

Journal Pre-proofs

Uncertainty Quantification in Extreme Learning Machine: Analytical Developments, Variance Estimates and Confidence Intervals

Fabian Guignard, Federico Amato, Mikhail Kanevski

PII: S0925-2312(21)00554-3
DOI: <https://doi.org/10.1016/j.neucom.2021.04.027>
Reference: NEUCOM 23719

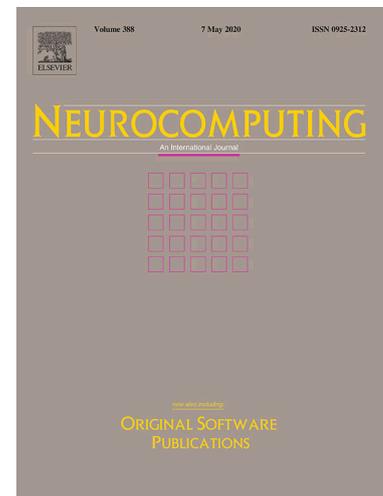
To appear in: *Neurocomputing*

Received Date: 3 November 2020
Accepted Date: 8 April 2021

Please cite this article as: F. Guignard, F. Amato, M. Kanevski, Uncertainty Quantification in Extreme Learning Machine: Analytical Developments, Variance Estimates and Confidence Intervals, *Neurocomputing* (2021), doi: <https://doi.org/10.1016/j.neucom.2021.04.027>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Published by Elsevier B.V.



Uncertainty Quantification in Extreme Learning Machine: Analytical Developments, Variance Estimates and Confidence Intervals

Fabian Guignard^{a,*}, Federico Amato^a, Mikhail Kanevski^a

^a*Institute of Earth Surface Dynamics, Faculty of Geosciences and Environment, University of Lausanne, Switzerland*

Abstract

Uncertainty quantification is crucial to assess prediction quality of a machine learning model. In the case of Extreme Learning Machines (ELM), most methods proposed in the literature make strong assumptions on the data, ignore the randomness of input weights or neglect the bias contribution in confidence interval estimations. This paper presents novel estimations that overcome these constraints and improve the understanding of ELM variability. Analytical derivations are provided under general assumptions, supporting the identification and the interpretation of the contribution of different variability sources. Under both homoskedasticity and heteroskedasticity, several variance estimates are proposed, investigated, and numerically tested, showing their effectiveness in replicating the expected variance behaviours. Finally, the feasibility of confidence intervals estimation is discussed by adopting a critical approach, hence raising the awareness of ELM users concerning some of their pitfalls. The paper is accompanied with a scikit-learn compatible Python library enabling efficient computation of all estimates discussed herein.

Keywords: Extreme Learning Machine, standard error, model variance, confidence interval, uncertainty quantification, regularization

1. Introduction

Statistical accuracy measures such as variance, standard error and Confidence Intervals (CI) are crucial to assess the quality of a prediction. Model uncertainty quantification is needed to build CI and has a direct impact on the prediction interval, especially when dealing with small datasets [1]. Uncertainty quantities for Feed-forward Neural Networks (FNN) solving regression tasks can be obtained by means of different methods[2, 3]. Here these quantities are investigated in relationship with the use of the Extreme Learning Machine (ELM)

*Corresponding author

Email address: `fabian.guignard@unil.ch` (Fabian Guignard)

model [4]. ELM is a single-layer FNN with random input weights and biases, therefore allowing the optimization of the output weights through the Least Squares (LS) procedure. One can think about ELM as a projection of inputs in a random feature space where a Multiple Linear Regression (MLR) with a null intercept is performed.

Three main uncertainty sources can be distinguished [5]. A first one comes from the data, and in particular from sampling variation and unexplained fluctuations, or noise. A second uncertainty source is related to the estimation of the model parameters, which in the case of an FNN correspond to the weights and biases [3]. In ELM, input weights and biases are randomly chosen, which clearly generates uncertainty. Moreover, despite being optimized through a procedure with a unique solution, the estimation of the output weights depends on the random input weights and on the data, which therefore induces additional fluctuations. Finally, a third type of uncertainty source is due to the model structure. This source, generally referred at as structural uncertainty, is not considered in this paper.

A number of methods were proposed to obtain confidence or prediction intervals with ELM. A Bayesian formulation was introduced to integrate prior knowledge and produce directly CI [6, 7]. In the frequentist paradigm, bootstrap methods were investigated in the context of time series [8]. Akusok et al. proposed a method to estimate prediction intervals using a covariance matrix estimate coming from MLR [9].

Most of these methods make Gaussian assumption on the output distribution or do not consider the bias in interval estimation, which may cause misleading conclusions. Moreover, resampling methods lead to important computational burden when the number of data is high. Finally, it is often argued that randomness of the input weights and biases is supposed to be negligible providing the training set large enough. However, it is not always clear how many data is needed in practice. Indeed, while stochastic input layer initialization can have weak impact in low dimension, it is still unclear what could happen when number of features or/and neurons are large. Because of the curse of dimensionality, the random drawing of the input weights and biases could have a higher impact than suspected. To further investigate such impact, the development of ELM variance estimation methods taking into account its stochastic nature is therefore extremely relevant. Additionally, ELM is also used efficiently with small training dataset [10] — in which case precise variance estimate is crucial — where the randomness of the input weights and biases should not be ignored.

The contribution of this paper is threefold. First, analytical development are proposed to derive ELM variance taking into account also the contribution induced by the random input weights and biases. This is done without any other assumptions on the noise distribution than the facts that it is centered and have a finite variance. In particular, the presented theoretical results hold for dependant and non-identically distributed data. Second, homoskedastic and heteroskedastic variance estimates are provided, and some of their properties are investigated. While it may be argued that the homoskedastic case could be unrealistic, its study is of great interest as it provides an insightful propaedeutic

55 value and develops the intuition for more advanced situations. Moreover, in case of applications with a small number of data, homoskedastic assumption may yield to better results. Third, the paper proposes empirical bases to move towards CI estimations, which include the variability induced by the random input weights and biases. Their discussion will also raise the awareness of ELM users about some pitfalls of confidence interval estimations. Overall, the results presented in this paper are expected to clarify the impact of input weights variability and noise, hence increasing the understanding of ELM variability.

60 The remainder of the paper is organized as follows. Starting from general assumption on the noise covariance matrix, probabilistic formulas are derived for predicted output variance knowing the training input for single and ensemble of (regularized) ELM in section 2. Based on these formulas, section 3 provides variance estimates when noise is independent with constant variance (homoskedastic case) and non-constant variance (heteroskedastic case), for which a Python implementation is available on GitHub, see the software availability at the end of the paper. The effectiveness of the proposed estimates is demonstrated through numerical experiments in section 4, where estimation of CI is also discussed. 70 Finally, section 5 concludes the paper.

2. Analytical developments

This section begins by recalling ELM theory and fixing notations. Subsequently, the bias and variance for a single ELM are derived. The results are then generalized to ELM ensemble. Finally, correlation between two ELMs is investigated.

2.1. Background and notations

Assume that an output variable y depends of d input variables x_1, \dots, x_d through the relationship

$$y = f(\mathbf{x}) + \varepsilon(\mathbf{x}),$$

where $\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d$ is the vector composed by the input variables, $(\cdot)^T$ denotes the transpose operator, f is a function of \mathbf{x} whose value represent the deterministic part of y , and $\varepsilon(\mathbf{x})$ is a random noise depending on the input representing the stochastic part of y . It is assumed that, whatever the value of \mathbf{x} , the noise is centered and has a finite variance.

Let the training set $\mathcal{D} = \{(\mathbf{x}_i, y_i) : \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}\}_{i=1}^n$ be a sample from the joint distribution of (\mathbf{x}, y) . Given a new input point $\mathbf{x}_0 \in \mathbb{R}^d$, one wants to predict the corresponding output y_0 . The value $f(\mathbf{x}_0)$ seems a good guess. However, the function f is unknown. The true function f needs to be approximated based on the sample \mathcal{D} in order to provide an estimate of the prediction $f(\mathbf{x}_0)$.

90 For convenience, the matrix composed by all training input points will be noted $X = [\mathbf{x}_1 \mid \dots \mid \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. Moreover, at the training points, the n -dimensional vectors $\mathbf{y} = (y_1, \dots, y_n)^T$, $\mathbf{f} = (f(\mathbf{x}_1), \dots, f(\mathbf{x}_n))^T$, and $\boldsymbol{\varepsilon} = (\varepsilon(\mathbf{x}_1), \dots, \varepsilon(\mathbf{x}_n))^T$ are defined. The covariance matrix of $\boldsymbol{\varepsilon}$ knowing X is noted $\Sigma \in \mathbb{R}^{n \times n}$.

