht-Nielsen R (1989), Neurocomputing. Addison Wesley, Boston, USA

Ho TK (1998) The random subspace method for constructing decision forests. IEEE PAMI 20(8):832–844. https://doi.org/10.1109/34.709601

Karagiannopoulos M, Anyfantis D, Kotsiantis S, Pintelas P (2007). Feature selection for regression problems. Proceedings of the 8th Hellenic European Research on Computer Mathematics and its Applications (HERCMA), Athens, Greece

Kazemitabar J, Amini A, Bloniarz A, Talwalkar AS (2017) Variable importance using decision trees. In Advances in Neural Information Processing Systems 30:426-435

Liu Y, Yao X (1999) Ensemble learning via negative correlation. Neural Networks 12(10):1399-1404. https://doi.org/10.1016/S0893-6080(99)00073-8

Mendes-Moreira J, Soares C, Jorge AM, De Sousa JF (2012) Ensemble approaches for regression: A survey. ACM Computing Surveys 45(1):1-40. https://doi.org/10.1145/2379776.2379786

Poggio T, Smale S (2003) The mathematics of learning: dealing with data. Not Am Math Soc 50(5):537–544

Reeve HWJ, Brown G (2018) Diversity and Degrees of Freedom in Regression Ensembles. Neurocomputing 298(12):55-68

Romero J, Queipo N (2017) Reliability-based and deterministic design optimization of a FSAE brake pedal: a risk allocation analysis. Struct Multidisc Optim 56(3): 681-695. https://doi.org/10.1007/s00158-017-1747-8

Rooney N, Patterson D, Anand S, Tsymbal A (2004) Dynamic integration of regression models. In: Roli F, Kittler J, Windeatt T (eds). Multiple Classifier Systems. MCS 2004. Lecture Notes in Computer Science 3077:164-173. https://doi.org/10.1007/978-3-540-25966-4_16

Rosen BE (1996) Ensemble learning using decorrelated neural networks. Connection Science 8(3-4):373-384. https://doi.org/10.1080/095400996116820

Rosenbrock HH (1960) An automatic method for finding the greatest or least value of a function. The Computer Journal 3(3):175–184. http://dx.doi.org/10.1093/comjnl/3.3.175

Sacks J, Welch W, Mitchell T, Wynn H (1989) Design and analysis of computer experiments. Statistical Science 4(4):409–435.

SAE International (2016a) SAE Collegiate Design Series http://students.sae.org/cds/

Sanchez E, Pintos S, Queipo NV (2008) Toward an optimal ensemble of kernel-based approximations with engineering applications. Struct Multi Optim 36(3):247–261

Schapire RE (1990) The strength of weak learnability. Machine Learning 5(2):197–227. https://doi.org/10.1007/BF00116037

Schölkopf B, Smola AJ (2002) Learning with kernels. MIT Press, Cambridge, MA

Sobol I (1993) Sensitivity analysis for non-linear mathematical models. Mathematical Modeling & Computational Experiment 1:407–414

Syed A (2012) Technology Focus: Mature Fields and Well Revitalization. Journal of Petroleum Technology 64:74-74. https://doi.org/10.2118/0112-0074-JPT

Tenne Y (2013) An optimization algorithm employing multiple metamodels and optimizers. International Journal of Automation and Computing 10(3):227-241. https://doi.org/10.1007/s11633-013-0716-y

Ueda N and Nakano R (1996). Generalization error of ensemble estimators. In Proceedings of International Conference on Neural Networks(ICNN'96), 90-95. http://doi.org/10.1109/ICNN.1996.548872

Viana FAC, Simpson TW, Balabanov V, Toropov V (2014) Metamodeling in multidisciplinary design optimization: how far have we really come?. AIAA Journal 52 (4):670-690. https://doi.org/10.2514/1.J052375

Yin H, Fang H, Wen G, Gutowski M,Xiao Y (2018) On the ensemble of metamodels with multiple regional optimized weight factors. Struct Multidisc Optim 58:245-263. https://doi.org/10.1007/s00158-017-1891-1

Zerpa L, Queipo NV, Pintos S, Salager JL (2005) An optimization methodology of alkaline–surfactant–polymer flooding processes using field scale numerical simulation and multiple surrogates. J Pet Sci Eng 47(3-4):197–208. https://doi.org/10.1016/j.petrol.2005.03.002

Zhang J, Chowdhury S, Messac A (2012) An adaptive hybrid surrogate model. Struct Multi Optim 46(2):223–238. https://doi.org/10.1007/s00158-012-0764-x

13. Appendix: Metamodels

13.1. Linear regression

In this work the regression models (Draper and Smith 1969) considered are of linear form:

$$\hat{y}(\boldsymbol{x}) = \beta_0 + \sum_{i=1}^{n_{rv}} \beta_i x_i$$

where n_{rv} is the total number of features, x_i, and the parameters β_0 and β_i are estimated by the least-square method. The implementation used in this study corresponds to *LinearRegressor* in the Scikit-learn library.

