Unsupervised local cluster-weighted bootstrap aggregating the output from multiple stochastic simulators

Imad Abdallah*, Konstantinos Tatsis*, Eleni Chatzi*

Department of Civil, Environmental and Geomatic Engineering, ETH Zürich
Stefano-Franscini-Platz 5, 8093 Zürich, Switzerland

Abstract

In the present work, we consider the problem of aggregating the output from multiple stochastic computer simulators to make inference on a stochastic quantity of interest, as a means of reducing the inherent model-form uncertainty in the absence of any validatory measurements. Inference by relying on the so-deemed best simulator may not be adequate as oftentimes it is impossible to judge the output from multiple simulators as being different when compared to each other for a given input. To this end, we propose an ensemble learning framework based on local *Clustering* and bootstrap aggregation (*Bagging*), which rather than treating the stochastic predictions of the simulators as competing individual information sources, treats those as part of an ensemble, thus diversifying the hypothesis space. Unsupervised variational Bayesian Gaussian mixture clustering is the first step in this ensemble learning approach for discriminating the diversified outputs, and deriving the probability map (weighting) of the clusters. Unsupervised clustering is performed on the stochastic output corresponding to the binned input space. Performing the clustering independently and deriving the probability map for each local region of the binned input space is a novelty that guarantees an adaptive solution, whereby certain simulators are potentially more fitting than others in corresponding regions of the input space. The second step consists in a local cluster-weighted Bootstrap Aggregation, which serves the purpose of weighted aggregation of the clustered ensemble of outputs from the individual simulators. Analytical demonstrations show that the cluster-weighted Bootstrap Aggregation outperforms conventional Bagging. In addition, contrary to classical Bagging, cluster-weighted Bagging is shown to be a robust technique, since it is less sensitive to input bin size, sample size, simulations output dispersion and level of agreement amongst the simulators. To our knowledge, the insights that local cluster-weighted Bagging can reduce model-form uncertainty has not been published in the context of stochastic simulators in the civil and mechanical engineering fields nor in the machine learning, mining or statistical communities.

Keywords: Ensemble learning, Clustering, Variational Bayesian Gaussian Mixture, Boostrap Aggregation (Bagging), Stochastic simulators, Model Uncertainty

*Corresponding author
Email address: `abdallah@ibk.baug.ethz.ch` (Imad Abdallah)

1. Introduction

xxx Complex physical systems and phenomena are studied by mathematical models, implemented as computer simulators. In many instances, relating to engineering analysis tasks, predictions from multiple stochastic computer simulators are available. Attention is generally directed towards comparing and benchmarking simulators against each other and against measurements in order to establish a single “true” or “best” simulator. This task has historically proven problematic when no simulator may emerge as clearly superior in terms of accuracy and precision. Consequently, basing inference regarding a quantity of interest on a single “best” simulator may underestimate the model-form uncertainty. The objective is to build the best possible aggregated numerical predictor of a stochastic quantity of interest based on the available output from multiple computer simulators. The scope of this work is to formulate and present the local cluster-weighted bootstrap aggregation method, and demonstrate its performance on analytical response functions and an engineering application.

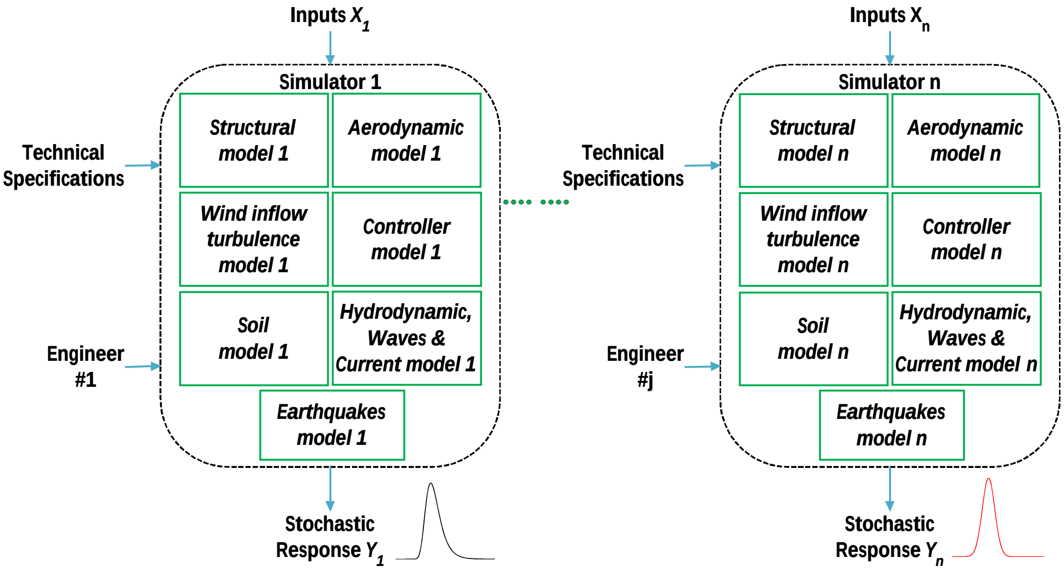


Figure 1: A depiction of a stochastic quantity of interest Y predicted by multiple computer simulators.

1.1. Problem statement

In this paper, we address the problem of aggregating the output from multiple stochastic computer simulators to make inference on a stochastic quantity of interest, as a means of reducing the inherent model-form uncertainty in the absence of any validatory measurements. We contextualize the problem by offering an example from the wind energy infrastructure domain. The analysis of wind turbines’ structures relies on stochastic aero-hydro-servo-elastic simulators. Figure 1 illustrates how a quantity of interest Y is predicted via n multi-physics simulators, which implement multiple solution methodologies to solve the same problem. For instance, the structural dynamics in one simulator may implement a multi-body/Finite Element formulation, and another simulator may implement a classical modal-superposition approach; The aerodynamics in one simulator may implement a Blade Element Moment theory, whereas the actuator line formulation may be adopted in another. It must be noted that, contrary to deterministic simulators, stochastic simulators yield different results when they are called repeatedly with the same input [1], such as shown in the synthetic analytical example in Figure 2.

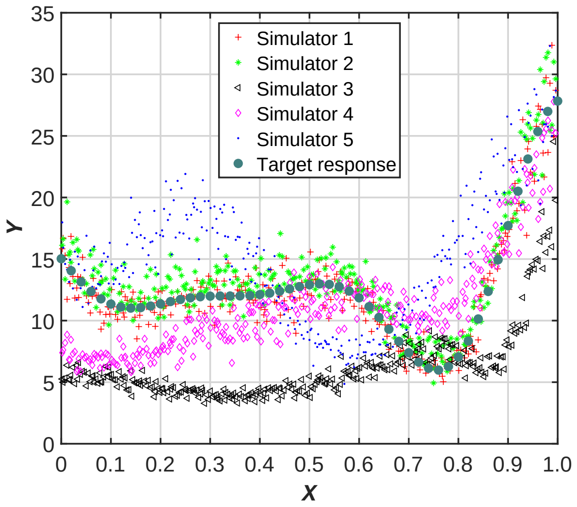


Figure 2: Output $\mathcal{Y}$ of five synthetic analytical stochastic computer simulators, sampled at 500 locations in the input space $\mathcal{X}$. The target response is the actual response of a certain physical system that each of the simulators is aiming to predict.

Historically, simulator-to-simulator comparisons have been performed with relevant results presented in a series of reports and publications; for instance [2, 3, 4, 5, 6, 7]. Simulators deployed in these comparison and benchmarking activities reflect the state-of-the-art, have been calibrated, verified and validated (i.e., none of the simulators exhibit flagrant bias or errors), and have been used in the design of real-world machines and structures, or adopted by decision makers. Thus, one is unable to definitively prove that a single simulator is superior to the others when considering all possible combinations of input environmental conditions, operating conditions and loading patterns on the various sub-structures of the wind turbines. This is rendered especially complicated when no validatory measurements are available of the physical system being simulated.

1.2. Contributions and novelties of this research

There is mounting evidence to suggest that aggregating the output from a diverse set of simulators may provide better predictive ability and lower model-form uncertainty than adoption of a any single simulator [8, 9]. This is very much in the spirit of model combination and model averaging, which has primarily been developed for the purpose of statistical inference [10]. The majority of model averaging or model aggregation methods in the literature assign weights that do not distinguish between different regions of the covariate input space [11], and make an inherent assumption that one of the models in the ensemble is the actual data-generating model. Furthermore, several methods require prior information about the model probabilities, prior distribution for the parameters or likelihood. Based on these reasons we turn to alternative approaches. We propose an unsupervised ensemble learning approach that rather than treating the predictions of the simulators as competing individual information sources, considers these as part of an ensemble. In the absence of any available measurements, and based solely on the available stochastic output from a diverse set of simulators, the approach consists in first deriving the simulators *local weights* using variational Bayesian Gaussian mixture clustering (VBGM), and subsequently performing a weighted ensemble aggregation on the stochastic output via Bootstrap Aggregation, also known as Bagging. Abdallah et al. [12, 13] demonstrated the potential of this method in early works, and here we present a complete treatise of the framework. The overarching intent is to provide proof that the proposed approach affects the model-form uncertainty, often an overlooked topic in uncertainty quantification. Compared to state-of-the-art, the primary contributions of the paper are listed as follows:

(i) Classical model averaging and model aggregation methods assume the availability of a large model space, which is not the case for practical multi-physics type of problems (e.g. weather forecasting, aero-hydro-servo-elastic loads predictions, etc.). The proposed method can work equally well for such scenarios.

(ii) Contrary to classical methods, our proposed method explicitly combines the models via the clustering step before proceeding to bootsrtapp aggregation based on each cluster weight.

(iii) Contrary to classical model averaging or model aggregation methods, where a single weight is assigned to a given simulators over the whole input space (each models predictions are weighted by the belief (posterior) that it is the true model over the whole input space), cluster-weighted Bagging assigns cluster-based weights independently, and in an unsupervised manner, for each local region of the binned input space. This guarantees a solution whereby certain simulators are more fitting than others in certain regions of the input space.

(iv) Furthermore, contrary to classical model averaging or model aggregation methods, cluster-weighted Bagging does not assign weights to individual simulators for each local region of the binned input space. Rather it assigns weights to clusters of simulators outputs, i.e., to the collection of "similar" output data points originating from the various simulators.

(v) The assignment of weights by classical global model averaging or model aggregation can be affected by sample bias or outliers in the global data set. In contrast, Our proposed method assigns weights independently for each local region of the binned input space and thus avoiding this issue.

(vi) Our proposed method is applicable in its current form, without any single modification or extension, for any number of simulators and any $n - dimensional$ input support space, $\forall \boldsymbol{X} \in \mathbb{R}^n$.

(vii) Our proposed method seamlessly assimilates any number of measurements replications (when these become available) without any further modifications to the method. Each measurement replication is considered as yet another "simulator" and added to the bucket of already available simulators and are not labelled as the ground truth.

1.3. Related work

Some of the more common approaches that have been used for combining the output from multiple simulators include the Multivariate Normal Aggregation method [14, 15, 16, 17], which aims at combining the normally distributed output from various simulators, while introducing dependence via a covariance matrix, resulting in effect in a multivariate Normal distribution of the output of individual simulators. A natural extension of this method is to use the Copula formalism instead to capture the dependence structure amongst the simulators.

Copula models for aggregation were originally proposed by Jouini and Clemen [18], who employ Copulas as the basis for modelling dependence among experts' opinions; here the experts' opinions are substituted for the output of simulators. The Adjustment Factor Approach is another method that makes use of an adjustment factor that is added (or multiplied) to the best model amongst all models considered [19, 20, 21, 22, 23]. The weight of the simulators is assigned by expert opinion based on the merit and accuracy of each individual simulator. The method however assumes that the outputs from individual simulators are Normally, Lognormally or Beta distributed and the adjustment factor is normal or lognormal.

Bayesian Model Averaging (BMA), can be used to combine the predictions from several numerical models through a model averaging procedure [24]. BMA accounts for model uncertainty, as distinct from parameter uncertainty, by integrating over the model space and weighting each individual model by the estimated probability of being the correct model. The weights (or probabilities) may begin with assumed subjective probabilities (e.g. expert judgement), which are then updated quantitatively using the training data set $\mathcal{D}$ [25]. The weighting factors for averaging are essentially related to the model performance according to the data, which is reflected via the relative likelihood of model $\mathcal{M}_j$ given the training data. In a Bayesian framework the weighting factors become Posterior Model Probabilities (PMP), which involves the calculations of a marginal likelihood. Apart from simple settings, this does not have a closed form and must be approximated using one of many numerical or analytical methods to solve the marginal likelihood, see [26, 27, 28, 29]. The final predictions of BMA are a weighted average of the set of model predictions, which Madigan and Raftery [30] and Hoeting et al. [10] (amongst many) demonstrated to provide better average predictive ability than using any single model for certain applications. Rings et al. [31] have introduced a variant of BMA with a joint particle filtering and Gaussian mixture modeling framework to derive the evolving forecast density of each constituent ensemble member. These distributions are subsequently combined with BMA and used to derive one overall predictive distribution. In a recent work, Liu et al. [32] proposed a Bayesian model averaging based reliability analysis method for monotonic degradation dataset based on inverse Gaussian process and Gamma process. Using BMA, they successfully introduced both the parametric and model uncertainties in the reliability analysis. Yu et al. [11] advanced the BMA by proposing the concept of local Bayes factors, where they calculate the Bayes factors by restricting the models to regions of the covariate space. The covariate input space is split in such a way that the relative model efficiencies of the various Bayesian models are similar in the same region, while differing in different regions. An algorithm for clustered Bayes averaging is then proposed for model combination, where local Bayes factors are used to guide the weighting of the Bayesian models.

However, the criticism of BMA is that it tries to assign weights presuming that one of the models in the ensemble is the actual data-generating model. Essentially, implicit to BMA is a model selection problem, which would work well if the ensemble were big enough to sample the entire model-space, but such is impossible for practical engineering applications. Alternatively, Bayesian Model Combination (BMC; [33]) provides an algorithmic correction to BMA in order to allow for the selection from an ensemble of combinations of the individual simulators, whose weights are obtained by sampling from a Dirichlet distribution with uniform parameters. Essentially the difference between BMA and BMC is that BMC marginalizes over the uncertainty in the correct model combination, where-as BMA marginalizes over the uncertainty in identifying the correct model from the entire ensemble [34]. Strategies for computing the weights are explained in [33]. BMC has proven to be superior to BMA and Bagging for certain applications.

An interesting sampling based approach is proposed by Peherstorfer et al. [35] to combining multiple surrogate models to accelerate failure probability estimation. The method is based on mixed multifidelity importance sampling that leverages computationally cheap but erroneous models for the construction of the biasing distribution and that uses the original high-fidelity model to guarantee unbiased estimates of the probability of failure using Importance sampling. An appealing feature of this approach, similar to the algorithm proposed in the current work, is that it avoids the problem of model selection while guaranteeing to achieve a small mean squared error.

In this work we opted to use Variational Bayesian Gaussian Mixture in order to identify the posterior distribution of the clusters conditional on the binned input space. The choice of approximate posterior distribution is one of the core problems in variational inference. Instead, Variational Autoencoders and Normalizing Flows is a new approach for specifying flexible, arbitrarily complex and scalable approximate posterior distributions compared to the known limitations of variational inference [36, 37].