95 2.1.1. *Extreme Learning Machine*

ELM is a single-layer FNN with a random initialization of the input weights $\mathbf{w}_j \in \mathbb{R}^d$ and biases $b_j \in \mathbb{R}$, for $j = 1, \dots, N$, where N denotes the number of neurons of the hidden layer. All input weights and biases are independent and identically distributed (i.i.d.), and are generally sampled from a Gaussian or uniform distribution. They map the input space into a random feature space in a non-linear fashion by-way-of the *non-linear feature mapping* [11]

$$\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_N(\mathbf{x}))^T \in \mathbb{R}^N,$$

where $h_j(\mathbf{x}) = g(\mathbf{x}^T \mathbf{w}_j + b_j)$ is the output of the j -th hidden node, for $j = 1, \dots, N$, and g is any infinitely differentiable activation function [12]. A N -hidden neurons ELM can generate output functions of the form

$$f_N(\mathbf{x}) = \mathbf{h}(\mathbf{x})^T \boldsymbol{\beta},$$

where $\boldsymbol{\beta} \in \mathbb{R}^N$ is the vector of output weights that relate the hidden layer with the output node.

The output weights are trained using the sample \mathcal{D} and optimized regarding the L_2 criterion performing the following procedure. The $n \times N$ hidden layer matrix, denoted H and defined element-wise by $H_{ij} = h_j(\mathbf{x}_i)$, $i = 1, \dots, n$, and $j = 1, \dots, N$, is computed. Then the cost function J_1 , defined by

$$J_1(\boldsymbol{\beta}) = \|\mathbf{y} - H\boldsymbol{\beta}\|_2^2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm, is minimized. This is exactly the LS procedure for a design matrix H [13, 14]. If the matrix $H^T H$ is of full rank and then invertible, the output weights are estimated as the classical linear regression, with the analytical solution of the minimization of E ,

$$\hat{\boldsymbol{\beta}} = (H^T H)^{-1} H^T \mathbf{y},$$

where the matrix $(H^T H)^{-1} H^T$ is the Moore-Penrose generalized inverse of the matrix H and will be denoted H^\dagger in the following. Thus, ELM can be thought as a MLR with a null intercept, performed on regressors obtained by a random non-linear transformation of the input variables.

At a new point, the prediction is given by $\hat{f}(\mathbf{x}_0) = \mathbf{h}(\mathbf{x}_0)^T \hat{\boldsymbol{\beta}}$. In the remainder of the paper, all dependencies in \mathbf{x}_0 will be dropped for convenience and the prediction will be noted $\hat{f}(\mathbf{x}_0) = \mathbf{h}^T \hat{\boldsymbol{\beta}} = \mathbf{h}^T H^\dagger \mathbf{y}$. The vector of model predictions at training points will be noted $\hat{\mathbf{f}}$, defined element-wise by $\hat{\mathbf{f}}_i = \hat{f}(\mathbf{x}_i)$. The (random) matrix of all input weights and biases will be denoted

$$W = \left[\begin{array}{c|c|c} \mathbf{w}_1 & \dots & \mathbf{w}_N \\ \hline b_1 & \dots & b_N \end{array} \right].$$

2.1.2. *Regularized Extreme Learning Machine*

To avoid overfitting and reduce outlier effects, a regularized version of ELM was proposed [15]. Highly variable output weights due to multicollinearity among

neurons can be stabilized with regularization, too. As mentioned by [16], this model is basically a particular case of *Tikhonov regularization* — also known as *ridge regression* [17] — performed on the random feature space. The output weights are optimized regarding the cost function J_2 , where

$$J_2(\boldsymbol{\beta}) = \|\mathbf{y} - H\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_2^2,$$

for some real number $\alpha > 0$, sometimes called the *Tikhonov factor*, which controls the penalization of taking big output weights. Noting I the identity matrix, the analytical solution of this optimization problem for a fixed α is given by

$$\hat{\boldsymbol{\beta}} = (H^T H + \alpha I)^{-1} H^T \mathbf{y},$$

thanks to the fact that the matrix $(H^T H + \alpha I)$ is always invertible, see [18]. To lighten the notation, the matrix $H^\alpha = (H^T H + \alpha I)^{-1} H^T$ is defined. Remark 105 that as α goes to zero, H^α goes to H^\dagger and the classical ELM is recovered [15]. In the remainder of the paper, most of results are presented with H^α , but they remain valid for the non-regularized case, unless the contrary is clearly specified.

2.1.3. Extreme Learning Machine Ensemble

Another way to avoid overfitting is to combine several ELM models. This also reduce the randomness induced by the input weight initialization, which could be beneficial — especially for small datasets. Several ensemble techniques have been developed for ELM [19, 16]. In this paper, each model of a given ensemble will have the same activation function and number of neurons, and all models will be averaged after training. This corresponds to retrain M times the model and average the results, where M is the number of ELM networks in the ensemble. The hidden layer matrix and the matrix of input weights and biases of the m -th retraining will be noted respectively H_m and W_m , for $m = 1, \dots, M$. If the m -th prediction is noted $\hat{f}_m(\mathbf{x}_0)$, the final prediction is

$$\hat{f}(\mathbf{x}_0) = \frac{1}{M} \sum_{m=1}^M \hat{f}_m(\mathbf{x}_0) = \frac{1}{M} \sum_{m=1}^M \mathbf{h}_m^T H_m^\alpha \mathbf{y},$$

where \mathbf{h}_m and H_m^α are the analogous quantities defined previously for the m -th model. Note that the weights have the same joint distribution across all 110 the models, which allows us to drop the m index in most calculations of the remainder of this paper.

2.2. Bias and variance for a single ELM

In this section, the uncertainty for (regularized) ELM is explored. For all derivations, it is supposed that hyper-parameters N and α — when applicable — are considered as fixed and non-stochastic. Also, all formulas are derived knowing X . However, this conditioning is dropped to avoid cumbersome notations. Note that $\hat{f}(\mathbf{x}_0)$ is a random variable depending on the noise at training

points ε , but also on the input weights and biases W used in the construction of \mathbf{h} and H^α . As the noise is centred,

$$\mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W \right] = \mathbf{h}^T H^\alpha \mathbb{E}[\mathbf{f} + \varepsilon] = \mathbf{h}^T H^\alpha \mathbf{f}. \quad (1)$$

Using the law of total expectation, one can compute the bias of the model at \mathbf{x}_0 ,

$$\text{Bias} \left[\hat{f}(\mathbf{x}_0) \right] = \mathbb{E} \left[\mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W \right] \right] - f(\mathbf{x}_0) = \mathbb{E} \left[\mathbf{h}^T H^\alpha \mathbf{f} \right] - f(\mathbf{x}_0).$$

Let us now compute the variance of the model at a new point. First, one have

$$\text{Var} \left[\hat{f}(\mathbf{x}_0) \mid W \right] = \mathbf{h}^T H^\alpha \text{Var}[\mathbf{f} + \varepsilon] H^{\alpha T} \mathbf{h} = \mathbf{h}^T H^\alpha \Sigma H^{\alpha T} \mathbf{h}, \quad (2)$$

which is the typical variance expression for MLR. With equations (1) and (2), the variance of the model at \mathbf{x}_0 can be computed by using the law of total variance,

$$\begin{aligned} \text{Var} \left[\hat{f}(\mathbf{x}_0) \right] &= \mathbb{E} \left[\text{Var} \left[\hat{f}(\mathbf{x}_0) \mid W \right] \right] + \text{Var} \left[\mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W \right] \right] \\ &= \mathbb{E} \left[\mathbf{h}^T H^\alpha \Sigma H^{\alpha T} \mathbf{h} \right] + \text{Var} \left[\mathbf{h}^T H^\alpha \mathbf{f} \right]. \end{aligned} \quad (3)$$

The first term of the right-hand side (RHS) is the variance of the LS step averaged on all possible random feature spaces generated by input weights and biases, while the second term is the variation of the LS step bias across all random feature spaces. Note that the second term appears if and only if the random input weights and biases are considered. In the non-regularized case with independent homoskedastic noise, if W and X are deterministic, the classical MLR formula for the variance at a prediction point is recovered, see [14].