13.2. Artificial neural networks

The neural network model used corresponds to the so-called multi-layer perceptron whose output can be expressed as:

$$O_P(I_P) = G[W \times (G[V \times I_P])]$$

where I_P denotes the input values, V and W the weights matrices between the input and hidden layers, and the hidden and output layers, respectively. The symbol G represents the application of the logistic function or the rectified linear unit (ReLU) to each of the elements of its arguments.

It can be shown that the three-layer perceptron is, for function approximation purposes, as capable as another neural network with higher number of layers, and can estimate a function with arbitrary precision provided the number of neurons in the hidden layers and the weights are properly set (Hecht-Nielsen 1989). As a result, our attention will be restricted to three-layer perceptron.

The number of neurons in the input and output layers are established by the number of input and output variables of the function to be estimated. The number of neurons in the hidden layer can be settled through different criteria; for example, limiting the number of parameters of the model (weights) to be a fraction of the total number of data points available to the learning process. The implementation used in this study corresponds to *MLPRegressor* in the Scikit-learn library.

13.3. Support vector regression

The kernel-based regression models $M_i s$ can be seen as solutions of the following variational problem:

$$\min_{M \in H} Z(M) = \frac{1}{n} \sum_{i=1}^{n} L\big(y_i - M(x_i)\big) + \lambda \|M\|_H^2$$

over some large space of functions H where L and λ denote a particular loss function (e.g., quadratic, Laplace, ϵ-insensitive, and Huber loss functions) and a regularization parameter, respectively. The operator $\|\cdot\|_H^2$ is the Hilbert-space norm which penalizes models that are too complex. If we restrict ourselves to reproducing kernel Hilbert spaces (RKHS) the variational problem can be formulated as stated as:

$$\min_{M \in H} Z(M) = \frac{1}{n}\sum_{i=1}^{n} L\big(y_i - \langle M, K_{x_i}\rangle\big) + \lambda\langle M, M\rangle_H$$

It can be shown that independently of the form of the loss function, the solution of the variational problem can be expressed as:

$$M(x) = \sum_{i=1}^{n} \alpha_i k(x, x_i)$$

where k represents a kernel function. Table 13 shows kernel functions associated with a variety of surrogate modeling schemes. In particular, if the loss function is quadratic, the coefficients α_i can be found by solving the following linear system:

$$(n\lambda\boldsymbol{I} + \boldsymbol{K})\alpha_i = y_i$$

Where $\boldsymbol{K}$ denotes the so-called Gram matrix with component K_{ij} denoting $k(x_i, x_j)$, and $\boldsymbol{I}$ representing the identity matrix. Alternatively, if the ϵ-insensitive loss function is used, the coefficients α_i are found by solving a quadratic programming problem. See Schölkopf and Smola (2002) and Poggio and Smale (2003) for details.

Table 13. Kernel functions associated with a variety of modeling schemes

Kernel	Parametrization
Polynomial order d	$k(\boldsymbol{x}, \boldsymbol{x}') = (\langle \boldsymbol{x}, \boldsymbol{x}'\rangle + c)^d \quad d \in N,\ c \geq 0$
Spline	$k(\boldsymbol{x}, \boldsymbol{x}') = 1 + \langle \boldsymbol{x}, \boldsymbol{x}'\rangle + \frac{1}{2}\langle \boldsymbol{x}, \boldsymbol{x}'\rangle \min(\boldsymbol{x}, \boldsymbol{x}') - \frac{1}{6}\min(\boldsymbol{x}, \boldsymbol{x}')$
RBF	$k(\boldsymbol{x}, \boldsymbol{x}') = exp(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2/2h^2) \quad h > 0$

The implementation used in this study corresponds to *SVR* in the Scikit-learn library.

13.4. Gaussian processes

More specifically, a Kriging model (Sacks et al. 1989) postulates a combination of a polynomial model $f(\boldsymbol{x})$ and an error model of the form $Z(\boldsymbol{x})$ as:

$$\hat{y}(\boldsymbol{x}) = f(\boldsymbol{x}) + Z(\boldsymbol{x})$$

Where $\hat{y}(\boldsymbol{x})$ is the unknown function of interest, $f(\boldsymbol{x})$ is a regression function of $\boldsymbol{x}$, and $Z(\boldsymbol{x})$ is a random function (stochastic process) with mean zero, and non-zero covariance given by the following expression:

$$Cov\left[Z\left(x_i, x_j\right)\right] = \sigma^2 \boldsymbol{R}(x_i, x_j)$$

Where σ^2 is the process variance and $\boldsymbol{R}$ is the correlation function. In this work, consideration is given to constant regression and Gaussian correlation functions. The parameters of the regression and correlation functions are identified using a maximum likelihood estimator. The implementation used in this study corresponds to *GaussianProcessRegressor* in the Scikit-learn library.