Finally, the so-called class of the ensemble learning techniques include but are not limited to Bagging [38], Boosting, adaptive boosting (AdaBoost) [39] and Stacking [40]. These techniques are simple yet effective at reducing the error rate of unstable learners, in the sense that small variations in the training set can lead them to produce very different models. The general principle of ensemble learning methods is to construct a **combination** (not averaging) of some model fitting method (from a concrete finite set of alternative models), instead of adopting a single fit of the method [41].

1.4. Paper organization

The remainder of this article is organized as follows. In Section 2 we provide an overview of the proposed ensemble framework. We revisit the theory of Variational Bayesian Gaussian Mixture Clustering, and then derive the probability maps (simulators weights) in Section 3. In Section 4 we describe how the local cluster-weights are introduced to the Boostrapp Aggregation process. We demonstrate the novelty and principles of the proposed framework with a set of computational experiments in Section 5. Finally, in Section 7 we provide a practical engineering Engineering Application.

2. The ensemble learning algorithm

Assume that stochastic predictions from several simulators, for the same target quantity of interest $\mathcal{Y}$, are available as a function of $\boldsymbol{x}$. We let $\boldsymbol{x} = \{x_1, \ldots, x_d\}^T$ be a d-dimensional vector of input variables. This input vector is sampled at N distinct locations in the input space $\mathcal{D}_{\boldsymbol{x}}$ and the corresponding scalar stochastic output realizations are $\left\{\mathcal{Y}^{(i)}, i = 1, \ldots, N\right\}$. The N distinct input samples are collected in matrix $\boldsymbol{\mathcal{X}} = \left(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)}\right)$, whereas the output in vector $\boldsymbol{\mathcal{Y}} = \left\{\mathcal{Y}^{(1)}, \ldots, \mathcal{Y}^{(N)}\right\}^T$. For a given simulator $\{\mathcal{S}_l, l = 1, \ldots, s\}$ the vector of stochastic output is $\boldsymbol{\mathcal{Y}}_l$, s being the total number of available simulators.

Given this scenario, the two central questions we strive to answer are: how to take advantage and make use of the output from the various stochastic simulators? How can predictions from multiple simulators, demonstrably, result in reduced model-form uncertainties? To answer these questions we propose Algorithm 1, which we call local cluster-weighted bootstrap aggregation. A graphical illustration of the local Cluster-weighted Bagged ensemble learning algorithm is shown in Figure 3.

The algorithm starts by discretising the domain of the input space into a finite number of bins. The training set is all the response data falling within a specific bin. The learning set in each bin is then split into a training and a validation set. Unsupervised Variational Bayesian Gaussian Mixture clustering (VBGM) is then applied to data in each bin, and serves the purpose of distinguishing the output from each simulator, and deriving the probability map (weighting) of the simulators' output. VBGM is chosen because the number of clusters is an optimization variable and not given as input in advance. Performing the clustering independently for each local region of the binned input space guarantees an adaptive solution, whereby certain simulators are more fitting than others in corresponding regions of the input space. An intermediate step examines the cluster stability on perturbed versions of the data in a given bin. The reason for performing cluster stability analysis is that, as opposed to supervised classification, there is no ground truth against which we could “test our clustering results. The final major step of the algorithm consists in a local cluster-weighted Bootstrap Aggregation (Bagging), which serves the purpose of weighted aggregation of the clustered ensemble of outputs from the individual simulators.

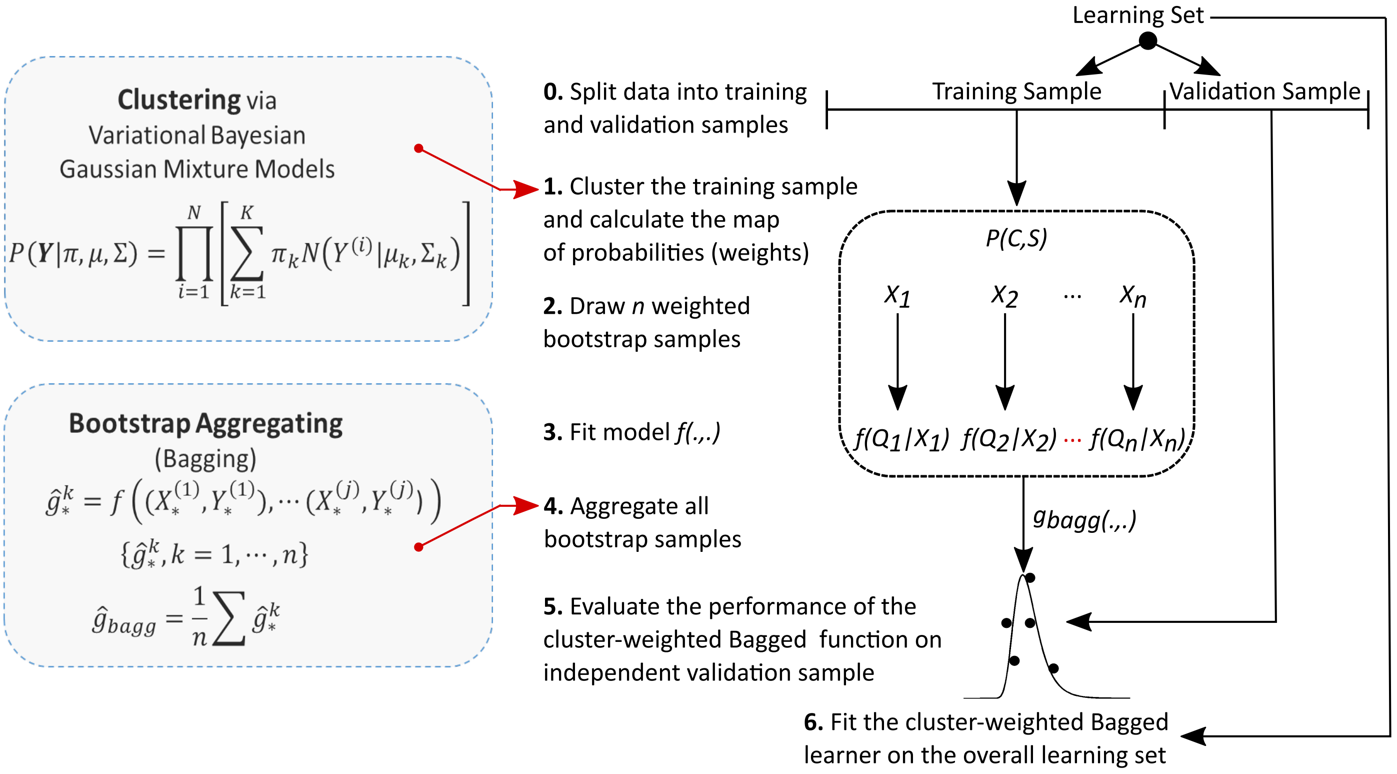

Clustering via
Variational Bayesian
Gaussian Mixture Models
$P(\boldsymbol{Y}|\pi,\mu,\Sigma) = \prod_{i=1}^{N}\left[\sum_{k=1}^{K}\pi_k N(Y^{(i)}|\mu_k,\Sigma_k)\right]$
Bootstrap Aggregating
(Bagging)
$\hat{g}_*^k = f\left((X_*^{(1)},Y_*^{(1)}),\cdots(X_*^{(j)},Y_*^{(j)})\right)$
$\{\hat{g}_*^k, k=1,\cdots,n\}$
$\hat{g}_{bagg} = \frac{1}{n}\sum\hat{g}_*^k$
Learning Set
Training Sample
Validation Sample
0. Split data into training and validation samples
1. Cluster the training sample and calculate the map of probabilities (weights)
2. Draw n weighted bootstrap samples
3. Fit model f(.,.)
4. Aggregate all bootstrap samples
5. Evaluate the performance of the cluster-weighted Bagged function on independent validation sample
6. Fit the cluster-weighted Bagged learner on the overall learning set
P(C,S)
X_1
X_2
...
X_n
$f(Q_1|X_1)$
$f(Q_2|X_2)$
...
$f(Q_n|X_n)$
$g_{bagg}(.,.)$

Figure 3: Graphical illustration of the local Cluster-weighted Bagged ensemble learning algorithm.

Algorithm 1: Local cluster-weighted bootstrap aggregating stochastic simulators.

```
Input  : $\{\boldsymbol{\mathcal{Y}}_l\}_{l=1}^s$: vector of stochastic observations from each simulator $\{\mathcal{S}_l\}_{l=1}^s$, $nMCs >> 1$: Number of
         repeats, $\boldsymbol{\mathcal{X}} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^N$: $d$-dimensional explanatory input variables i.i.d. draws from the joint
         distribution, $\Delta x$: input bin size, $Q$: a large pre-specified number, $M$: a large pre-specified number
Result: $\hat{g}_{w,bagg}$: cluster-weighted Bagged estimator
Set $\boldsymbol{\mathcal{Z}} = \left[\boldsymbol{\mathcal{X}}^T \left\{\boldsymbol{\mathcal{Y}}_l^T\right\}_{l=1}^s\right]$;
for $iter = 1, \ldots, nMCs$ do
    Randomly permute the cases in $\boldsymbol{\mathcal{Z}}$, and then select $n$ training samples such that $n = 0.7N$;
    Assign every $\left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^n$ to be the center of bins of size $\Delta x$;
    foreach $\Delta \boldsymbol{x}$ do
        for $q = 1, ..., Q$ do // Clustering
            Randomly draw 90% of local responses of simulators $\left\{\boldsymbol{\mathcal{Y}}_l^T\right\}_{l=1}^s$ without replacement;
            Identify local clusters $C_q\left(\left\{\boldsymbol{\mathcal{Y}}_l^T\right\}_{l=1}^s, K_q \mid \Delta \boldsymbol{x}\right)$, using Variational Bayesian Gaussian Mixture;
            Compute local weights $P_q\left(\mathcal{S}, C \mid \Delta \boldsymbol{x}\right)$;
        end
        // Expected local cluster weights
        Use majority vote $out - of - Q$ to establish the expected local number of clusters $K_{\Delta \boldsymbol{x}}$;
        Compute expectation of local weights: $P\left(\mathcal{S}, C \mid \Delta \boldsymbol{x}\right) = \mathbb{E}\left[P_q\left(\mathcal{S}, C \mid \Delta \boldsymbol{x}\right)\right]_Q$;
        for $k = 1, ..., M$ do // Bagging
            Construct a weighted bootstrap sample by randomly drawing $n$ times with replacement from
            the clustered data $\left(\boldsymbol{x}^{(1)}, \mathcal{Y}^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(n_{\Delta x})}, \mathcal{Y}^{(n_{\Delta x})}\right)$, the local weights being $P\left(\mathcal{S}, C \mid \Delta \boldsymbol{x}\right)$;
            Compute the bootstrapped estimator trained separately based on each bag:
                $$\hat{g}_*^k = h_n\left(\left(\boldsymbol{x}_*^{(1)}, \mathcal{Y}_*^{(1)}\right), \ldots, \left(\boldsymbol{x}_*^{(n_{\Delta x})}, \mathcal{Y}_*^{(n_{\Delta x})}\right)\right)$$
            where the function $h_n(\cdot)$ defines the estimator as a function of the data;
        end
        Compute the cluster-weighted Bagged estimator as the expectation of all the bootstrap estimators:
            $$\hat{g}_{w,bagg} = \frac{\sum_{k=1}^M \hat{g}_*^k}{M}$$
    end
end
```

3. Variational Bayesian Gaussian Mixture Clustering

The clustering step in the proposed ensemble learning approach serves the purpose of (1) distinguishing the output from each simulator (or model) and (2) deriving an *apriori* probability map (weights) of the simulators in the absence of any validatory measurements data.

A number of clustering algorithms have been proposed in the literature, such as *k-means*, *Gaussian Mixtures*, *Hierarchical* clustering [42, 43], self organizing maps (SOMs), and deep unsupervised clustering with Gaussian Mixture variational autoencoders. Here we chose a Bayesian treatment using Variational Bayesian Gaussian Mixture Clustering (VBGM).

3.1. Overview of VBGM

Finite Gaussian mixtures are a flexible probabilistic modeling tool for irregularly shaped densities and samples from heterogeneous stochastic populations generated from a mixture of a finite number K of Gaussian distributions with unknown parameters:

$$p\left(\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \tag{1}$$

where π_k is the mixing coefficient, and each Gaussian density $\mathcal{N}\left(\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)$ is a component of the mixture comprising its own parameters: mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}_k$ [44]. The traditional approach to estimating the parameters is by maximizing the likelihood function of the Gaussian mixture:

$$p\left(\boldsymbol{\mathcal{Y}} \mid \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) = \prod_{i=1}^{N} \left[\sum_{k=1}^{K} \pi_k \mathcal{N}\left(\boldsymbol{\mathcal{Y}}^{(i)} \mid \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\right] \tag{2}$$

which is not a well posed problem because of the singularities that will always occur whenever one of the Gaussian components “collapses” onto a specific data point in the dataset (see [44] for proof). Training of Gaussian mixture models is faced with the classic chicken-and-egg problem that most unsupervised learning algorithms face. If we know which component a sample belongs to, we can use the Maximum Likelihood Estimation (MLE) estimates to update the component. Conversely, if we know the parameters of the components we can predict where each sample lies in each component. This problem is solved using the expectation-maximization (*EM*) algorithm [45], which iterates between the two until convergence. *EM* is a well-founded statistical algorithm for getting around these problems by means of an iterative process in order to determine maximum likelihood solutions. Formally, the algorithm begins with an initial estimate of the parameters vector (including the mixing coefficients) and then alternates between two steps [42]: an expectation step (*E-step*), in which the conditional expectation of the log-likelihood function is computed given the observed data and the current parameter estimates, and a maximization step (*M-step*), in which the parameters that maximize the expected likelihood from the *E-step* are determined. Consequently, the EM algorithm represents each cluster using a Gaussian probability density function (for details see [44] or [46]). A limitation of *EM* is that it requires a proper/multiple initialization(s) in order to consistently find good maximum likelihood solutions. Another limitation is that there is no clear guidance on the choice of the number of Gaussian mixtures (components) K to be used. The Bayesian variational treatment of the Gaussian mixture model circumvents this by automatically providing the number of clusters determined by model selection. The variational framework can be viewed as a complementary approach to that of Markov Chain Monte Carlo (MCMC), which delivers considerable computational advantages at the cost of not being asymptotically exact [47]. In practical terms, the ultimate objective is to cluster the data into K components, each of which comprises a mixing coefficient $\{\pi_k, k = 1, \ldots, K\}$. The end goal is to evaluate the posterior distribution:

$$P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)=\frac{P\left(\boldsymbol{Y},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)}{P\left(\boldsymbol{Y}\right)}=P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\frac{P\left(\boldsymbol{Y}\,\middle|\,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)}{P\left(\boldsymbol{Y}\right)} \tag{3}$$

which is generally intractable. Variational methods are rather used in order to define a tractable lower bound on $P\left(\boldsymbol{Y}\right)$, by minimizing the Kullback-Leibler divergence $KL\left(q(\boldsymbol{\Theta})\,\|\,P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)\right)$. Replacing the posterior distribution, it turns out that the following approximation holds true [48, 49]:

$$\begin{aligned} ln\,P\left(\boldsymbol{Y}\right) &= ln\,\frac{P\left(\boldsymbol{Y},\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\right)}{P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)} \\ &= \underbrace{\mathcal{F}\left(q\left(\boldsymbol{\Theta}\right)\right)}_{\text{Neg. free energy}} + \underbrace{KL\left(q\left(\boldsymbol{\Theta}\right)\,||\,P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)\right)}_{\text{KL divergence}} \end{aligned} \tag{4}$$

where $\mathcal{F}\left(q\left(\boldsymbol{\Theta}\right)\right)$ is the so-called negative free energy, which is a lower bound approximation of $P\left(\boldsymbol{Y}\right)$. The variational method involves the introduction of a distribution $q(\boldsymbol{\Theta})$, which provides an approximation to the true posterior distribution, where $\boldsymbol{\Theta}$ is a vector collecting all parameters. $q(\boldsymbol{\Theta})$ generally corresponds to a simple parametric family of the posterior probability density. In this regard, the Kullback-Leibler divergence is often chosen as a relative measure of the dissimilarity of the two probability densities $p\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)$ and $q(\boldsymbol{\Theta})$[50]. Minimization of $KL\left(q(\boldsymbol{\Theta})\,||\,P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)\right)$ reduces the divergence between the true posterior $p\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)$ and its approximation $q\left(\boldsymbol{\Theta}\right)$. The problem then reduces to finding the set of probability densities $q(\boldsymbol{\Theta})$ that maximize the lower bound $\mathcal{F}\left(q\left(\boldsymbol{\Theta}\right)\right)$, which is equivalent to minimizing the $KL\left(q(\boldsymbol{\Theta})\,||\,P\left(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Sigma}\,\middle|\,\boldsymbol{Y}\right)\right)$ or tightening $\mathcal{F}$ as a lower bound to the log model evidence $ln\,P\left(\boldsymbol{Y}\right)$. An iterative algorithm is usually used to solve the maximization problem [51]. In this work, the algorithm in [52, 53] is adopted. Further details on the Variational Bayes (inference) can be found in [52, 54, 44, 55], while for applications to Gaussian mixture models the interested reader is referred to [53, 56, 57, 58].