2.3. Bias and variance for ELM ensemble

As mentioned before, the training could be done several times and averaged. A direct calculation — which can be found in the appendix — can be done for bias and variance of the averaged predictor. Basically, it uses the law of total variance and elementary probability calculus from which one get

$$\text{Var} \left[\hat{f}(\mathbf{x}_0) \right] = \frac{1}{M} \mathbb{E} \left[\mathbf{h}^T H^\alpha \Sigma H^{\alpha T} \mathbf{h} \right] + \frac{M-1}{M} \mathbb{E} \left[\mathbf{h}^T H^\alpha \right] \Sigma \mathbb{E} \left[H^{\alpha T} \mathbf{h} \right] + \frac{1}{M} \text{Var} \left[\mathbf{h}^T H^\alpha \mathbf{f} \right], \quad (4)$$

while the bias still unchanged. The RHS first and third terms are the single ELM variance divided by the number of models. The bias variation of the LS step is reduced by a $1/M$ factor. Although the average variance of the LS step represented by the RHS first term seems to decrease by a $1/M$ factor, models are pairwise dependent which yields the RHS second term. Notice that if $M = 1$, equation (3) is recovered. If M grows, the RHS second term tends to dominate the model variance. Remark also that using the law of total covariance, it is easily checked by analogous computation that the covariance between two members of an ELM ensemble correspond to $\mathbb{E} \left[\mathbf{h}^T H^\alpha \right] \Sigma \mathbb{E} \left[H^{\alpha T} \mathbf{h} \right]$.

2.4. Use of random variable quadratic forms

Formulation of variance in equations (3) and (4) are convenient for the interpretation of ELM as a MLR on random features. However, quadratic forms in random variables appears in these formulas, which allows to pursue calculations. With the Corollary 3.2b.1 of [20], the expectation of random variable quadratic form can be computed as the quadratic form in its expected values plus the trace of its covariance matrix times the matrix of the quadratic form. This Corollary will be used extensively in this paper, each time an expectation of a quadratic form in random variables appears.

Setting the random vector $\mathbf{z} = H^{\alpha T} \mathbf{h}$ and assuming the existence of its expectation $\boldsymbol{\mu}$ and covariance matrix C , equation (3) becomes

$$\text{Var} \left[\hat{f}(\mathbf{x}_0) \right] = \text{Tr} [\Sigma C] + \boldsymbol{\mu}^T \Sigma \boldsymbol{\mu} + \mathbf{f}^T C \mathbf{f},$$

where $\text{Tr}[\cdot]$ denotes the trace of a square matrix. Although the notation do not specify it, the quantities \mathbf{z} , $\boldsymbol{\mu}$ and C depend on the Tikhonov factor α in the regularized case. Similarly, $\mathbf{z}_m = H_m^{\alpha T} \mathbf{h}_m$ is set for ELM ensembles and the variance becomes

$$\text{Var} \left[\hat{f}(\mathbf{x}_0) \right] = \frac{1}{M} \text{Tr} [\Sigma C] + \boldsymbol{\mu}^T \Sigma \boldsymbol{\mu} + \frac{1}{M} \mathbf{f}^T C \mathbf{f}. \quad (5)$$

2.5. Correlation between two ELMs

As the covariance between two single ELMs $\hat{f}_1(\mathbf{x}_0)$ and $\hat{f}_2(\mathbf{x}_0)$ is $\boldsymbol{\mu}^T \Sigma \boldsymbol{\mu}$, their linear correlation at \mathbf{x}_0 is given by

$$\text{Corr} \left[\hat{f}_1(\mathbf{x}_0), \hat{f}_2(\mathbf{x}_0) \right] = \frac{\boldsymbol{\mu}^T \Sigma \boldsymbol{\mu}}{\text{Tr} [\Sigma C] + \boldsymbol{\mu}^T \Sigma \boldsymbol{\mu} + \mathbf{f}^T C \mathbf{f}}.$$

Remark that considering the input weights and biases as fixed is equivalent to ignore $\text{Tr} [\Sigma C] + \mathbf{f}^T C \mathbf{f}$ and to have a correlation of 1 between the two models.

An interesting insight is provided by the case of independent and homoskedastic noise, i.e. $\Sigma = \sigma_\varepsilon^2 I$ with $\sigma_\varepsilon^2 \in \mathbb{R}$, which yields

$$\text{Corr} \left[\hat{f}_1(\mathbf{x}_0), \hat{f}_2(\mathbf{x}_0) \right] = \frac{\boldsymbol{\mu}^T \boldsymbol{\mu}}{\text{Tr} [C] + \boldsymbol{\mu}^T \boldsymbol{\mu} + \frac{1}{\sigma_\varepsilon^2} \mathbf{f}^T C \mathbf{f}}.$$

Notice that in this particular case, when σ_ε^2 is small the linear correlation between two ELMs vanishes. Contrariwise, when σ_ε^2 is large the linear correlation between two ELMs tends to $b = \boldsymbol{\mu}^T \boldsymbol{\mu} / (\boldsymbol{\mu}^T \boldsymbol{\mu} + \text{Tr} [C])$. Therefore, the amount of noise has a direct impact on the linear correlation which takes its value between 0 and $b \leq 1$. The trace of C can be interpreted as a variability measure of \mathbf{z} , called sometimes the *total variation* or the *total dispersion* of \mathbf{z} [21]. In our case, it controls the linear correlation bound b , in the sense that more variable is \mathbf{z} , farther from 1 is the maximal value that the linear correlation can take, regardless the noise in the data.

3. Variance estimation of ELM ensemble

This section introduces novel estimates of the ELM variance. Although several ELMs are necessary to allow the estimation of quantities related to \mathbf{z} — which motivates the use of ELM ensembles — reliable results are also obtained with very small ensemble. First, the variation of the LS step bias overall random feature space is estimated. Then, assuming noise independence, the variance of the LS step averaged on all possible random feature spaces is estimated under homoskedasticity and heteroskedasticity for non-regularized and regularized ELM ensembles.

3.1. Estimation of the least squares bias variation

The quantity $\mathbf{f}^T C \mathbf{f} = \text{Var} [\mathbf{h}^T H^\alpha \mathbf{f}]$ doesn't depend on noise. It is the variance induced by the randomness of W , knowing the true function f at training points. Tentatively assume that the output weights are not regularized. As \mathbf{f} is unknown, one approximates it by the model prediction at the training points $\hat{\mathbf{f}} = HH^\dagger \mathbf{y}$. For each model of the ensemble,

$$\text{Var} [\mathbf{h}^T H^\dagger \mathbf{f}] \approx \text{Var} [\mathbf{h}^T H^\dagger \hat{\mathbf{f}} \mid \boldsymbol{\varepsilon}] = \text{Var} [\hat{f}(\mathbf{x}_0) \mid \boldsymbol{\varepsilon}].$$

This motivates the following estimate for $\mathbf{f}^T C \mathbf{f}$,

$$\hat{\sigma}_f^2 = \frac{1}{M-1} \sum_{m=1}^M \left(\hat{f}_m(\mathbf{x}_0) - \frac{1}{M} \sum_{l=1}^M \hat{f}_l(\mathbf{x}_0) \right)^2. \quad (6)$$

The same estimate will be used for the regularized case.

The expectancy of $\hat{\sigma}_f^2$ can be easily computed. Knowing the noise at the training points,

$$\begin{aligned} \mathbb{E} [\hat{\sigma}_f^2 \mid \boldsymbol{\varepsilon}] &= \text{Var} [\hat{f}(\mathbf{x}_0) \mid \boldsymbol{\varepsilon}] = \text{Var} [\mathbf{z}^T \mathbf{y} \mid \boldsymbol{\varepsilon}] \\ &= \text{Var} [\mathbf{z}^T \mathbf{f} \mid \boldsymbol{\varepsilon}] + \text{Var} [\mathbf{z}^T \boldsymbol{\varepsilon} \mid \boldsymbol{\varepsilon}] + 2\text{Cov} [\mathbf{z}^T \mathbf{f}, \mathbf{z}^T \boldsymbol{\varepsilon} \mid \boldsymbol{\varepsilon}] \\ &= \text{Var} [\mathbf{z}^T \mathbf{f}] + \boldsymbol{\varepsilon}^T C \boldsymbol{\varepsilon} + 2\mathbf{f}^T C \boldsymbol{\varepsilon}, \end{aligned}$$

using the unbiasedness of the estimate in the first equality. By taking the expectation over $\boldsymbol{\varepsilon}$ on both sides,

$$\mathbb{E} [\hat{\sigma}_f^2] = \text{Var} [\mathbf{h}^T H^\alpha \mathbf{f}] + \text{Tr} [\Sigma C].$$

This shows that the bias of the estimate defined in equation (6) is given by $\text{Tr} [\Sigma C]$, which is the first term of the RHS of equation (5) up to a factor $1/M$. Therefore, regardless of a particular form of Σ or whether ELM is regularized or not, it is unnecessary to estimate the latter and $(1/M) \hat{\sigma}_f^2$ is an unbiased estimate of the sum of the first and last terms of equation (5).