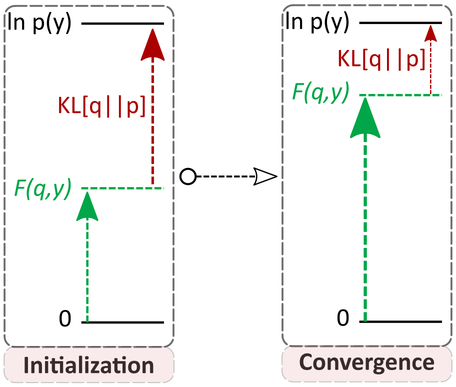


Figure 4: Illustration of the KL Divergence [47].

3.2. *Probability map (weights)*

As mentioned, the clustering step in the ensemble learning approach serves the purpose of (1) distinguishing the output from each simulator (or model structure) and (2) deriving *apriori* probability map (weights) of the simulators in the absence of any validatory measurements data. Let us elaborate via use of a simple example where we cluster the stochastic output from 3 simulators for a given sample $\mathcal{X}^{(k)}$ from an N-dimensional input space $\mathcal{X}^{(i)}$. The output from each of the simulators consists of twelve data points, which we assume are clustered as shown in Figure 5.

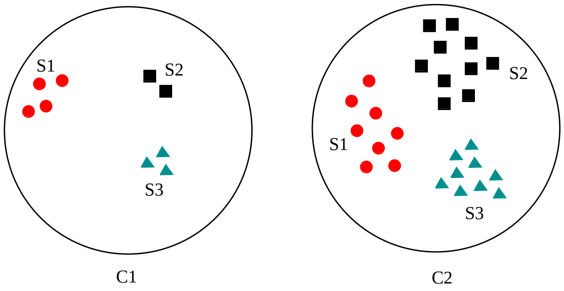


Figure 5: Clustering of the stochastic output from 3 simulators for a given sample $\mathcal{X}^{(j)}$ from an n-dimensional input space $\mathcal{D}_{\mathcal{X}}$. The output from each of the simulators consists of 12 data points.

The weight should not be tied the total probability of a simulator as this is not very relevant. Why? If we have 12 output data points, from each simulator as in Figure 5, then the $P(\mathcal{S}_1) = 1/3$, $P(\mathcal{S}_2) = 1/3$ and $P(\mathcal{S}_3) = 1/3$, which does not offer any additional information. On the other hand, the weight should not be assigned by the conditional probabilities $P\left(\mathcal{S}_i \mid Cj, \Delta\boldsymbol{x}\right)$, since this ignores the probability of a cluster itself. The idea would be to push the aggregate toward larger density clusters (i.e., areas of stronger agreement), allowing the models to reinforce each other when consensus exists, or, conversely, negate each other when there is no consensus. Further research is needed here in order to address the case of a correct simulator output being clustered separately from the larger density clusters for certain combinations of input $\boldsymbol{x}^{(i)}$ in the d-dimensional input space $\mathcal{D}_{\boldsymbol{x}}$. In other words, the derived weights may be inadequate because the method fails to recognize certain simulators are more fitting than others in certain regions of the input space. Based on the clusters depicted in Figure 5, we can calculate the weights, which are the joint probability $P\left(\mathcal{S}_i, C_j \mid \Delta\boldsymbol{x}\right)$ of a cluster $\{C_j; j = 1, 2\}$ and a simulator $\{\mathcal{S}_i; i = 1, 3\}$:

$$P\left(\mathcal{S}_i, C_j \mid \Delta\boldsymbol{x}\right) = P\left(\mathcal{S}_i \mid C_j, \Delta\boldsymbol{x}\right) \cdot P\left(C_j \mid \Delta\boldsymbol{x}\right) \tag{5}$$

where,

$$P\left(C_j \mid \Delta\boldsymbol{x}\right) = \frac{N_{C_j \middle| \Delta\boldsymbol{x}}}{N} \tag{6}$$

N_{C_j} is the number of data points in cluster C_j and N is the total number of all available output samples from all simulators, and $P\left(\mathcal{S}_i \mid C_j, \Delta\boldsymbol{x}\right)$ is cast as:

$$P\left(\mathcal{S}_i \mid C_j, \Delta\boldsymbol{x}\right) = \frac{N_{\mathcal{S}_i|C_j,\Delta\boldsymbol{x}}}{N_{C_j \middle| \Delta\boldsymbol{x}}} \tag{7}$$

where, $N_{\mathcal{S}_i|C_j,\Delta\boldsymbol{x}}$ is the number of data points in cluster C_j corresponding to simulator $\mathcal{S}_i$. The outcome is shown in Figure 6. The Bayesian Gaussian Mixture clustering identified 9 data points belonging to cluster 1 and 27 data points belonging to cluster 2. In cluster 1, 4 data points correspond to simulator 1, 2 data points correspond to simulator 2 and 3 data points correspond to simulator 3. In cluster 2, 8 data points correspond to simulator 1, 10 data points correspond to simulator 2 and 9 data points correspond to simulator 3. Finally, the joint probabilities of a cluster and a simulator are shown in Table 1.

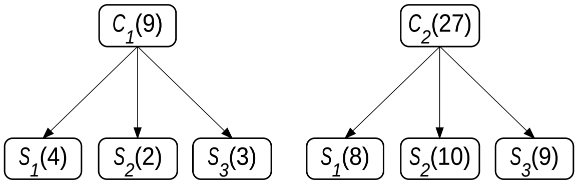


Figure 6: Splitting of of the 3 simulators output according to clusters 1 and 2.

Table 1: Joint probability of simulator and cluster.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$
C1	0.111	0.056	0.083
C2	0.222	0.278	0.250

4. Local Cluster-Weighted Bootstrap Aggregating

The clustering step is then followed by the weighted Bootstrap Aggregating (Bagging) step, which serves the purpose of weighted aggregation of the ensemble of outputs from the individual simulators. The weights are indeed derived form the clustering step as shown in the previous section. Bagging predictors comprise a method for generating multiple versions of a predictor, each on random subsets of the original dataset, and fusing these into a unique final aggregated predictor. This aggregated predictor can typically be used for reducing the variance of black-box estimators, by introducing randomization into the construction procedure and forming an ensemble (for proof refer to [38, 41]). The bagging algorithm consists in:

[1] constructing a weighted bootstrap sample $\left(\boldsymbol{x}_*^{(1)}, \mathcal{Y}_*^{(1)}\right), \ldots, \left(\boldsymbol{x}_*^{(n_{\Delta x})}, \mathcal{Y}_*^{(n_{\Delta x})}\right)$ by randomly drawing $n_{\Delta \boldsymbol{x}}$ times with replacement from the local clustered data $\left(\boldsymbol{x}^{(1)}, \mathcal{Y}^{(1)}\right), \ldots, \left(\boldsymbol{x}^{(n_{\Delta x})}, \mathcal{Y}^{(n_{\Delta x})}\right)$, the local weights being $P\left(\mathcal{S}_i, C_j \mid \Delta \boldsymbol{x}\right)$,

[2] computing the bootstrapped estimator trained separately for each bag:

$$\hat{g}_* = h_n\left(\left(\boldsymbol{x}_*^{(1)}, \mathcal{Y}_*^{(1)}\right), \ldots, \left(\boldsymbol{x}_*^{(n_{\Delta x})}, \mathcal{Y}_*^{(n_{\Delta x})}\right)\right) \tag{8}$$

where the function $h_n\left(\cdot\right)$ defines the estimator as a function of the data,

[3] and (3) repeating steps 1 and 2 M times, where M is often chosen between 50 and 100, yielding $\left\{\hat{g}_*^k, k = 1, \ldots, M\right\}$ and the weighted bagged estimator is defined as the expectation of all the bootstrap estimators:

$$\hat{g}_{w,bagg} = \frac{\sum_{k=1}^{M} \hat{g}_*^k}{M} \tag{9}$$

The estimator $h_n\left(\cdot\right)$ may be formulated a function that computes the expected value (mean) of a dataset, it may be a function that fits a parametric (or non-parametric) probability distribution to a data, or may even be a function that fits surrogate or auto-regressive models. Therefore, $h_n(\cdot)$ can be any learning algorithm $\Psi : \gamma \rightarrow \boldsymbol{\phi}$ that given any input dataset $L \in \gamma$, it produces a predictor $\Phi = \Psi(L) \in \boldsymbol{\phi}$. $h_n(\cdot)$ is designated as the base learner when it is is trained with all original data. Empirically, bagging has been well acknowledged to achieve better performance than the base learner, especially when the base learner is unstable with respect to the random training data. Meanwhile, theoretical investigations [59] offer a theoretical explanation as to how bagging reduces the asymptotic variance and mean squared error for some non-smooth and unstable predictors.

5. Computational Experiments

We present idealized analytical examples to motivate the approach, and demonstrate the principles and improvements that are possible with the local cluster-weighted bootstrap aggregation algorithm. Suppose that the target (true) expected response of a certain physical system is given by an analytical function of the form [60]:

$$y_{target}(x) = (6x-2)^2 \sin(12x-4) \tag{10}$$

where the input parameter x varies over $[0,1]$. Furthermore, we assume that the response is stochastic and Normally distributed conditional on x:

$$P\left(\mathcal{Y}_{target} \mid X\right) \sim \mathcal{N}\left(y_{target}(x), COV = 0.05\right) \tag{11}$$

where COV stands for the coefficient of variation, which we assume to be homoscedastic over the input random variable X, which is uniform over the $[0,1]$ support, $X \sim U(0,1)$. We propose five synthetic analytical stochastic simulators aiming at predicting $\mathcal{Y}_{target}$. In a real application these five simulators might take the form of multi-physics numerical models, surrogate models, auto-regressive models, etc. The simulators may be based on varying mathematical formulations, computational methods, and assumptions and simplifications of the physics describing the problem. In this demonstration, the cluster-weighted bagged estimator is the expected value of the stochastic response conditional on binned input space Δx and the output from the five stochastic simulators: $\mathbb{E}\left[\mathcal{Y} \mid \Delta x, \mathcal{Y}_1, \mathcal{Y}_2, \mathcal{Y}_3, \mathcal{Y}_4, \mathcal{Y}_5\right]$. We compare our results to classical bagging. This is a sensible comparison since we want to understand if and when using unsupervised cluster-based weighted Bagging outperforms classical bagging. The comparison is based on (1) the generalization error with respect to the known target response and (2) change in the 95% empirical bootstrap confidence interval. The generalization error GE is a global metric of the accuracy of the surrogate model [61]. It is computed as:

$$GE = \frac{\sum_{i=1}^{n_v}\left[\mathcal{Y}^{(i)} - \mathcal{M}^K\left(\boldsymbol{x}^{(i)}\right)\right]^2}{\sum_{i=1}^{n_v}\left[\mathcal{Y}^{(i)} - \mu_{\mathcal{Y}}\right]^2}, \quad \mu_{\mathcal{Y}} = \frac{1}{n}\sum_{i=1}^{n_v}\mathcal{Y}^{(i)} \tag{12}$$

where n_v is the size of the validation set, and $\mu_{\mathcal{Y}}$ is the mean of the computer simulator response for the validation set. On the other hand the 95% empirical bootstrap confidence interval (CI_{95}) reflects the local predictive precision and is given by:

$$CI_{95} = \left[\hat{g}_{bagg} - \delta^*_{0.025}, \hat{g}_{bagg} - \delta^*_{0.975}\right] \tag{13}$$

$$\begin{cases} \delta^*_k = \hat{g}^k_* - \hat{g}_{bagg}, \left\{\hat{g}^k_*, k = 1, \ldots, 100\right\} \\ \textit{Order } \delta^*_k \text{ and pick out the .975 and .025 critical values, } \delta^*_{0.975} \text{ \& } \delta^*_{0.025}, \textit{ respectively} \end{cases} \tag{14}$$

In the following case studies, we analyse the performance when varying the number of simulators in agreement with the target response, bias of simulators with respect to target response, COV, number of stochastic samples in the input space, and bin sizes Δx. 100 Bootstraps are used in the weighted Bagging step. The results of the evaluations are averaged across 50 repetitions, so that they are not susceptible to a specific split of the sampled data. It is important to note that in the following case studies the target response does not represent a linear combination of the individual simulators.

5.1. Case study 1: Majority of simulators in perfect agreement with target response

In this case study four out of five simulators are in perfect agreement amongst each other and with the target response, i.e., $\left\{P\left(\mathcal{Y}_i \mid X\right) \sim \mathcal{N}\left(y_{target}\left(x\right), COV = 0.05\right), i = 1, \ldots, 4\right\}$. Simulator 5 is in disagreement with the target, and its expected response is described by an analytical transformation function of the form:

$$y_5(x) = Ay_{target}(x) + B(x - 0.5) - C \tag{15}$$

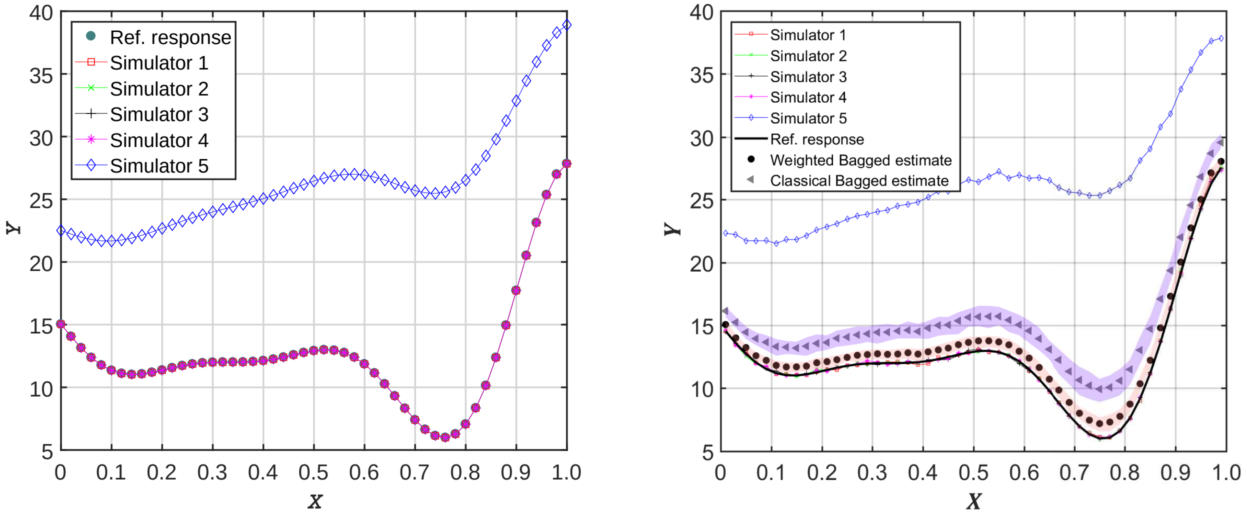
where the constants are set to $A = 0.5$, $B = 10$ and $C = 20$. We choose the constants such that $y_5(x)$ largely displays the same trend as y_{target} but exhibit a bias. The stochastic output of simulator 5 is given by $P\left(\mathcal{Y}_5 \mid X\right) \sim \mathcal{N}\left(y_5\left(x\right), COV = 0.05\right)$. The performance of the cluster-weighted Bagged estimator is evaluated when 2000 observational samples are available from each simulator, and $\Delta x = 0.02$. The expected values of the target and the five simulators' responses are shown in Figure 7a. The formation of clusters conditional on the binned input space is the first step in the proposed ensemble learning framework. An example of clustering and their corresponding log-evidence for $x \in [0.6, 0.62]$ are illustrated in Figures 8a and 8b, respectively. As per intuition, for $x \in [0.6, 0.62]$ two clusters, comprising the highest log-evidence, are the most likely grouping of the output predictions from the 5 simulators, yielding clusters weights (probability map) $P\left(S_i, C_j \mid X\right)$ as shown in Table 2.

Table 2: Weights $\left\{P\left(S_i, C_j \mid X\right)\ i = 1, \ldots, 5, j = 1, \ldots, 2\right\}$ for $x \in [0.6, 0.62]$.

	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$	$\mathcal{S}_4$	$\mathcal{S}_5$
Cluster 1	0.0	0.0	0.0	0.0	0.2
Cluster 2	0.2	0.2	0.2	0.2	0.0

The weights $\left\{P\left(S_i, C_j \mid X\right)\ i = 1, \ldots, 5, j = 1, 2\right\}$ for each Δx in the input space are then used in the cluster-weighted Bagged estimation step. The cluster-weighted Bagged estimator, shown in Figure 7b, yields a notably improved predictor of the target (true) response compared to conventional Bagging. We report that the generalization error is reduced by 90%, while the 95% confidence interval is reduced by 40% averaged across the binned input space x. A reduction in the generalization error translates into a better accuracy of the estimator, while a reduction in the confidence interval, for a given confidence level, translates into a better precision of the estimator.

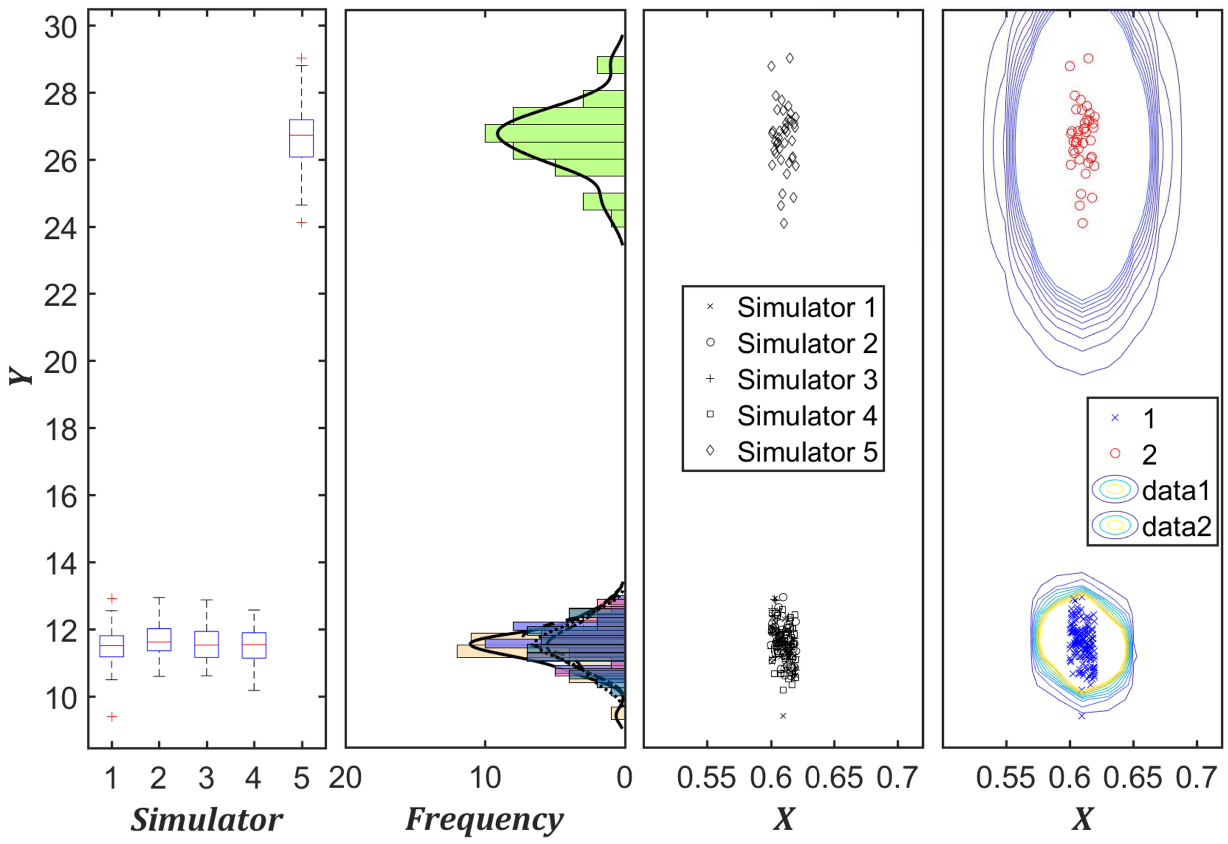
Furthermore, we investigate the cases when 3 out of 5, 2 out of 5 and 1 out of 5 simulators are in perfect agreement with the target response, while the remaining simulators are in perfect mutual agreement and take the form shown in Equation 15. The results shown in Figure 9b are not surprising and indicate that the generalization error of the cluster-weighted Bagged estimator drops by $40 - 90\%$ compared to the classical Bagged estimator when the majority of simulators agree with the target response. Otherwise, the classical Bagged estimator exhibit a better generalization error. Figure 9a shows that the 95% confidence interval decreases by $35 - 40\%$ when the majority of simulators are in mutual agreement and not necessarily in agreement with the target response. When 2 out of 5 or 3 out 5 simulators are in mutual agreement and further in agreement with the target response (i.e., no clear established majority), the reduction in confidence interval is rather marginal and of the order of 5%. Those results can be interpreted by the fact that the weights $P\left(S_i, C_j \mid X\right)$ are high when the majority of simulators are in mutual agreement, and not necessarily with the target response.



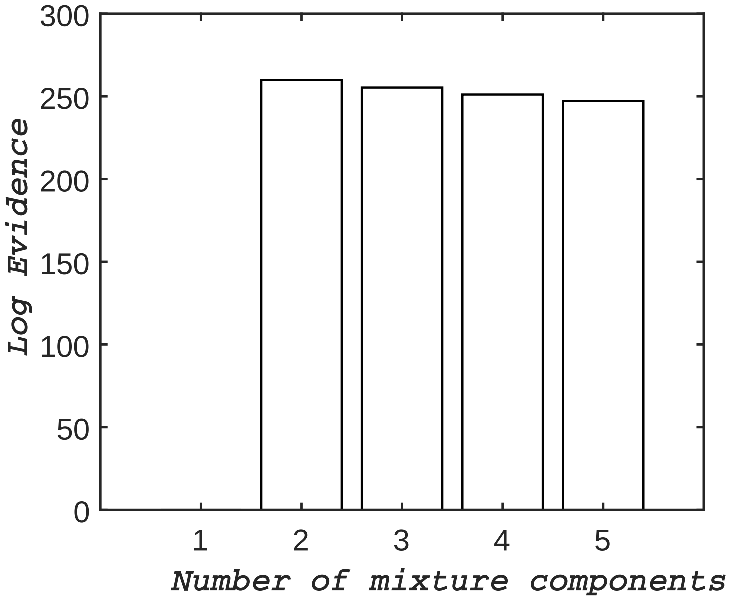
(a)

(b)

Figure 7: Case study 1: (a) The expected values of the target and the five simulators' responses. (b) The cluster-weighted Bagged estimator.

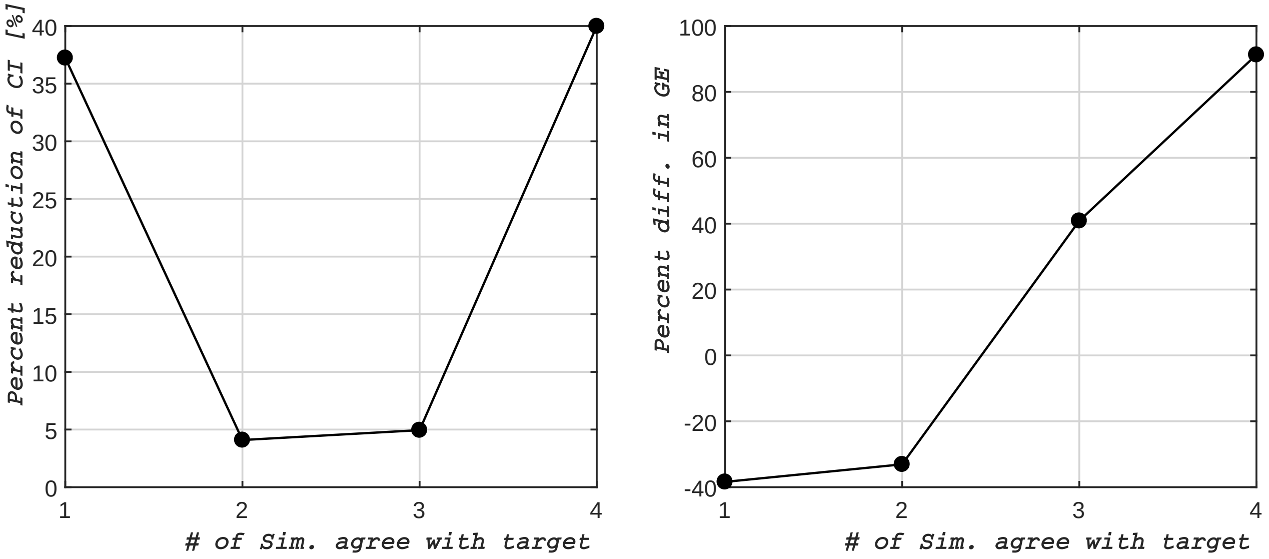


(a)



(b)

Figure 8: Case study 1: (a) Clustering of response for $x \in [0.6, 0.62]$. (b) Log-evidence of clusters.



(a)

(b)

Figure 9: Case study 1: Effect of the number of simulators in perfect agreement with target (true) response on (a) percent reduction of 95% confidence interval (CI), and (b) percent reduction in generalization error (GE), across 50 repeats of the cluster-weighted bagged estimator (100 bootstraps), compared to classical bagged estimator for 2000 samples from each simulator.

5.2. Case study 2: Minority of simulators in mutual disagreement

In this case study three simulators are in perfect agreement amongst each other and with the target response, i.e., $\left\{P\left(\mathcal{Y}_i \mid X\right) \sim \mathcal{N}\left(y_{target}\left(x\right), COV = 0.05\right), i = 1, \ldots, 3\right\}$. However, unlike case study 1, simulators 4-5 are in mutual disagreement and in disagreement with the target response. The expected responses of simulators 4-5 are shown in Figure 10a and are described by analytical transformation functions:

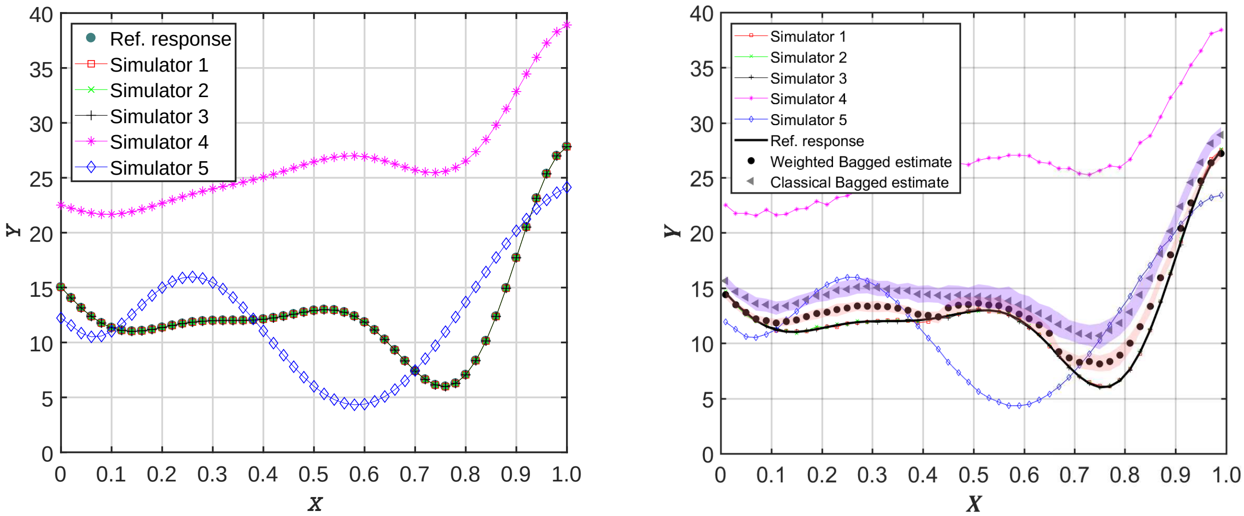
$$\begin{aligned} y_4(x) &= A_4 y_{target}(x) + B_4(x - 0.5) - C_4 \\ y_5(x) &= A_5 y_{target}(x) + B_5(x - 0.5) - C_5 \end{aligned} \tag{16}$$

The constants are set to $\{A_4 = 0.5, B_4 = 10, C_4 = 20\}$, and $\{A_5 = 0.25, B_4 = 0.5, C_5 = 9\}$. We choose the constants such that $y_4(x)$ largely displays the same trend as y_{target} but exhibits a bias, and $y_5(x)$ is non-linear for $x \in [0.1, 0.6]$ when the actual target response is linear in the same input range. The expected values of the target and the five simulators' responses are shown in Figure 10a. The stochastic predictions of simulators 4-5 are given by:

$$\begin{aligned} P\left(\mathcal{Y}_4 \mid X\right) &\sim \mathcal{N}\left(y_4\left(x\right), COV = 0.05\right) \\ P\left(\mathcal{Y}_5 \mid X\right) &\sim \mathcal{N}\left(y_5\left(x\right), COV = 0.05\right) \end{aligned} \tag{17}$$

The cluster-weighted Bagged estimator is shown in Figure 10b yielding a much improved predictor of the target (true) response compared to conventional Bagging. The performance of the cluster-weighted Bagged estimator is evaluated when 2000 observational samples are available from each simulator, and $\Delta x = 0.02$. We report that the generalization error is reduced by 83% while the 95% confidence interval is reduced by 32%

averaged across the binned input space x. In case study 1 we idealized a situation when 2 out of 5 simulators are in perfect mutual agreement, and at the same time in perfect disagreement (in unison) with the target response, the reduction in confidence interval was rather marginal and of the order of 5%. Here, the 2 simulators are in mutual disagreement and with the target response, resulting instead in the reduction of the confidence interval of the cluster-weighted Bagged estimator by 32%. In a sense, this is a result of diversifying the hypothesis space.