3.2. Estimation under independence and homoskedastic assumptions

Only the second term of RHS of equation (5) remains to be estimated. If the noise is assumed to be independent and have a constant variance, the covariance matrix of ε writes $\Sigma = \sigma_\varepsilon^2 I$ and the second term of RHS of equation (5) becomes $\boldsymbol{\mu}^T \Sigma \boldsymbol{\mu} = \sigma_\varepsilon^2 \boldsymbol{\mu}^T \boldsymbol{\mu}$. The quantity $\boldsymbol{\mu}^T \boldsymbol{\mu}$ — which, knowing X , stochastically depends only on W — will be estimated separately from σ_ε^2 .

3.2.1. Estimation of $\boldsymbol{\mu}^T \boldsymbol{\mu}$

As a first step, $\boldsymbol{\mu}^T \boldsymbol{\mu}$ is naively estimated with

$$\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} = \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{z}_m^T \mathbf{z}_l, \quad \text{where} \quad \hat{\boldsymbol{\mu}} = \frac{1}{M} \sum_{m=1}^M \mathbf{z}_m. \quad (7)$$

However, remark that

$$\begin{aligned} M^2 \mathbb{E} [\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}] &= \sum_{m=1}^M \mathbb{E} [\mathbf{z}_m^T \mathbf{z}_m] + \sum_{m \neq l} \mathbb{E} [\mathbf{z}_m]^T \mathbb{E} [\mathbf{z}_l] \\ &= M (\text{Tr} [C] + \boldsymbol{\mu}^T \boldsymbol{\mu}) + M(M-1) \boldsymbol{\mu}^T \boldsymbol{\mu} \\ &= M \text{Tr} [C] + M^2 \boldsymbol{\mu}^T \boldsymbol{\mu}, \end{aligned}$$

and dividing by M^2 shows that the estimate given in (7) has a bias equal to $(1/M) \text{Tr} [C]$, which comes from the expected values of the M quadratic terms in \mathbf{z}_m .

To remove this bias, one can estimate it by

$$\frac{1}{M} \text{Tr} [\hat{Q}], \quad \text{where} \quad \hat{Q} = \frac{1}{M-1} \sum_{m=1}^M (\mathbf{z}_m - \hat{\boldsymbol{\mu}}) (\mathbf{z}_m - \hat{\boldsymbol{\mu}})^T. \quad (8)$$

This is an unbiased estimate of $(1/M) \text{Tr} [C]$, which immediately follows from the fact that \hat{Q} is an unbiased estimate of C . Therefore, subtract (8) from (7) yields an unbiased estimate of $\boldsymbol{\mu}^T \boldsymbol{\mu}$. Note that

$$\begin{aligned} (M-1) \text{Tr} [\hat{Q}] &= \sum_{m=1}^M (\mathbf{z}_m - \hat{\boldsymbol{\mu}})^T (\mathbf{z}_m - \hat{\boldsymbol{\mu}}) \\ &= \sum_{m=1}^M \mathbf{z}_m^T \mathbf{z}_m - M \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}. \end{aligned}$$

Hence, the unbiased estimate of $\boldsymbol{\mu}^T \boldsymbol{\mu}$ that was just developed results in

$$\begin{aligned} \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr} [\hat{Q}] &= \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M(M-1)} \sum_{m=1}^M \mathbf{z}_m^T \mathbf{z}_m + \frac{1}{M-1} \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} \\ &= \frac{M}{M-1} \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M(M-1)} \sum_{m=1}^M \mathbf{z}_m^T \mathbf{z}_m. \end{aligned} \quad (9)$$

Substituting equation (7) into equation (9), the computation still goes on, and

$$\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] = \frac{1}{M(M-1)} \sum_{m \neq l} \mathbf{z}_m^T \mathbf{z}_l. \quad (10)$$

This shows that the estimate (10) removes the quadratic terms $m = l$ from which the bias of the naive estimate (7) was induced. However, note that formulation
 180 (9) is more convenient to compute than (10), from an algorithmic perspective.

3.2.2. Noise estimation

The estimation of σ_ε^2 is separated in two cases, the non-regularized and the regularized ones. A couple of notations is needed to make readable the equations. The residuals for the m -th model are $\mathbf{r}_m = \mathbf{y} - P_m \mathbf{y}$, where $P_m = H_m H_m^\alpha$. Also, $\mathbf{b} = \mathbf{f} - \mathbb{E}[\hat{\mathbf{f}}] = (I - \mathbb{E}[P])\mathbf{f}$ is the vector of bias of the model
 185 predictions at the training points $\hat{\mathbf{f}}$, and $\mathbf{b}_{w,m} = \mathbf{f} - \mathbb{E}[\hat{\mathbf{f}} | W_m] = (I - P_m)\mathbf{f}$ is the vector of conditional bias of the model predictions at the training points $\hat{\mathbf{f}}$ for the m -th model knowing W_m .

Let us first concentrate on the non-regularized case. Then, P_m is a projection matrix. A natural way to obtain estimate for $\hat{\sigma}_\varepsilon^2$ is to start with the expectation of the residual sum of squares (RSS) based on the averaged ensemble. However, mainly due to the fact that the expectation of a projection matrix is not a projection matrix, it is preferred here to work with the RSS of each model. Using this, the expectation of the residual sum of squares for the m -th ELM knowing input weights and biases is given by

$$\begin{aligned} \mathbb{E}[\mathbf{r}_m^T \mathbf{r}_m | W_m] &= \mathbb{E}[\mathbf{y}^T (I - P_m) \mathbf{y} | W_m] \\ &= \sigma_\varepsilon^2 \text{Tr}[I - P_m] + \mathbf{f}^T (I - P_m) \mathbf{f} \\ &= \sigma_\varepsilon^2 (n - N) + \mathbf{f}^T \mathbf{b}_{w,m}, \end{aligned} \quad (11)$$

and taking the expectation over the input weights and biases yields $\mathbb{E}[\mathbf{r}_m^T \mathbf{r}_m] = \sigma_\varepsilon^2 (n - N) + \mathbf{f}^T \mathbf{b}$. This motivates the following estimate,

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{M(n - N)} \sum_{m=1}^M \mathbf{r}_m^T \mathbf{r}_m, \quad (12)$$

which is the average of all MLR estimates of σ_ε^2 . Its bias is directly obtained from previous calculation, yielding

$$\text{Bias}[\hat{\sigma}_\varepsilon^2] = \frac{1}{n - N} \mathbf{f}^T \mathbf{b} \geq 0, \quad (13)$$

where non-negativity results from the facts that a projection matrix is positive
 190 semidefinite and expectation of a positive semidefinite matrix is positive semidefinite.