(a)

(b)

Figure 10: Case study 2: (a) The expected values of the target and the five simulator responses. (b) The cluster-weighted Bagged estimator.

5.3. *Case study 3: Majority of simulators in biased agreement with target response*

In this case study four out of five simulators are biased with respect to the the expected value of the target response, and are defined as follows:

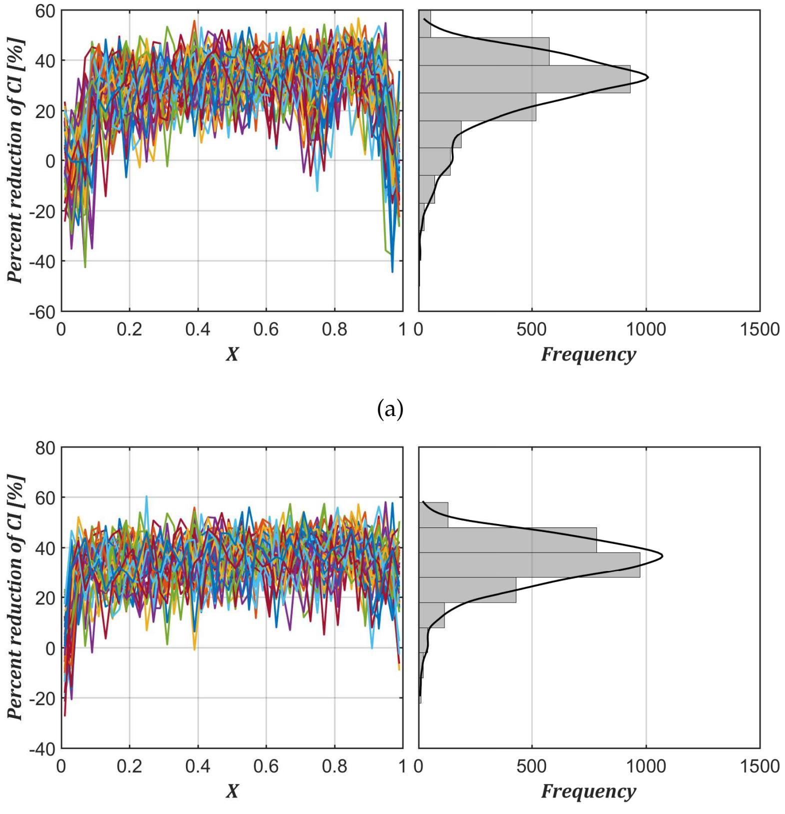
$$
\begin{aligned}
P\left(\mathcal{Y}_1 \,\middle|\, X\right) &\sim \mathcal{N}\left(y_{target}\left(x\right) - 2, COV = 0.05\right)\\
P\left(\mathcal{Y}_2 \,\middle|\, X\right) &\sim \mathcal{N}\left(y_{target}\left(x\right) - 1, COV = 0.05\right)\\
P\left(\mathcal{Y}_3 \,\middle|\, X\right) &\sim \mathcal{N}\left(y_{target}\left(x\right) + 2, COV = 0.05\right)\\
P\left(\mathcal{Y}_4 \,\middle|\, X\right) &\sim \mathcal{N}\left(y_{target}\left(x\right) + 3, COV = 0.05\right)
\end{aligned}
\tag{18}
$$

Simulator 5 is the sole predictor in significant disagreement with the true, and takes the form shown in Equation 15. In case study 1 we idealized a situation when 4 out of 5 simulators are in mutual agreement and in perfect agreement with the target response, resulting in the generalization error of the cluster-weighted Bagged estimator to be 90% lower compared to that of classical Bagged estimator, while the 95% confidence interval is reduced by 40% averaged across the binned input space x. Here, instead, we introduce a slight bias in the simulators predictions and We report that the generalization error is reduced by 77% instead of 90% while the 95% confidence interval is reduced by 28% instead of 40% averaged across the binned input space x.

According to Figure 11a, the percent reduction of the predictor 95% confidence interval across 50 repeats (with 100 bootstrapps) deteriorates for $x \in [0, 0.1]$ and $x \in [0.9, 1]$. It is useful to notice that simulator 5 is nearest to the target response and to the biased simulators 3 and 4 in this region of the input space. As a consequence, clustering via Variational Bayesian Gaussian Mixture is challenged to discriminate clear clusters of majority simulators, and hence the probability map (weights) defaults to those of classical Bagging, i.e., equal weights for all simulators conditional on the binned input space. When stochastic data are not well separated, the notion of a cluster is not anymore well defined.

Reducing the bias of simulators 3 and 4 by 0.75 points closer to the target (true) response, results in the confidence interval being reduced by 34% instead of 28%. This improvement is especially apparent for $x \in [0.9, 1]$ as depicted in Figure 11b as compared to Figure 11a. This reflects an improvement in discriminating amongst the simulators in the clustering step; Figure 12a shows the clusters of the response for $x \in [0.96, 0.98]$ for the original bias of simulators 3 and 4, and Figure 12b shows the clusters of the response for $x \in [0.96, 0.98]$ when the bias of simulators 3 and 4 is reduced by 0.75 points.

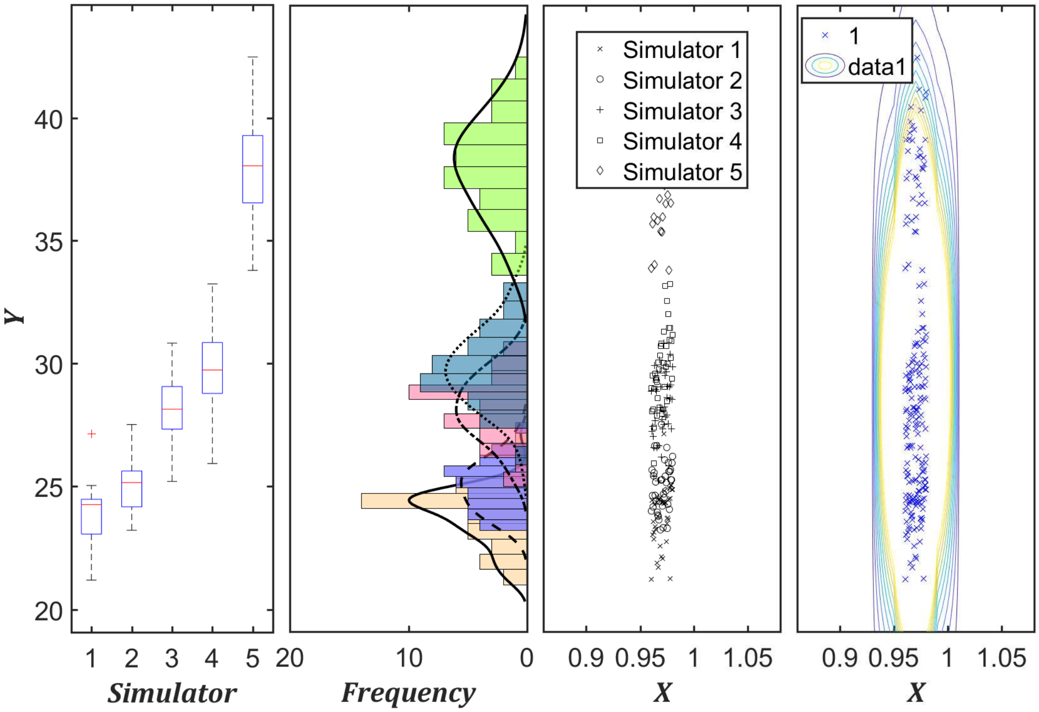
This case study showcases how local weighting stands in contrast to other methods, such Bayesian Model Averaging, which assign global weights to each model. Other effects contributing to the performance of the clustering step such as the COV, number of stochastic samples and the bin size are analysed next.



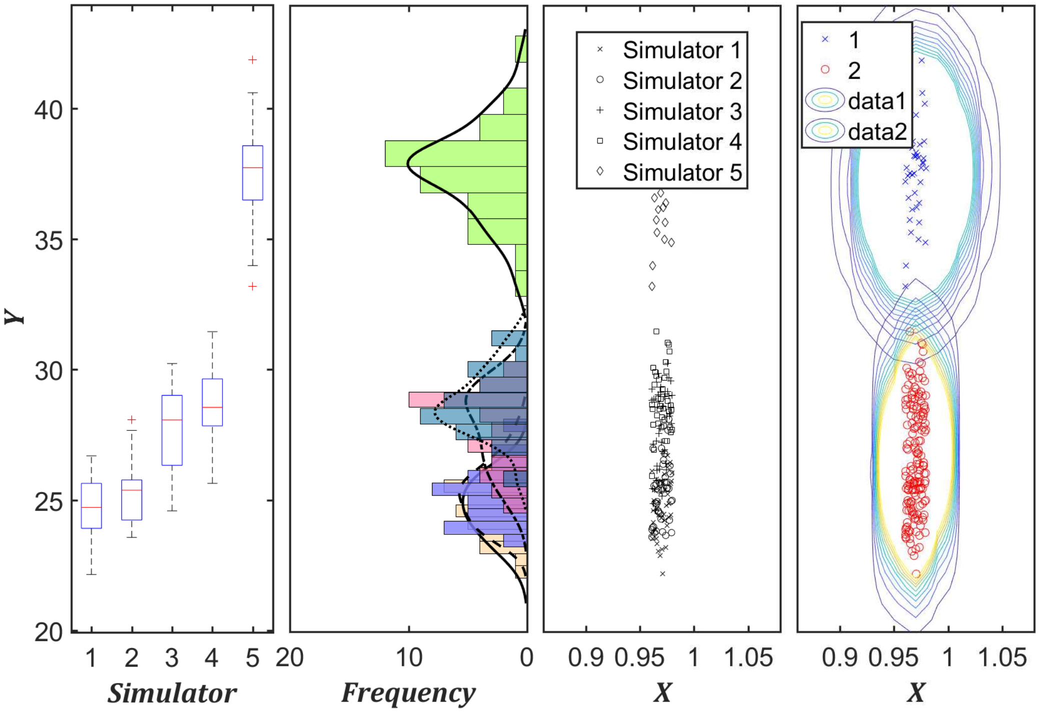
(a)

(b)

Figure 11: Case study 3: Percent reduction of the 95% confidence interval as a function of x over 50 repetitions (a) for the original bias of simulators 3 and 4, and (b) when the bias of simulators 3 and 4 is reduced by 0.75 points.



(a)



(b)

Figure 12: Case study 3: Clustering of response for $x \in [0.96, 0.98]$ for (a) original bias of simulators 3 and 4, and (b) when the bias of simulators 3 and 4 is reduced by 0.75 points.

5.4. *Case study 4: Majority of simulators are imperfect*

In this case study, four out five simulators (simulators 2-5) are in disagreement with the target response, and are defined as follows (see Figure 13a).

$$
\begin{aligned}
&y_1\left(x\right) = y_{target}\left(x\right); P\left(\mathcal{Y}_1 \mid X\right) \sim \mathcal{N}\left(y_1\left(x\right), COV = 0.05\right) \\
&y_2\left(x\right) = y_{target}\left(x\right) + 1; P\left(\mathcal{Y}_2 \mid X\right) \sim \mathcal{N}\left(y_2\left(x\right), COV = 0.05\right) \\
&y_3(x) = A_3 y_{target}(x^2) + B_3\left(x^3 - 0.5\right) + C_3; P\left(\mathcal{Y}_3 \mid X\right) \sim \mathcal{N}\left(y_3\left(x\right), COV = 0.05\right) \\
&y_4(x) = A_4 y_{target}(x) + B_4\left(x - 0.5\right) + C_4; P\left(\mathcal{Y}_4 \mid X\right) \sim \mathcal{N}\left(y_4\left(x\right), COV = 0.05\right) \\
&y_5(x) = A_5 y_{target}\left(\sqrt{x} + B_5\right) + C_5; P\left(\mathcal{Y}_5 \mid X\right) \sim \mathcal{N}\left(y_5\left(x\right), COV = 0.05\right)
\end{aligned}
\tag{19}
$$

The constants are set to $\{A_3 = 0.5, B_3 = 8, C_3 = 2\}$, $\{A_4 = 0.5, B_4 = 10, C_4 = 5\}$, and $\{A_5 = 0.25, B_5 = 0.5, C_5 = 12\}$. The choice of the constants is motivated by the introduction of a more diverse set of simulators exhibiting various (mixed) types of disagreements, namely, on the nature of the physical response (linear versus non-linear), on the expected value of the response (systematic versus random bias), and finally on the value and location of specific maximum/minimum peak responses. The diversity amongst the synthetic stochastic simulators in this case study are described in Table 3.

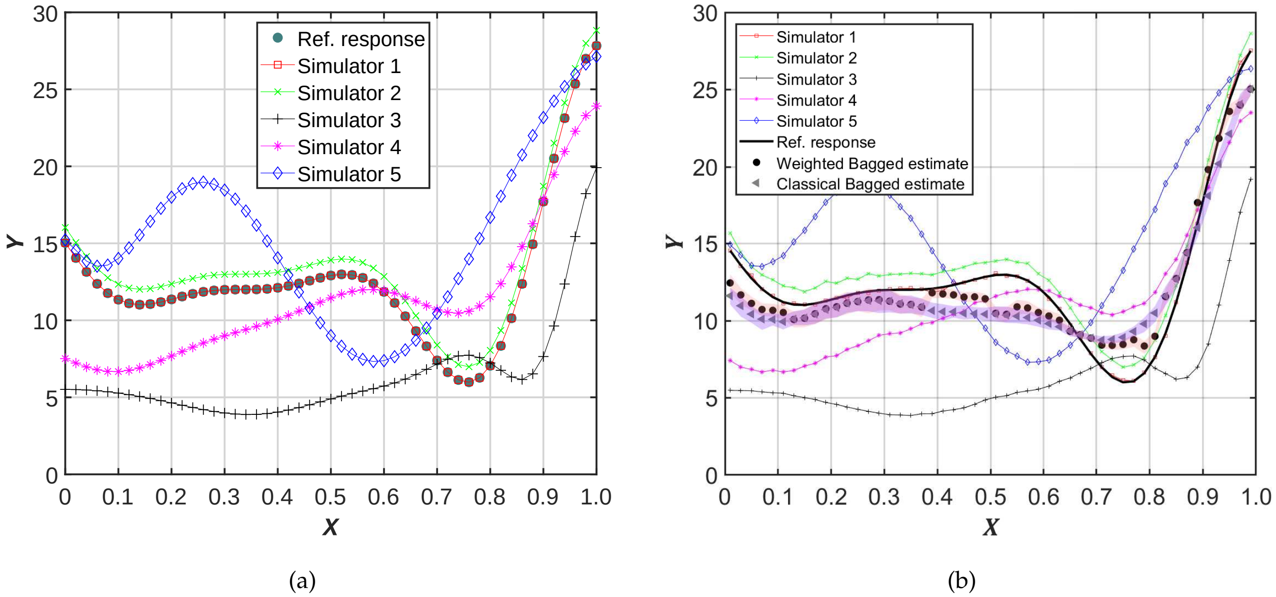
Table 3: Diversity amongst the stochastic simulators.

1	Diversity in the nature of the physical response	Simulator 5 is non-linear for $x \in [0.1, 0.6]$, while Simulator 4 exhibits a steep linear response over the same range, and shallower drop for $x \in [0.7, 0.8]$ compared to target response
2	Diversity in the value of the expected response	Simulator 2 systematically over-estimates the true response. Simulator 3 systematically under-estimate the true response. Simulator 4 under-estimates the true response for $X \in [0, 0.6]$ and over-estimates the true response for $X \in]0.6, 0.9]$
3	Diversity in the expected value of the minimum peak response	In contrast to simulator 4, simulators 1, 2 & 3 tend to correctly agree on the expected value of the minimum peak response
4	Diversity in the expected location of the minimum peak response	The true minimum peak response occurs for $X \in [0.7, 0.8]$. Simulator 3 predicts the peak to occur at $X \in [0.8, 0.9]$, while simulator 5 predicts the peak to occur at $X \in [0.5, 0.6]$

The cluster-weighted Bagged estimator is shown in Figure 13b yielding an improved predictor of the target response compared to conventional Bagging, based on 2000 observational samples from each simulator, and $\Delta x = 0.02$. We report that the generalization error of the cluster-weighted Bagged estimator is reduced by 39% while the 95% confidence interval is reduced by 7% averaged across the binned input space x when compared to that of the classical Bagged estimator.

Next, we study the performance of the cluster-weighted Bagged estimator when varying the simulators' output coefficient of variation for a fixed input bin size $\Delta x = 0.02$. The hypothesis here is that large output variability amongst simulators would challenge the VGBM clustering algorithm to discriminate unique clusters. The results shown in Figure 14 point out to a deterioration in the average reduction in confidence interval and generalization error of the cluster-weighted Bagged estimator compared to classical Bagging with increasing output coefficient of variation. The reduction in confidence interval and generalization error tends to zero for $COV \geq 10\%$ when 500-2000 observations are sampled for each simulator. However, it turns out that increasing the number of sampled observations to 5000 for each simulator help improve the performance because when more points are nearer to the Gaussians distribution center, the more likely it belongs to that specific cluster (component of the mixture). This should make intuitive sense, since with a Gaussian distribution we are assuming that most of the data lies closer to the center of the cluster. It is not clear however if this behaviour of the Gaussian mixture clustering holds true when dealing with skewed and extreme value distributions of the simulators outputs, something that needs to be further investigated in the future.

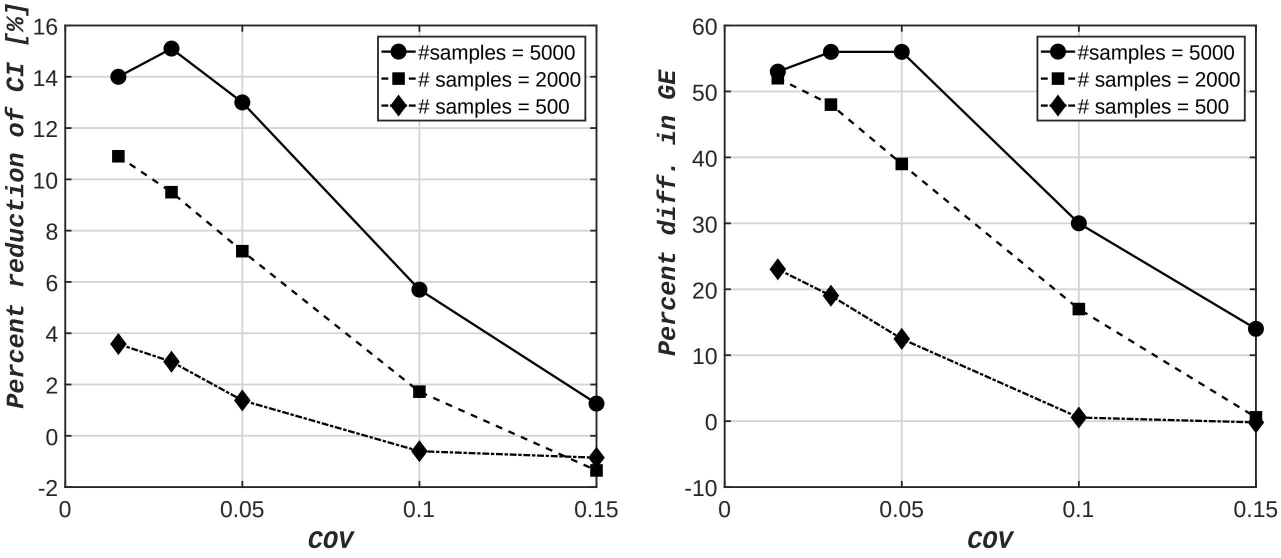
Finally, we evaluate the effect of the input bin size Δx on the the performance of the cluster-weighted Bagged estimator when 500, 2000 and 5000 observations are sampled for each simulator. When the target response is highly non-linear, evaluating the cluster-weighted Bagged estimator (or any estimator for that matter) over a large input bin size will tend to miss important aspects of the response. The results shown in Figure 15 point to the presence of an input bin size Δx that maximizes the performance of the cluster-weighted Bagged estimator in terms of reduction of the estimator confidence interval as well as the generalization error, which in this very case study seems to lie in the range $\Delta x \in [0.03, 0.06]$. Interestingly though, increasing the observational sample size per simulator from 2000 to 5000 deteriorates the performance of the cluster-weighted Bagged estimator for $\Delta x > 0.05$. This is attributed to the increased likelihood of additional response clusters being elucidated "horizontally" over the width of Δx, and can be seen as the trade-off between the bin size and the number of observational samples per simulator.



(a)

(b)

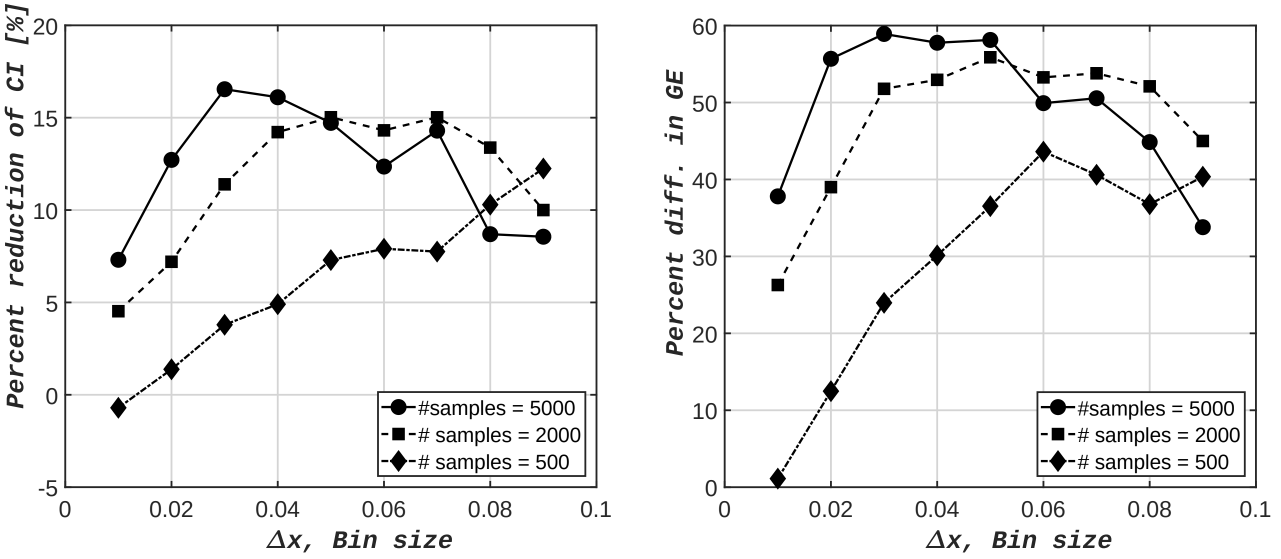
Figure 13: Case study 4: (a) The expected values of the target and the five simulators' responses. (b) The cluster-weighted Bagged estimator.



(a)

(b)

Figure 14: Case study 4: Effect of the output coefficient of variation on (a) percent reduction of 95% confidence interval (CI), and (b) percent reduction in generalization error (GE), across 50 repeats of the cluster-weighted bagged estimator (100 bootstraps), compared to classical bagged estimator for 500, 2000 and 5000 samples from each simulator.



(a)

(b)

Figure 15: Case study 4: Effect of the bin size on (a) percent reduction of 95% confidence interval (CI), and (b) percent reduction in generalization error (GE), across 50 repeats of the cluster-weighted bagged estimator (100 bootstraps), compared to classical bagged estimator for 500, 2000 and 5000 samples from each simulator.

5.5. Case study 5: Several measurements replications become available

The local weights are derived purely based on an unsupervised learning approach, where a-priori information about the performance of any given simulator in a given point in the input space is not available or is highly uncertain. In other words, the derived local weights may be inadequate, since the method fails to recognize certain simulators are more fitting than others in certain regions of the input space. The classical approach in such a case is to update the weights via measurements of the quantity of interest, traditionally from one machine or one structure. Integrating measurements of the quantity of interest from multiple replications of the same machine or structure is rarely done. However, in the wind energy infrastructure domain, it is customary to build multiple full scale prototype models (replications) and their sub-structures (e.g., towers, blades, gearbox, etc.) tend to be tested and measured substantially in various terrains and operational environments. Even though measurements assimilation is not the focus of this paper, we advance the method, exploiting measurement data to update and condition the weights. Our insight is that the measurements are just another set of physical (real, non-numerical) "simulators" output, which are added to the bucket of available output from the numerical simulators. The unsupervised ensemble cluster-weighted Bagged framework assimilates the measurements replications without explicitly assuming that the measurements are the "ground truth", and as such the simulators are not assigned individual weights for their likelihood being the "true system process" predictors, but are assigned weights for being part of clusters. Therefore, the measurements (be it one or more replications of the same system) become yet another physical simulator(s) in the mix with other numerical simulators, and algorithm 1 is simply re-run without any additional mathematical manipulations.

Here we re-use the same analytical simulators as in the previous case study, and assimilate 1, 3, 5 and 10 measurements replications. Each measurement replication are samples from the target (true) response, but we also include an error term ϵ, which is normally distributed with zero-mean and 0.25 standard deviation: $y_{meas}(x) = y_{target}(x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, 0.25)$. 2000 observational samples are available from each simulator and measurement replication, and $\Delta x = 0.02$. We calculate the generalization error and the 95% confidence interval of the estimator as a function of the coefficient of variation of the output. According to Figure 16, for lower coefficient of variations (i.e., $COV \leq 0.05$) we observe that we can achieve a 2-order of magnitude reduction in the generalization error, when upto 10 measurement replications are assimilated via the ensemble cluster-weighted Bagged framework. For $COV \geq 0.10$ the drop in the generalization error is only 1-order of magnitude. Similarly, Figure 17 shows that the 95% confidence interval drops as more and more measurement replications are included to update the cluster-weighted Bagged estimator, which translates into a better precision of the estimator for a given confidence level. Interestingly, for lower coefficient of variations (i.e., $COV \leq 0.05$) when 10 measurements replications are assimilated in the algorithm, the 95% confidence interval exhibits little fluctuations over the whole input range x.