If regularized ELMs are used, P_m is no more a projection matrix, and the expected RSS for each ELM of the ensemble knowing W_m becomes

$$\begin{aligned}\mathbb{E}[\mathbf{r}_m^T \mathbf{r}_m \mid W_m] &= \mathbb{E}[\mathbf{y}^T (I - P_m)^T (I - P_m) \mathbf{y} \mid W_m] \\ &= \sigma_\varepsilon^2 \text{Tr}[(I - P_m)^T (I - P_m)] + \mathbf{f}^T (I - P_m)^T (I - P_m) \mathbf{f} \quad (14) \\ &= \sigma_\varepsilon^2 (n - 2\text{Tr}[P_m] + \text{Tr}[P_m^2]) + \mathbf{b}_{w,m}^T \mathbf{b}_{w,m}.\end{aligned}$$

Analogously to what is done in [22], the *effective degrees of freedom for error* can be defined as $n - \gamma$, with $\gamma = 2\mathbb{E}[\text{Tr}[P]] - \mathbb{E}[\text{Tr}[P^2]]$. Expectation over input weights and biases of equation (14) gives $\mathbb{E}[\mathbf{r}_m^T \mathbf{r}_m] = \sigma_\varepsilon^2 (n - \gamma) + \mathbb{E}[\mathbf{b}_w^T \mathbf{b}_w]$, hence,

$$\begin{aligned}\mathbb{E}\left[\frac{1}{M(n - \gamma)} \sum_{m=1}^M \mathbf{r}_m^T \mathbf{r}_m\right] &= \sigma_\varepsilon^2 + \frac{1}{n - \gamma} \mathbb{E}[\mathbf{b}_w^T \mathbf{b}_w] \\ &= \sigma_\varepsilon^2 + \frac{1}{n - \gamma} (\mathbf{b}^T \mathbf{b} + \text{Tr}[\text{Var}[\mathbf{b}_w]]),\end{aligned}$$

which makes appear the squared bias and the total variation of the conditional bias. This motivates the following estimate in the regularized case,

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{M(n - \hat{\gamma})} \sum_{m=1}^M \mathbf{r}_m^T \mathbf{r}_m, \quad \text{with} \quad \hat{\gamma} = \frac{1}{M} \sum_{m=1}^M (2\text{Tr}[P_m] - \text{Tr}[P_m^2]), \quad (15)$$

and it is easy to check that

$$\text{Bias}[\hat{\sigma}_\varepsilon^2] = \mathbb{E}\left[\frac{1}{n - \hat{\gamma}} \mathbf{b}_w^T \mathbf{b}_w\right] \geq 0. \quad (16)$$

Computationally, $\hat{\gamma}$ can be efficiently calculated using the singular value decomposition of H_m . Indeed, it can be shown [17, 23] that the trace of P_m and P_m^2 are given by

$$\text{Tr}[P_m] = \sum_{i=1}^N \frac{\lambda_{m,i}}{\lambda_{m,i} + \alpha} \quad \text{and} \quad \text{Tr}[P_m^2] = \sum_{i=1}^N \left(\frac{\lambda_{m,i}}{\lambda_{m,i} + \alpha}\right)^2, \quad (17)$$

where $\lambda_{m,i}$, $i = 1, \dots, N$ are the eigenvalues of $H_m^T H_m$. In particular, substitution of (17) in (15) and elementary manipulations allow to write

$$\hat{\gamma} = N - \frac{1}{M} \sum_{m=1}^M \sum_{i=1}^N \left(\frac{\alpha}{\lambda_{m,i} + \alpha}\right)^2$$

where $\sqrt{\lambda_{m,i}}$, $i = 1, \dots, N$ are the singular values of H_m . Note that this latter equation also shows the drop in the degrees of freedom lost due to regularization, comparing to the non-regularized case.

195 *3.2.3. Estimations of ELM ensemble variance*

In [24], the authors proposed — only for the non-regularized case — the following naive homoskedastic estimate of the variance of $\hat{f}(\mathbf{x}_0)$,

$$\hat{\sigma}_{NHo}^2 = \hat{\sigma}_\varepsilon^2 \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} + \frac{1}{M} \hat{\sigma}_f^2.$$

This naive estimate directly use equation (7) to approximate $\boldsymbol{\mu}^T \boldsymbol{\mu}$ without considering its bias. However, a bias-reduced estimate is obtained by estimating $\boldsymbol{\mu}^T \boldsymbol{\mu}$ by equation (10), yielding

$$\hat{\sigma}_{BR}^2 = \hat{\sigma}_\varepsilon^2 \left(\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] \right) + \frac{1}{M} \hat{\sigma}_f^2.$$

Using the covariance definition, it is easy to see that

$$\text{Bias} [\hat{\sigma}_{BR}^2] = \text{Bias} [\hat{\sigma}_\varepsilon^2] \boldsymbol{\mu}^T \boldsymbol{\mu} + \text{Cov} \left[\hat{\sigma}_\varepsilon^2, \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] \right], \quad (18)$$

where the bias of $\hat{\sigma}_\varepsilon^2$ is (13) or (16). Note that in both cases, while the first term of the RHS of equation (18) is always non-negative, the estimates of $\hat{\sigma}_\varepsilon^2$ and $\boldsymbol{\mu}^T \boldsymbol{\mu}$ could be correlated, introducing the second term. However, this supplementary bias could be negative, potentially compensating the first term. Note that its magnitude is bounded by

$$\left| \text{Cov} \left[\hat{\sigma}_\varepsilon^2, \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] \right] \right| \leq \frac{2}{M(n-N)} \sqrt{\mathbf{f}^T \text{Var}[\mathbf{b}_w] \mathbf{f} \cdot \boldsymbol{\mu}^T C \boldsymbol{\mu}}, \quad (19)$$

in the non-regularized case — see the appendix — showing that this covariance term vanishes with large M . Remark also that the bias of $\hat{\sigma}_{NHo}^2$ has an additional non-negative term, $(1/M) \text{Tr}[C] \mathbb{E}[\hat{\sigma}_\varepsilon^2]$, which disappears in (18) thanks to the unbiasedness of $\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - (1/M) \text{Tr}[\hat{Q}]$.

200

3.3. Estimation under independence and heteroskedastic assumptions

Suppose the noise is independent but have variance which have a dependence of unknown form on \mathbf{x} . Then $S = \boldsymbol{\mu}^T \Sigma \boldsymbol{\mu}$ has to be estimated considering the noise covariance matrix Σ as diagonal. To this aim, it could be possible to reuse estimates from MLR. However, several estimates are based on the evaluation of the covariance matrix $H_m^\alpha \Sigma_m H_m^{\alpha T}$ of the output weights $\hat{\boldsymbol{\beta}}_m$. In this paper, the modified heteroskedastic-consistent covariance matrix estimator (HCCME) obtained from the (ordinary) Jackknife [25] — noted HC₃ and extended to the ridge regression case [26] — is used,

$$\text{HC}_3 = H_m^\alpha \hat{\Sigma}_m H_m^{\alpha T}, \quad \text{with} \quad \hat{\Sigma}_m = \frac{n-1}{n} \left[\hat{\Omega}_m - \frac{1}{n} \tilde{\mathbf{r}}_m \tilde{\mathbf{r}}_m^T \right], \quad (20)$$

where $\tilde{\mathbf{r}}_m$ is the vector defined element-wise by $\tilde{r}_{m,i} = \mathbf{r}_{m,i}/(1-p_{m,i})$, $p_{m,i}$ is the i -th diagonal element of P_m and $\hat{\Omega}_m$ is the diagonal matrix with the i -th diagonal

element equal to $\hat{\mathbf{r}}_{m,i}^2$. This estimate is still valid for the non-regularized case, for which — under some technical assumptions — it is consistent [27, 25], while $\hat{\Sigma}_m$ is an inconsistent estimator of Σ . Nevertheless, other HCCME estimates could be used, such as HC₀ [27], HC₁ got from the weighted Jackknife [28], HC₂ proposed in [29] or HC₄ proposed more recently in [30]. The HC notation follows what can be found in [13], which provides useful insight on this kind of estimators. Note that for sufficiently large n , HC₃ is close to $H_m^\alpha \hat{\Omega}_m H_m^{\alpha T}$, which corresponds to the estimate used in [9] to build prediction intervals for large amounts of data, assuming fixed input weights.

If a unique ELM model is performed, the use of the HCCME is straightforward. Nonetheless, as one attempts to take into account the input weight variability through ELM replications, the HCCME is applied in three different ways. Suppose first that Σ is known and write the estimate

$$\hat{\boldsymbol{\mu}}^T \Sigma \hat{\boldsymbol{\mu}} = \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{z}_m^T \Sigma \mathbf{z}_l = \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{h}_m^T H_m^\alpha \Sigma H_l^{\alpha T} \mathbf{h}_l. \quad (21)$$

Inspecting equations (20) and (21), a first natural suggestion is to estimate $\boldsymbol{\mu}^T \Sigma \boldsymbol{\mu}$ with

$$\hat{S}_1 = \frac{1}{M} \sum_{m=1}^M \mathbf{z}_m^T \hat{\Sigma}_m \mathbf{z}_m.$$

Note that \hat{S}_1 estimates the covariance matrix of the output weights with the HCCME for each of the M random feature spaces. Although it has the advantage of reusing the HCCME in its original formulation, the quadratic forms in random vectors depending on input weight may produce an additional bias. Another estimator is obtained by naively evaluating all cross terms of equation (21),

$$\hat{S}_{NHe} = \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{z}_m^T \hat{\Sigma}_m \mathbf{z}_l.$$

Note that this is equivalent to estimate Σ with $(\hat{\Sigma}_m + \hat{\Sigma}_l)/2$ in equation (21), see [24]. However, terms for which $m = l$ may produce additional biases, similarly to what was shown for the homoskedastic estimate of equation (7), which motivates

$$\hat{S}_2 = \frac{1}{M(M-1)} \sum_{m \neq l} \mathbf{z}_m^T \hat{\Sigma}_m \mathbf{z}_l.$$

Analogously to the homoskedastic case — see equation (10) — the terms corresponding to $m = l$ are not taken into account in \hat{S}_2 , avoiding the introduction of potential biases from quadratic forms. Remark also that

$$\hat{S}_2 = \frac{1}{M-1} \left[M \hat{\boldsymbol{\nu}}^T \hat{\boldsymbol{\mu}} - \hat{S}_1 \right], \quad \text{where} \quad \hat{\boldsymbol{\nu}} = \frac{1}{M} \sum_{m=1}^M \hat{\Sigma}_m \mathbf{z}_m,$$

which is algorithmically be more convenient to compute. Also, $\widehat{S}_{NH_e} = \widehat{\boldsymbol{\nu}}^T \widehat{\boldsymbol{\mu}}$. The estimates \widehat{S}_{NH_e} and \widehat{S}_2 have some similarities with $\widehat{\boldsymbol{\mu}}^T \widehat{\boldsymbol{\mu}}$ and $\widehat{\boldsymbol{\mu}}^T \widehat{\boldsymbol{\mu}} - (1/M)\text{Tr}[C]$ as estimated in the homoskedastic case, see section 3.2.1. However, \mathbf{z}_m still interacts with the covariance matrix estimate within $\widehat{\boldsymbol{\nu}}$. This motivates a third estimate,

$$\widehat{S}_3 = \frac{(M-3)!}{M!} \sum_{(m,l,k) \in A_M^3} \mathbf{z}_m^T \widehat{\Sigma}_k \mathbf{z}_l,$$

where A_M^3 is the set of 3-permutations of M . Looking at equation (21), \widehat{S}_3 can also be obtained by replacing Σ by a single estimate consisting of the average of the estimate of each model, where terms corresponding to $m = l$, $m = k$, or $l = k$ are ignored to avoid additional biases. To compute efficiently \widehat{S}_3 , it can be rewritten as

$$\widehat{S}_3 = \frac{1}{(M-1)(M-2)} \left[M^2 \widehat{\boldsymbol{\mu}}^T \widehat{U} \widehat{\boldsymbol{\mu}} - M \widehat{V} - 2(M-1) \widehat{S}_2 \right],$$

with

$$\widehat{U} = \frac{1}{M} \sum_{m=1}^M \widehat{\Sigma}_m \quad \text{and} \quad \widehat{V} = \frac{1}{M} \sum_{m=1}^M \mathbf{z}_m^T \widehat{U} \mathbf{z}_m.$$

Finally, the proposed heteroskedastic estimates of ELM ensemble variance, noted $\widehat{\sigma}_{S_1}^2$, $\widehat{\sigma}_{S_2}^2$, $\widehat{\sigma}_{S_3}^2$ and $\widehat{\sigma}_{NH_e}^2$, are given by respectively adding $(1/M)\widehat{\sigma}_f^2$ to \widehat{S}_1 , \widehat{S}_2 , \widehat{S}_3 and \widehat{S}_{NH_e} .

To increase computation speed, approximated versions of \widehat{S}_1 , \widehat{S}_2 , \widehat{S}_3 , and \widehat{S}_{NH_e} can be obtained by replacing $\widehat{\Sigma}_m$ by $\widehat{\Omega}_m$ in the above reasoning. As a matter of fact, these two matrices are very close for sufficiently large n , but $\widehat{\Omega}_m$ is a diagonal matrix, while $\widehat{\Sigma}_m$ is a full matrix. Remark that the approximated version of $\widehat{\sigma}_{NH_e}^2$ is exactly the heteroskedastic estimate proposed in [24].

4. Synthetic experiments

This section discusses the results obtained over different experimental settings. First, a simple non-regularized homoskedastic one-dimensional experiment is conducted. The variance estimate is thoroughly examined and assessed with quantitative measures and visualizations. Subsequently, the results are generalized to multi-dimensional settings with homoskedastic or heteroskedastic noise, both for the regularized and non-regularized cases. Finally, CI estimation is discussed. All the experiments presented will adopt the sigmoid as activation function, while input weights and biases will always be drawn uniformly between -1 and 1 . All computations are done with the provided Python library, see the software availability at the end of the paper for more details.

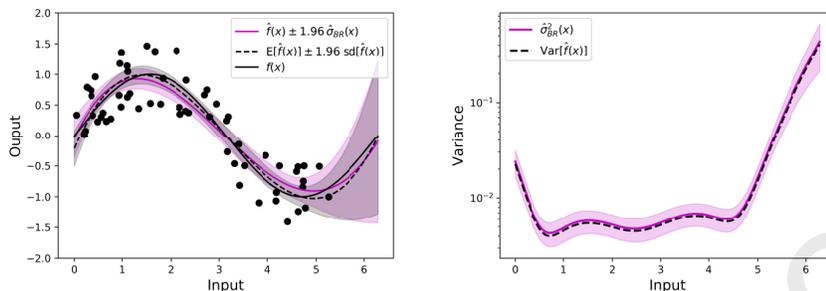


Figure 1: One-dimensional synthetic experiment : (left) In magenta, a single estimation of $\hat{f}(x)$ with $M = 10$ and its estimated ± 1.96 standard-error bands based on $\hat{\sigma}_{BR}^2$. In black dashed line, the mean of 10'000 ensembles for $M = 10$, with ± 1.96 standard-error bands. The true $f(x)$ is displayed in full black line; (right) In magenta, the mean of $\hat{\sigma}_{BR}^2$ for $M = 10$ with ± 1.96 standard-error bands, based on 1'000 replications of the experiment. In black dashed line, the variance computed from the 10'000 ensembles, considered as ground truth. Note the logarithmic scale on the y-axis.

4.1. One-dimensional case

To assess operationally the estimates proposed in section 3, a simple one-dimensional simulated case study of $n = 60$ training points is firstly proposed. A trapeze shape probability density function defined by

$$\rho(x) = -\frac{x}{4\pi^2} + \frac{3}{4\pi}, \quad \text{if } x \in [0, 2\pi],$$

$\rho(x) = 0$ otherwise, is used to draw the input, i.e. the number of data decreases as x increases. Outputs are generated according to

$$y = \sin(x) + \varepsilon,$$

where ε is an independent uniform noise $]-\sqrt{0.3}, \sqrt{0.3}[$ of constant variance $\sigma_\varepsilon^2 = 0.1$.

ELM ensembles are trained with $M = 5, 10, 20, 100$, allowing variance estimation. This experience is repeated 1'000 times. Each time, new outputs and new weights are drawn, but inputs are fixed. In order to avoid variability induced by hyper-parameter selection, a fixed number of neurons $N = 4$ was chosen by a 5-fold cross validation process repeated 5 times on 1'000 dataset generations. An example of one prediction is displayed in Figure 1 (left) for $M = 10$. Estimation of pointwise standard-error bands based on $\pm 1.96 \hat{\sigma}_{BR}$ is also reported.