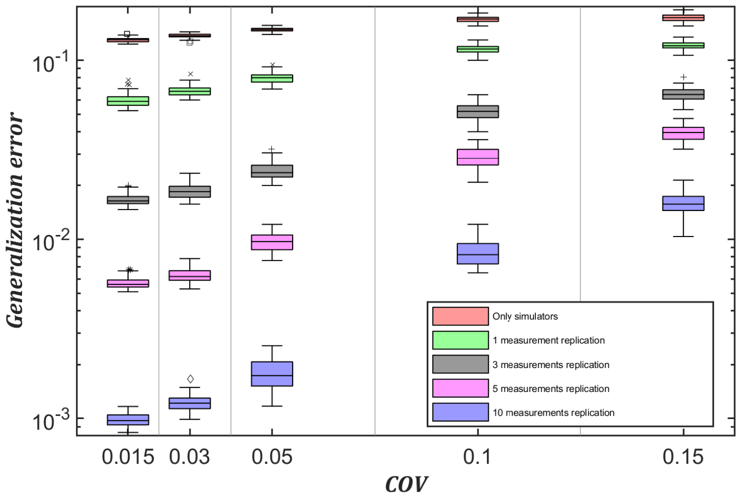
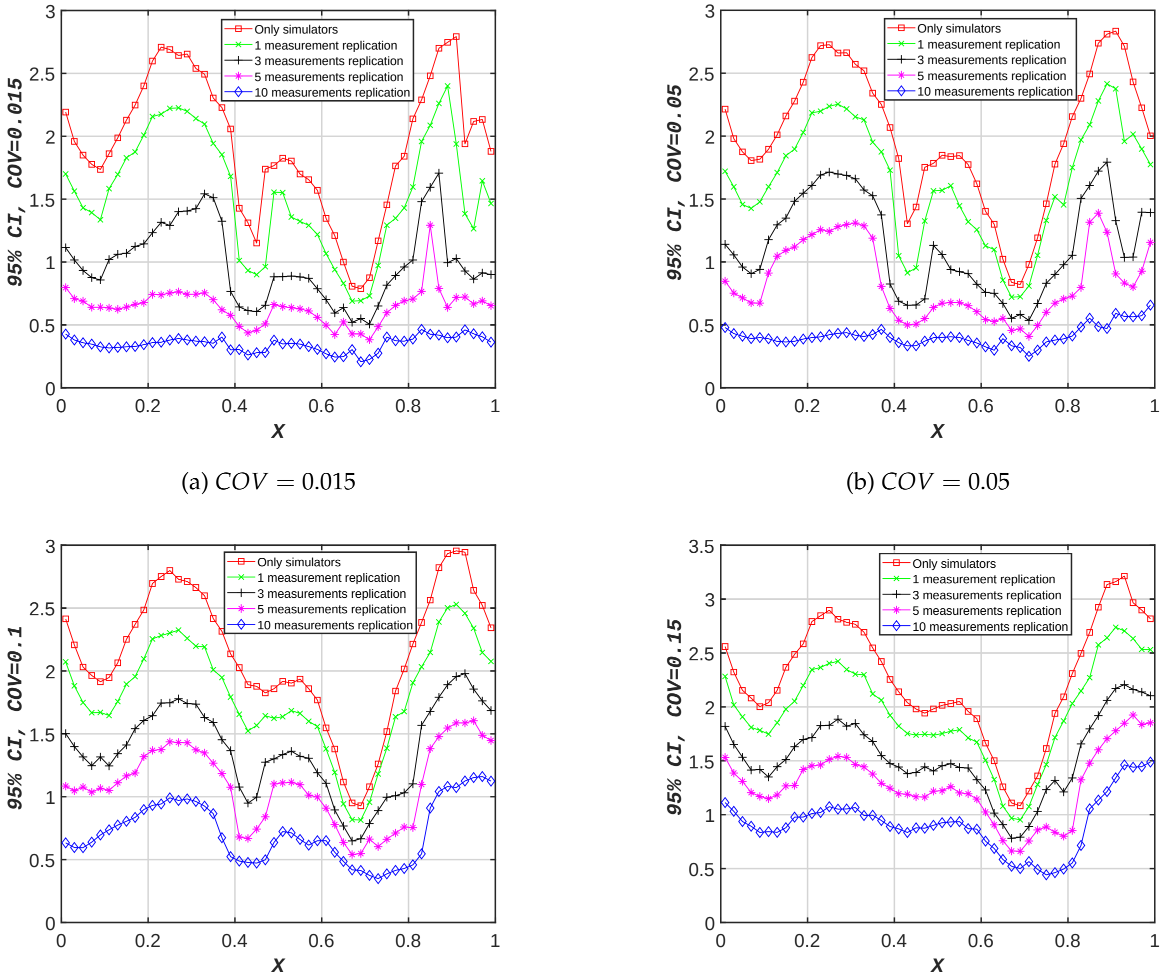


Figure 16: Generalization error for the measurements replications assimilations as a function of the simulators' coefficient of variation.



(a) $COV = 0.015$

(b) $COV = 0.05$

(c) $COV = 0.10$

(d) $COV = 0.15$

Figure 17: 95% Confidence Interval for the measurements replications assimilations as a function of input x.

6. Comparison to other methods

BMA, Pseudo-BMA, Pseudo-BMA-BB, Stacking, BMC, VAE

7. Engineering Application

7.1. Wind loads and FEM models

We demonstrate the method by evaluating the Fatigue Damage Equivalent Load (DEL) on a 30 m long wind turbine blade, using 10 Finite Element based (FEM) simulators.

Wind loads: The turbulent wind field (Figure 18a) is generated using spatially correlated wind inflow time series along the span of the blade, based on an exponential coherence model and the Kaimal turbulence auto-spectrum to account for the spatial correlation structure of the longitudinal velocity component (for further details please refer to Annex B in [62]). Once the wind field is generated, the time varying normal F_N and tangential F_T aerodynamic forces along the span of the blade (Figure 18b and 19) are computed using a quasi-static Blade Element Momentum model (BEM) [63].

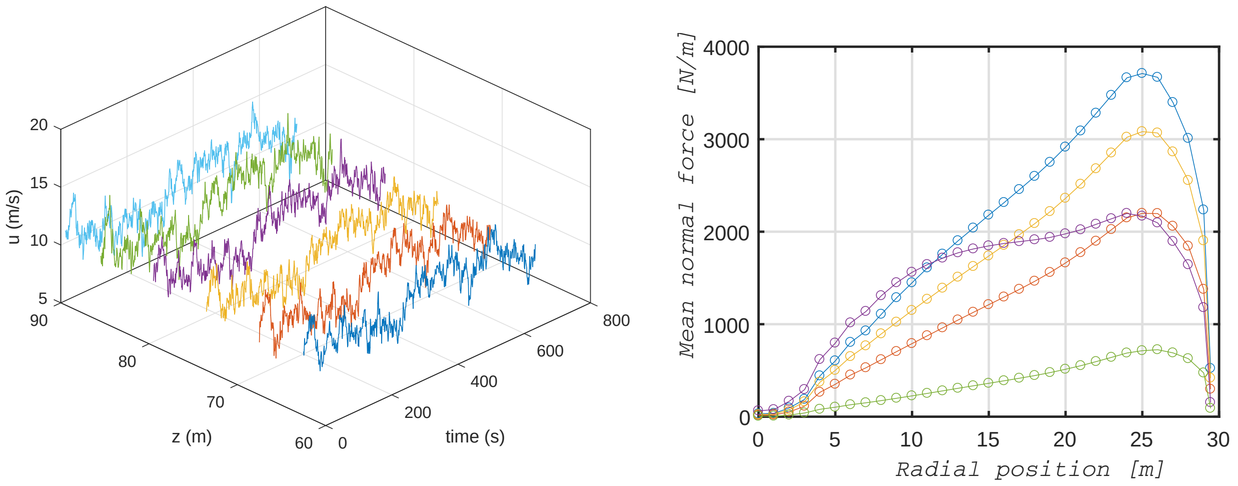
Finite element models: In order to simplify this application, the complex structure of the blade is converted into an equivalent tapered clamped-free beam. With reference to Figure 19 we first define the following coordinate systems and centres; (X, Y) is the reference coordinate system of the beam cross-section, (EI_1, EI_2) is the principal axes, X_{aero} is the distance from leading edge to the aerodynamic center, X_s is the distance from the leading edge to the shear center, X_e is the distance from the leading edge to the elastic center, h is the height of the beam cross-section, and b is the width of the beam cross-section. Ten finite element simulators of the beam were developed to compute the time varying blade root in-plane bending moment M_X from the distributed normal F_N and tangential F_T aerodynamic forces along the span of the blade.

The constitutive parameters of the FEM simulators are shown in Table 4, where the applied loads orientation "In-plane projection" means that the resultant of the normal and tangential forces at a given cross-section is projected onto the $YZ - plane$, and "According to γ" means that the normal and tangential forces at a given cross-section are oriented according to the angle γ, which is computed in the Blade Element Momentum model, and is dependent on the wind inflow angle, the blade pitch and twist angles. The input are 100 stochastic times series F_N and F_T, each 600 seconds long with a time step of 0.01 s, applied at the cross-sectional aerodynamic center X_{aero} along the span of the tapered beam. A torsional moment is thus induced because the point of application of the load at a given cross-section is offset from the shear centre.

Fatigue: The short-term fatigue damage equivalent loads (DEL) at the blade clamped root end are calculated on the basis of the M_X output times series, which, for a given mean wind speed, is determined by:

$$DEL = \left(\frac{1}{N_{eq}} \sum_i n_i \left(\Delta M_i \right)^m \right)^{1/m} \tag{20}$$

where n_i is the number of load cycles with range ΔM_i in a time series, i is the fatigue cycle index, and N_{eq} is the equivalent number of load cycles, typically 10^7 cycles. The Whöler exponent m is set to $m = 10$ for a composite structure.



(a) (b)

Figure 18: (a) An example of the free stream turbulent wind speed at various radial positions along the span of the blade. (b) An example of the mean value of the normal force F_N along the span of the blade for different mean wind speeds.

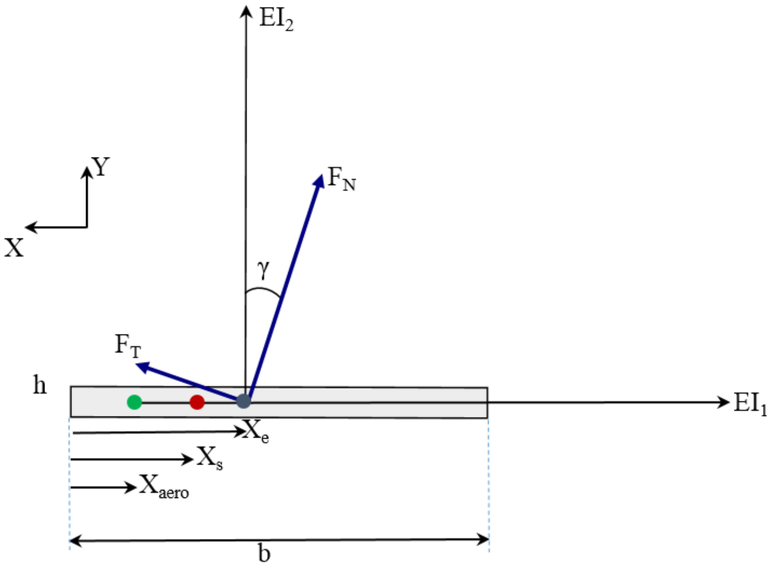


Figure 19: Coordinate system of the beam cross-section.

7.2. *Results and discussions*

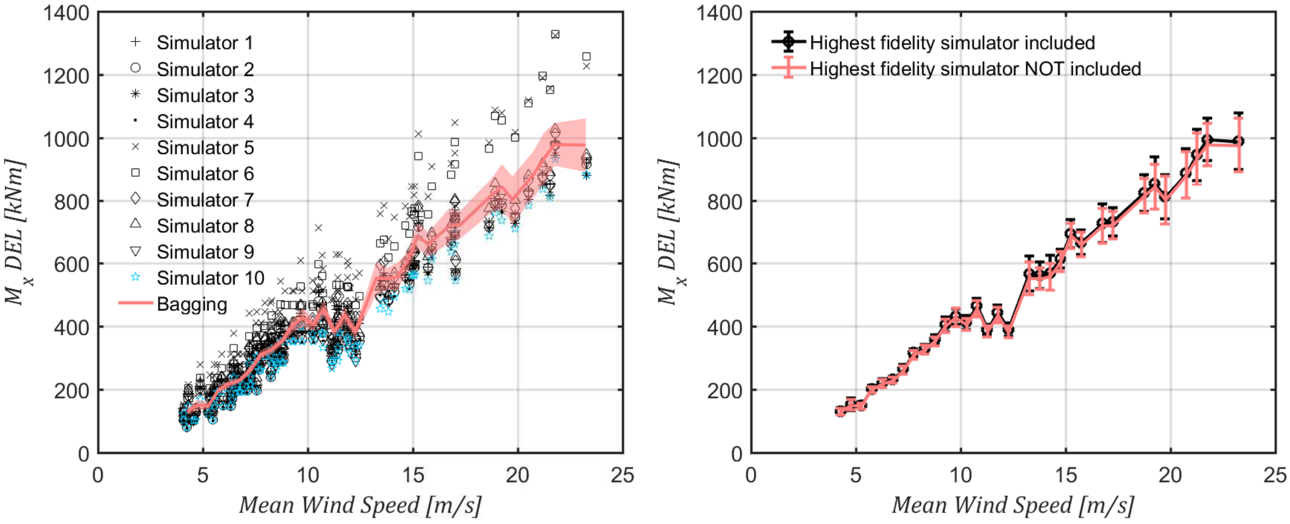
The computed DEL for each of the simulators and the ensemble Cluster-Weighted Bagged expected DEL (with 95% confidence interval) are plotted and compared as a function of wind speed in Figure 20(a). We observe that the dispersion of the simulators predictions is not significant because we are examining damage equivalent fatigue loads, which by itself is an aggregate quantity and under benign input wind loading conditions the simulators will exhibit similar predictions due to the linear nature of the blade response, except for the predictions from simulators 5 and 6 which start to drift away at wind speeds above 15m/s.

Table 4: Parameters of the 10 FEM simulators of the clamper-free beam.

Simulator	1	2	3	4	5	6	7	8	9	10
Dimensions	1D	1D	3D	3D	2D	2D	3D	3D	3D	3D
Type of beam	Euler-Bernoulli	Euler-Bernoulli	Euler-Bernoulli	Timoshenko	2D Elas.	2D Elas.	3D Elas.	3D Elas.	3D Elas.	3D Elas.
Elements	Beam	Beam	Beam	Beam	Plane-Stress	Plane-Stress	Solid	Solid	Solid	Solid
Shape function	Linear	Linear	Linear	Quadratic	Linear	Quadratic	Linear	Quadratic	Quadratic	Quadratic
Modes	2	4	4	8	4	8	4	8	10	12
Elements	16	48	48	48	16	48	2x2x96	4x4x192	8x8x384	16x16x768
Load nodes	6	14	10	14	6	14	6	10	14	28
Applied loads orientation	In-plane projection	In-plane projection	According to γ	According to γ	In-plane projection	In-plane projection	In-plane projection	According to γ	According to γ	According to γ
Torsion DOF	off	off	on	on	off	off	off	off	on	on
Stiffness Matrix	Full	Full	Full	Full	Full	Full	Full	Full	Full	Full

A natural outcome is that the ensemble Cluster-Weighted Bagged predictor displays a small variance compared to the original data set, which is a very important target because it reflects a lower variability -lower model uncertainty- in the prediction of fatigue on a structural component. The outcome of clustering is a map of probabilities, which effectively is the joint probability $P\left(\mathcal{S}_i, C_j \mid \Delta x\right)$ of a cluster C_j and a simulator S_i. Introducing these probabilities when bootstrapping the data in the Bagging algorithm has the effect of pulling the ensemble aggregated predictor towards the clusters with higher densities, which is indeed what happens at wind speeds above 15 m/s.

We experiment with removing the higher fidelity simulator (simulator 10) from the mix and re-run the ensemble Cluster-Weighted Bagging algorithm to compute the expected the damage equivalent loads. According to Figure 20(b), we observe that the predictor remains little changed, which is computationally attractive since the FEM simulator 10 takes an order of magnitude more time to run for little change in the ensemble aggregate predictive ability.



(a)

(b)

Figure 20: (a) Comparison of damage equivalent load of the blade root in-plane bending moment M_X from 10 FEM simulators, and the ensemble Cluster-Weighted Bagged predictor with the 95% confidence interval. (b) Comparison of the ensemble Cluster-Weighted Bagged predictor when the highest fidelity simulator 10 is not included in the mix of simulators.