Although simulated datasets are produced by user controlled processes, the true value of the variance of $\hat{f}(x)$ remains unknown. In order to evaluate the estimate, 10'000 ensembles with $M = 5, 10, 20, 100$ and $N = 4$ are trained with new outputs. The empirical mean and standard deviation of the 10'000 ensembles is relatively close to respectively $\mathbb{E}[\hat{f}(x)]$ and $\text{sd}[\hat{f}(x)] = (\text{Var}[\hat{f}(x)])^{1/2}$,

		$n = 60$			$n = 100$			
$M = 5$		BR	S3	Grnd tr.	BR	S3	Grnd tr.	
	Training set	se_k	0.077 (0.006)	0.080 (0.007)	0.075 (—)	0.058 (0.003)	0.060 (0.004)	0.058 (—)
		e_k	0.005 (0.004)	0.009 (0.005)	—	0.003 (0.002)	0.005 (0.002)	—
		re_k	0.066 (0.052)	0.113 (0.065)	—	0.046 (0.034)	0.077 (0.039)	—
Testing set	se_k	0.077 (0.006)	0.081 (0.007)	0.075 (—)	0.060 (0.003)	0.062 (0.004)	0.059 (—)	
	e_k	0.005 (0.004)	0.009 (0.005)	—	0.003 (0.002)	0.005 (0.002)	—	
	re_k	0.068 (0.053)	0.115 (0.066)	—	0.044 (0.033)	0.080 (0.039)	—	
$M = 10$		BR	S3	Grnd tr.	BR	S3	Grnd tr.	
	Training set	se_k	0.076 (0.005)	0.079 (0.006)	0.074 (—)	0.058 (0.003)	0.060 (0.004)	0.057 (—)
		e_k	0.005 (0.003)	0.008 (0.004)	—	0.003 (0.002)	0.004 (0.002)	—
		re_k	0.062 (0.046)	0.109 (0.057)	—	0.044 (0.032)	0.075 (0.037)	—
Testing set	se_k	0.076 (0.005)	0.080 (0.006)	0.074 (—)	0.060 (0.003)	0.062 (0.004)	0.059 (—)	
	e_k	0.005 (0.003)	0.009 (0.004)	—	0.003 (0.002)	0.005 (0.002)	—	
	re_k	0.062 (0.046)	0.111 (0.056)	—	0.043 (0.032)	0.078 (0.037)	—	
$M = 20$		BR	S3	Grnd tr.	BR	S3	Grnd tr.	
	Training set	se_k	0.076 (0.005)	0.079 (0.006)	0.074 (—)	0.058 (0.003)	0.060 (0.003)	0.057 (—)
		e_k	0.005 (0.003)	0.008 (0.004)	—	0.003 (0.002)	0.004 (0.002)	—
		re_k	0.061 (0.045)	0.109 (0.056)	—	0.042 (0.031)	0.074 (0.036)	—
Testing set	se_k	0.076 (0.005)	0.080 (0.006)	0.074 (—)	0.059 (0.003)	0.061 (0.004)	0.059 (—)	
	e_k	0.005 (0.003)	0.008 (0.004)	—	0.003 (0.002)	0.005 (0.002)	—	
	re_k	0.061 (0.044)	0.110 (0.056)	—	0.042 (0.031)	0.077 (0.037)	—	
$M = 100$		BR	S3	Grnd tr.	BR	S3	Grnd tr.	
	Training set	se_k	0.076 (0.005)	0.079 (0.006)	0.074 (—)	0.058 (0.003)	0.060 (0.003)	0.057 (—)
		e_k	0.004 (0.003)	0.008 (0.004)	—	0.002 (0.002)	0.004 (0.002)	—
		re_k	0.060 (0.045)	0.109 (0.055)	—	0.041 (0.031)	0.073 (0.036)	—
Testing set	se_k	0.076 (0.005)	0.080 (0.006)	0.074 (—)	0.059 (0.003)	0.061 (0.004)	0.059 (—)	
	e_k	0.004 (0.003)	0.008 (0.004)	—	0.002 (0.002)	0.005 (0.002)	—	
	re_k	0.060 (0.044)	0.110 (0.055)	—	0.041 (0.031)	0.077 (0.036)	—	

Table 1: Results of the one-dimensional synthetic experiment. Mean (standard deviation) of se_k , e_k and re_k .

and is reported as such for $M = 10$ in Figure 1. In particular, the empirical
 250 variance of the 10'000 ensembles will provide a reliable baseline for the variance
 estimation assessment and will be referred as the ground truth variance.

Figure 1 (right) shows the mean of $\hat{\sigma}_{BR}^2$ across the 1'000 experiments with
 ± 1.96 standard-error band. Compared with the ground truth variance, the
 proposed estimate recovers effectively — in average — the variance from the
 255 10'000 simulations baseline. The increasing variance in the borders due to the
 side effect of the modelling is fairly replicated. The uncertainty due to the
 trapezoidal shape of the input data distribution is also captured. Qualitatively,
 all aspects of the expected variance behaviour are globally reproduced. The
 naive estimate $\hat{\sigma}_{NHo}^2$ gives very similar results in one dimension, and is not
 260 shown. The improvement due to considering the bias of $\hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}$ will be more
 relevant in the multi-dimensional case. However, one can still observe a residual
 bias for $\hat{\sigma}_{BR}^2$, partially due to the bias of $\hat{\sigma}_\varepsilon^2$ and to the dependence between $\hat{\sigma}_\varepsilon^2$
 and the estimate of $\boldsymbol{\mu}^T \boldsymbol{\mu}$ — see equations (13) and (18).

To assess quantitatively each estimation, a measure is needed between the
 true standard error of $\hat{f}(\mathbf{x})$ and its estimations provided by each repetition of
 the experiment. Following [2], let us look at the median of the k^{th} standard

error estimate over the training set,

$$se_k = \text{median}_{1 \leq i \leq n} (\hat{\sigma}_k(\mathbf{x}_i)),$$

and the absolute error of the k^{th} standard error estimate over the training set defined by

$$e_k = \text{median}_{1 \leq i \leq n} |\hat{\sigma}_k(\mathbf{x}_i) - \sigma(\mathbf{x}_i)|,$$

where $\hat{\sigma}_k^2(\mathbf{x}_i)$ is an estimate of $\sigma^2(\mathbf{x}_i) = \text{Var}[\hat{f}(\mathbf{x}_i)]$, for $k = 1, \dots, 1'000$. Also, the relative error re_k of the k^{th} standard error estimate over the training set is defined by

$$re_k = \text{median}_{1 \leq i \leq n} \frac{|\hat{\sigma}_k(\mathbf{x}_i) - \sigma(\mathbf{x}_i)|}{\sigma(\mathbf{x}_i)},$$

for $k = 1, \dots, 1'000$. Similar measures are defined on a random testing set of 265 1'000 points. In order to compute these quantities, $\sigma(\mathbf{x}_i)$ is replaced by the ground truth standard deviation.

The means and standard deviations of se_k , e_k and re_k over the 1'000 experiment repetitions are presented in Table 1. For the training set, the median of the ground truth standard error is recovered by the median of $\hat{\sigma}_{BR}$, judging 270 through the se_k measure. Moreover, the mean and standard deviation of e_k appear quite small. The relative errors allow a better interpretation by comparing point-wise the absolute error with the true standard error. For instance for $M = 10$, the mean of re_k shows that — on average — the median error at training points represents 6.2% of the true standard error. The results on the 275 1'000 testing points are similar, which shows that the estimation is good both at testing and training points. Even for $M = 5$, all error measures are quite satisfactory, as well as their standard deviations. These results are also compared with $\hat{\sigma}_{S3}^2$. As expected for an homoskedastic dataset, estimate $\hat{\sigma}_{BR}^2$ based on homoskedastic assumption always results into better performance than $\hat{\sigma}_{S3}^2$ 280 which is based on heteroskedastic assumption.

The same experiment is done with $n = 100$ and $N = 5$. The results, reported in Table 1, show that all results are improved when increasing the number of data points, as expected.

4.2. Multi-dimensional case

An example on a multi-dimensional case study is now investigated. Specifically, the synthetic dataset described by Friedman in [31] is considered, with fixed inputs $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ drawn independently from uniform distribution on the interval $[0, 1]$ and outputs generated with an independent homoskedastic Gaussian noise according to

$$y(\mathbf{x}) = 10 \sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon, \quad (22)$$

285 with noise variance $\sigma_\varepsilon^2 = 0.5$. A number of $n = 500$ training points are drawn. The number of neurons is chosen by a similar cross-validation process as described above for the one-dimensional case, and fixed to $N = 91$. Ensembles are fitted and homoskedastic estimates $\hat{\sigma}_{BR}^2$ and $\hat{\sigma}_{NHo}^2$ are computed with