8. Conclusion and future perspective

In this contribution we proposed an ensemble learning framework based on unsupervised variational Bayesian Gaussian mixture clustering and local weighted bootstrap aggregating (Bagging) the stochastic output of a quantity of interest from multiple distinct simulators, in the absence of any validatory measurements data. Clustering served the purpose of deriving the probability map (weighting) of the simulators' output. Clustering is carried out on the stochastic output corresponding to the binned input space. Local cluster-weighted Bootstrap Aggregation, serves the purpose of weighted aggregation of the clustered ensemble of outputs from the individual simulators. Furthermore, even though assimilation of measurements is not the focus of this paper, we advance the method, demonstrating how measurement data may be exploited in order to update and condition the weights. Our insight is that the measurements (be it one or more replications of the same system) may be considered as yet another physical simulator(s) in the mix with other numerical simulators, with the proposed algorithm simply re-run without any additional mathematical manipulations. We illustrated the framework with simple analytical examples and compared its effectiveness to that of classical Bagging. We demonstrated tangible performance gains in terms of reduction in the generalization error and the 95% confidence interval of the estimator. This is crucial because a reduction in the generalization error translates into a better accuracy of the estimator, while a reduction in the confidence interval translates into a better precision of the estimator for a given confidence level. Finally, we show how the performance gains vary depending on the input bin size, observational sample size per simulator, simulations output dispersion and level of agreement amongst the simulators (diversity). Regardless, the results point to the need for practitioners to consider this as a useful framework, when output from multiple simulators is available, particularly when it is hard to distinguish a best model out of an ensemble.

That said, several issues remain to be tackled in future work. A limiting factor of the presented scheme stems from the fact that we only compare the unsupervised local cluster-weighted bootstrap aggregation to classical Bagging, and not to other state-of-the-art model aggregation/combination/averaging methods. This limitation is to be addressed formally in future work. Several interesting extensions to this research are identified. The first one is to expand the method to the domain of structural and system reliability, to fundamentally compute a probability of failure given a limit state function and the local cluster-weighted Bagged output near a design point in the input space. Second, we need to investigate if the potential gains from using the unsupervised local cluster-weighted Bagging algorithm holds in high dimensional engineering problems (e.g. 10 input random variables or more), especially in view of the challenges of binning data in a high dimensional input space. Third, in case the stochastic quantity of interest follows an extreme value distribution then the Variational Gaussian Mixture clustering may not be adequate. It is thus of interest to deploy Non-Gaussian based clustering, such as model-based clustering with non-normal mixture distributions or clustering with variational autoencoders for multidimensional clustering. Fourth, potential stability issues relating to the local clustering step, ought to be investigated [64, 65, 66, 67, 68, 69, 70]. In this contribution, a simple approach is adopted, where we repeatedly draw sub-samples from the population and apply the Variational Gaussian Mixture clustering algorithm, and subsequently perform a majority vote to select the most commonly occurring cluster structure. Hence, clusters are meaningful if they manifest in multiple independent samples from the same population. However, there is a trade-off, which needs to be carefully treated. If we overly disturb the original dataset (for example, the subsample is too small), then we might destroy the structure we want to discover by clustering. In case of minor alterations of the original dataset the clustering algorithm will always render the same results, resulting in trivial stability.

Finally, further research is needed in order to address the case when the correct simulator output is clustered separately from the larger density clusters for certain combinations of input $\boldsymbol{x}^{(i)}$ in the d-dimensional input space $\mathcal{D}_{\boldsymbol{x}}$. In other words, the derived weights may be inadequate because the method fails to recognize certain simulators are more fitting than others in certain regions of the input space.

Acknowledgements

The authors acknowledge the support of the European Research Council via the ERC Starting Grant WINDMIL (ERC-2015-StG #679843) on the topic of Smart Monitoring, Inspection and Life-Cycle Assessment of Wind Turbines. The authors would also like to acknowledge the support the ERC Proof-of-Concept Grant ERC-2018-PoC WINDMIL RT-DT (funded under H2020-EU.1.1., grant agreement ID 825833).

References

[1] V. Moutoussamy, S. Nanty, B. Pauwels, Emulators for stochastic simulation codes, in: ESAIM: Proceedings and Surveys, EDP Sciences, volume 48, 2015, pp. 116–155.

[2] M. Buhl, A. Wright, K. Pierce, Wind turbine design codes: A comparison of the structural response, Technical Report NREL/CP-500-27470, National Renewable Energy Laboratory, 2000.

[3] J. G. Schepers, J. Heijdra, D. Foussekis, S. Oye, R. R. Smith et. al, Verification of European Wind Turbine Design Codes , VEWTDC ; Final report, Technical Report ECN-C-01-055, ECN, 2002.

[4] M. Buhl, A. Manjock, A Comparison of Wind Turbine Aeroelastic Codes Used for Certification, Technical Report NREL/CP-500-39113, National Renewable Energy Laboratory, 2006.

[5] J. Jonkman, S. Butterfield, T. Camp, J. Nichols, J. Azcona, A. Martinez, Offshore Code Comparison Collaboration within IEA Wind Annex XXIII : Phase II Results Regarding Monopile Foundation Modeling, Technical Report NREL/CP-500-42471, National Renewable Energy Laboratory, 2008.

[6] J. Jonkman, W. Musial, Offshore code comparison collaboration (OC3) for IEA task 23 offshore wind technology and deployment, Technical Report NREL/TP-5000-48191, National Renewable Energy Laboratory, 2010.

[7] W. Popko, F. Vorpahl, A. Zuga, M. Kohlmeier, J. Jonkman, et al., Offshore code comparison collaboration continuation (OC4), phase i - results of coupled siumulations of an offshore wind turbine with jacket support structure, Journal of Ocean and Wind Energy 1 (2014) 1–11.

[8] R. Ranjan, T. Gneiting, Combining probability forecasts, J. Royal Stat. Soc., Series B 72 (2010) 71–91.

[9] J. Merrick, Aggregation of forecasts from mutiple simulation models, in: Proceedings of the 2013 Winter Simulation Conference, 2013.

[10] J. A. Hoeting, D. M. Madigan, A. E. Raftery, C. T. Volinsky, Bayesian model averaging: A tutorial, Statistical Science 14 (1999) 382–401.

[11] Q. Yu, S. N. MacEachern, M. Peruggia, Clustered bayesian model averaging, Bayesian Analysis 8 (2013) 883–908.

[12] I. Abdallah, K. Tatsis, E. Chatzi, Fatigue assessment of a wind turbine blade when output from multiple aero-elastic simulators are available, in: Proceedia Engineering, X International Conference on Structural Dynamics, EURODYN 2017, Rome, Italy, volume 199, 2017, pp. 3170–3175.

[13] I. Abdallah, , E. Chatzi, An ensemble learning approach to aggregate the output from multiple simulators and measurements, in: The 19th working conference of the IFIP Working Group 7.5 on Reliability and Optimization of Structural Systems (IFIP 2018), 2018.

[14] R. Winkler, Combining probability distribution from dependent information sources, Management Science 27 (1981) 479–488.

[15] R. Clemen, R. Winkler, Combining Probability Distributions From Experts in Risk Analysis, Risk Analysis 19 (1999) 187–203.

[16] D. Allaire, K. Willcox, O. Toupet, A Bayesian-Based Approach to Multifidelity Multidisciplinary Design Optimization, in: 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference, Fort Worth, Texas, 2010. doi:AIAA2010-9183.

[17] D. Allaire, K. Willcox, A mathematical and computational framework for multifidelity design and analysis with computer models, International Journal for Uncertainty Quantification 4 (2014) 1–20.

[18] M. Jouini, R. Clemen, Copula models for aggregating expert opinions, Operations Research 44 (1995) 444–458.

[19] A. Mosleh, G. Apostolakis, The assessment of probability distributions from expert opinions with an application to seismic fragility curves, Risk Analysis 6 (1986) 447–461.

[20] E. Zio, G. Apostolakis, Two methods for the structured assessment of model uncertainty by experts in performance assessments of radioactive waste repositories, Reliability Engineering & System Safety 54 (1996) 225–241.

[21] M. Riley, Quantification of model-form, predictive, and parametric uncertainties in simulation-based design, Doctoral thesis (2007).

[22] I. Park, H. Amarchinta, R. Grandhi, A Bayesian approach for quantification of model uncertainty, Reliability Engineering & System Safety 95 (2010) 777–785.

[23] M. Riley, R. Grandhi, Quantification of model-form and predictive uncertainty for multi-physics simulation, Computers & Structures 89 (2011) 1206–1213.

[24] J. Gibbons, G. Cox, A. Wood, J. Craigon, S. Ramsden, D. Tarsitano, N. Crout, Applying Bayesian Model Averaging to mechanistic models: An example and comparison of methods, Environmental Modelling & Software 23 (2008) 973–985.

[25] K. Alvin, W. Oberkamprt, K. Diegert, B. Rutherford, Uncertainty quantification in computational structural dynamics: a new paradigm for model validation, in: 16th International Modal Analysis Conference, 1998, pp. 1191–1198.

[26] S. Chib, I. Jeliakov, Marginal likelihood from the Metropolis-Hastings output, Journal of the American Statistical Association 96 (2001) 270–281.

[27] C. Bos, A Comparison of Marginal Likelihood Computation Methods, Technical Report No. 02-084/4, Tinbergen Institute, Discussion Paper, 2002.

[28] N. Friel, J. Wyse, Estimating the model evidence: a review, arXiv:1111.1957 (2011).

[29] T. Fragoso, T. Bertoli, F. Louzada, Bayesian model averaging: A systematic review and conceptual classification, International Statistical Review (2017).

[30] D. Madigan, A. Raftery, Model selection and accounting for model uncertainty in graphical models using Occam's window, Journal of the American Statistical Association 89 (1994) 1534–1546.

[31] J. Rings, J. Vrugt, G. Schoups, J. Huisman, H. Vereecken, Bayesian model averaging using particle filtering and Gaussian mixture modeling : Theory, concepts, and simulation experiments, Water Resources Resarch 48 (2012).

[32] D. Liu, S. Wang, C. Zhang, M. Tomovic, Bayesian model averaging based reliability analysis method for monotonic degradation dataset based on inverse gaussian process and gamma process, Reliability Engineering and System Safety 180 (2018) 25–38.

[33] K. Monteith, J. L. Carroll, K. Seppi, T. Martinez, Turning bayesian model averaging into bayesian model combination, in: 2011 International Joint Conference on Neural Networks, IEEE, San Jose, CA, USA, 2011, pp. 2657–2663.

[34] M. Carrasco, Probabilistic photometric redshifts in the era of petascale astronomy, Ph.D. thesis, University of Illinois at Urbana-Champaign, IL, USA, 2014.

[35] B. Peherstorfer, B. Kramer, K. Willcox, Combining multiple surrogate models to accelerate failure probability estimation with expensive high-fidelity models, Journal of Computational Physics 341 (2017) 61–75.

[36] D. Jimenez Rezende, S. Mohamed, Variational inference with normalizing flows, in: Proceedings of the 32nd International Conference on Machine Learning, Lille, France, volume 37, 2015.

[37] T. Kingma, D.P.and Salimans, R. Jozefowicz, X. Chen, I. Sutskever, M. Welling, Improved variational inference with inverse autoregressive flow, in: 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, 2016.

[38] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.

[39] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1997) 119–139.

[40] D. Wolpert, Stacked generlization, Neural Networks 5 (1992) 241–259.

[41] P. Bühlmann, Bagging, boosting and ensemble methods, in: J. Gentle, Y. Mori (Eds.), Handbook of comp. stat., Springer, 2012, pp. 985–1022.

[42] L. Rokach, O. Maimon, Clustering methods, Data Mining and Knowledge Discovery Handbook, Springer, 2005.

[43] D. Pham, A. Afify, Clustering techniques and their applications in engineering, Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 221 (2007).

[44] C. Bishop, Pattern recognition and machine learning, Springer, ISBN 978-0-387-31073-2, 2006.

[45] A. Dempster, N. Laird, D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Royal Stat. Soc. 39 (1977) 1–38.

[46] G. McLachlan, T. Krishnan, The EM algorithm and extensions, 2nd Edition, Wiley series in prob. and stat., ISBN 978-0-471-20170-0, 2008.

[47] K. Brodersen, J. Daunizeau, C. Mathys, J. Chumbley, J. Buhmann, K. Stephan, Variational bayesian mixed-effects inference for classification studies, NeuroImage 76 (2013) 345–361.

[48] J. Ormerod, M. P. Wand, Explaining variational approximations, in: Handbook of computational statistics, volume 64, Taylor & Francis, Ltd. on behalf of the American Statistical Association, 2010, pp. 140–153.

[49] S. Sun, A review of deterministic approximate inference techniques for bayesian machine learning, Neural Comput & Applic 23 (2013) 2039–2050.

[50] J. Nagel, B. Sudret, Spectral likelihood expansions for bayesian inference, Journal of Computational Physics 309 (2016) 267–294.

[51] C. Fox, S. Roberts, A tutorial on variational bayesian inference, Artif Intell Rev 38 (2012) 85–95.

[52] H. Attias, A variational bayesian framework for graphical models, in: Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99, MIT Press, Cambridge, MA, USA, 1999, pp. 209–215.

[53] W. Penny, Variational Bayes for d-dimensional Gaussian mixture models, Technical Report, University College London, 2001.

[54] W. Penny, S. Kiebel, K. Friston, Statistical Parametric Mapping: The analysis of functional brain images, Elsevier, ISBN 9780123725608, 2006.

[55] D. M. Blei, A. Kucukelbir, J. D. McAuliffe, Variational inference: A review for statisticians, arXiv:1601.00670 (2016).

[56] A. Corduneanu, C. Bishop, Variational bayesian model selection for mixture distributions, in: Proceedings Eighth International Conference on Artificial Intelligence and Statistics, Morgan Kaufmann, 2001, pp. 27–34.

[57] A. Teschendorff, Y. Wang, N. Barbosa-Morais, J. Brenton, C. Caldas, Bayesian mixture modelling framework for cluster analysis of gene-expression data, Bioinformatics 21 (2005) 3025–3033.

[58] C. McGrory, D. Titterington, Variational approximations in bayesian model selection for finite mixture distributions, Computational Statistics and Data Analysis 51 (2007) 5352–5367.

[59] P. Bühlmann, B. Yu, Analyzing bagging, Ann. Stat. 30 (2002) 927–961.

[60] A. Forrester, A. Sobester, A. Keane, Multi-fidelity optimization via surrogate modelling, Proceedings of the Royal Society (2007) 3251–3269.

[61] A. Marrel, B. Iooss, S. Da Veiga, M. Ribatet, Global sensitivity analysis of stochastic computer models with joint metamodels, Stat. Comput. 22 (2012) 833–847.

[62] Wind Turbines, Part 1 Design Requirements, Technical Report IEC 61400-1:2005(E), International Electrotechnical Commission, 2005.

[63] M. Hansen, Aerodynamics of wind turbines, James & James (Science Publishers) Ltd. pp. 152. ISBN 1902916069, 2001.

[64] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J. R. Statist Soc. 63 (2001) 411–423.

[65] A. Ben-Hur, A. Elisseff, I. Guyon, A stability based method for discovering structure in clustered data, in: Proceedings of the Pacific Sym02sium on Biocomputing, volume 7, 2002, pp. 6–17.

[66] R. Tibshirani, G. Walther, Cluster validation by prediction strength, J. Comp. and Graph. Statist 14 (2005) 511–528.

[67] J. Wang, Consistent selection of the number of clusters via cross-validation, Biometrika 97 (2010) 893–904.

[68] U. Von Luxburg, Clustering stability: An overview, arXiv:1007.1075v1 (2010).

[69] Y. Fang, J. Wang, Selection of the number of clusters via the bootstrap method, Computational Statistics and Data Analysis 56 (2012) 468–477.

[70] W. Fu, P. Perry, Estimating the number of clusters using cross-validation, arXiv:1702.02658v1 (2017).