		Non-regularized			Regularized		
$M = 5$		BR	NHo	Grnd tr.	BR	NHo	Grnd tr.
Training set	se_k	0.273 (0.009)	0.279 (0.009)	0.255 (—)	0.250 (0.009)	0.252 (0.009)	0.222 (—)
	e_k	0.018 (0.008)	0.023 (0.009)	—	0.025 (0.009)	0.027 (0.009)	—
	re_k	0.070 (0.032)	0.089 (0.037)	—	0.112 (0.039)	0.121 (0.039)	—
Testing set	se_k	0.282 (0.010)	0.289 (0.010)	0.264 (—)	0.253 (0.009)	0.255 (0.009)	0.228 (—)
	e_k	0.018 (0.008)	0.024 (0.009)	—	0.025 (0.009)	0.027 (0.009)	—
	re_k	0.069 (0.030)	0.090 (0.034)	—	0.110 (0.039)	0.119 (0.039)	—
$M = 10$		BR	NHo	Grnd tr.	BR	NHo	Grnd tr.
Training set	se_k	0.266 (0.009)	0.269 (0.009)	0.247 (—)	0.243 (0.008)	0.244 (0.008)	0.214 (—)
	e_k	0.019 (0.008)	0.022 (0.008)	—	0.028 (0.008)	0.029 (0.008)	—
	re_k	0.078 (0.033)	0.089 (0.034)	—	0.128 (0.035)	0.132 (0.035)	—
Testing set	se_k	0.274 (0.009)	0.277 (0.009)	0.255 (—)	0.246 (0.008)	0.247 (0.008)	0.219 (—)
	e_k	0.019 (0.008)	0.023 (0.009)	—	0.028 (0.008)	0.029 (0.008)	—
	re_k	0.078 (0.033)	0.090 (0.034)	—	0.127 (0.035)	0.132 (0.035)	—
$M = 20$		BR	NHo	Grnd tr.	BR	NHo	Grnd tr.
Training set	se_k	0.263 (0.008)	0.264 (0.008)	0.243 (—)	0.239 (0.007)	0.240 (0.007)	0.210 (—)
	e_k	0.020 (0.008)	0.021 (0.008)	—	0.028 (0.007)	0.029 (0.007)	—
	re_k	0.082 (0.033)	0.087 (0.033)	—	0.134 (0.034)	0.137 (0.034)	—
Testing set	se_k	0.270 (0.008)	0.272 (0.009)	0.250 (—)	0.242 (0.007)	0.243 (0.007)	0.214 (—)
	e_k	0.020 (0.008)	0.022 (0.008)	—	0.028 (0.007)	0.029 (0.007)	—
	re_k	0.081 (0.033)	0.088 (0.033)	—	0.134 (0.034)	0.137 (0.034)	—
$M = 100$		BR	NHo	Grnd tr.	BR	NHo	Grnd tr.
Training set	se_k	0.259 (0.008)	0.260 (0.008)	0.239 (—)	0.236 (0.007)	0.236 (0.007)	0.207 (—)
	e_k	0.020 (0.008)	0.020 (0.008)	—	0.029 (0.007)	0.029 (0.007)	—
	re_k	0.084 (0.033)	0.085 (0.033)	—	0.139 (0.033)	0.139 (0.033)	—
Testing set	se_k	0.267 (0.008)	0.267 (0.008)	0.239 (—)	0.239 (0.007)	0.239 (0.007)	0.210 (—)
	e_k	0.021 (0.008)	0.021 (0.008)	—	0.029 (0.007)	0.029 (0.007)	—
	re_k	0.084 (0.033)	0.086 (0.033)	—	0.140 (0.033)	0.140 (0.033)	—

Table 2: Results of the multi-dimensional synthetic experiment with homoskedastic noise. Mean (standard deviation) of se_k , e_k and re_k .

$M = 5, 10, 20, 100$. This is repeated 1'000 times, while ground truth mean and variance are computed based on 10'000 ensembles, as in the previous experiment. The same experiment is conducted with ensembles of regularized ELM. Tikhonov factor is selected with the help of generalized cross-validation [23, 17] repeated on 1'000 dataset generations and set to $\alpha = 6 \cdot 10^{-6}$. Regularized version of $\hat{\sigma}_{BR}^2$ and $\hat{\sigma}_{NHo}^2$ are computed.

Results of the regularized and non-regularized versions of the experiment are reported in Table 2. For both, the training se_k tends to slightly overestimate the true standard deviation median over the training points. The testing se_k has an analogous behaviour. Although the testing se_k tends globally to be greater than the training se_k , the testing e_k and re_k are similar to the training e_k and re_k . This suggests that regardless of the fact that the prediction is more uncertain at testing points, the variance estimation works at testing points as well as at the training points, as in the one-dimensional experiment. Note also that the true standard deviation median decreases as M increases, as suggested by equation (5). For the non-regularized case, the bias-reduced estimate $\hat{\sigma}_{BR}^2$ is systematically better than the $\hat{\sigma}_{NHo}^2$ estimate. Recalling that $\hat{\sigma}_{BR}^2$ reduce the bias by a quantity inversely proportional to M — see section 3.2 — one observes that for e_k and re_k the improvement over $\hat{\sigma}_{NHo}^2$ is decreasing with M . The regularization mechanism increases the bias of the model while its variance

decreases, which explains the decreasing of the true standard deviation median
 310 for a given M from the non-regularized to the regularized case. Moreover, for the
 regularized case, as the bias of the variance estimation depends directly from the
 conditional bias of the model, this could explain that the regularized experiment
 yields slightly weaker results in terms of e_k and re_k . However, observe that $\hat{\sigma}_{BR}^2$
 is still better than $\hat{\sigma}_{NHo}^2$.

To illustrate the heteroskedastic case, the same experiment is conducted with
 a non-constant noise variance. The Gaussian noise $\varepsilon(\mathbf{x})$ is now depending on
 the inputs variables through its variance by

$$\sigma_\varepsilon^2(\mathbf{x}) = 0.5 + 2 \sin^2(\pi \|\mathbf{x}\|_\infty),$$

315 where $\|\cdot\|_\infty$ denotes the maximum norm. The variance estimates $\hat{\sigma}_{S3}^2$, $\hat{\sigma}_{S2}^2$,
 $\hat{\sigma}_{NHe}^2$, $\hat{\sigma}_{S1}^2$ and $\hat{\sigma}_{BR}^2$ are computed in their approximated version 1'000 times
 with $n = 1000$, $M = 5, 10, 20, 100$, and $N = 109$.

Although the results on the 5-dimensional hypercube input cannot be visu-
 alized, a small subset such as its diagonal can be plot, see Figure 2. On the left,
 320 prediction with $\pm 1.96 \hat{\sigma}_{S2}$ is displayed for one experiment, for $M = 5$. The true
 noise variance $\sigma_\varepsilon^2(\mathbf{x})$ is also reported. On the right, **averaged** results for $\hat{\sigma}_{S2}^2$,
 $\hat{\sigma}_{S1}^2$ and $\hat{\sigma}_{BR}^2$ for the 1'000 experiments are shown. Results for $\hat{\sigma}_{S3}^2$ and σ_{Nhe}^2
 are visually close to $\hat{\sigma}_{S2}^2$ and are not reported. Heteroskedastic estimates repro-
 duce fairly well the behaviour of the true variance, but $\hat{\sigma}_{S2}^2$ shows a smaller bias
 325 than $\hat{\sigma}_{S1}^2$ along the input diagonal. The homoskedastic estimates $\hat{\sigma}_{BR}^2$ clearly
 fails to reproduce a coherent behaviour of the true variance, underestimating or
 overestimating it, depending on the location.

Quantitative results are shown in Table 3. Again, the true standard deviation
 median decreases when M increases, and the training (testing) se_k tends to
 330 somewhat overestimate the true training (testing) standard deviation median.
 The homoskedastic estimate $\hat{\sigma}_{BR}^2$ no longer gives the best results because of
 the heteroskedastic nature of the data, which justify the use of heteroskedastic
 estimates. The estimate $\hat{\sigma}_{S1}^2$ — which reuse HCCME in its original form — gives
 the worst results as suspected in section 3.3. The naive heteroskedastic estimate
 335 $\hat{\sigma}_{NHe}^2$ gives in general reasonable results. However, the heteroskedastic estimates
 $\hat{\sigma}_{S2}^2$ — which was developed based on the insight given by the homoskedastic
 case in section 3.2 — allows to improve the results by a quantity decreasing
 with M , as expected. Finally, results from $\hat{\sigma}_{S3}^2$ are very close to $\hat{\sigma}_{S2}^2$.

Regularized versions of the heteroskedastic estimates were also investigated
 340 and $\hat{\sigma}_{S2}^2$ and $\hat{\sigma}_{S3}^2$ gave quite reasonable results too.

4.3. Towards confidence intervals

Although visually it is tempting to say so, there is so far no guarantee that
 $\hat{f}(x) \pm 1.96 \hat{\sigma}_{BR}(x)$ in Figure 1 (left) or $\hat{f}(x) \pm 1.96 \cdot \hat{\sigma}_{S2}(x)$ in Figure 2 (left) define
 a 95% CI for $f(x)$. Let us investigate the possibility to build (approximate) CI
 in particular cases. The distribution of

$$g(\mathbf{x}) = \frac{\hat{f}(\mathbf{x}) - f(\mathbf{x})}{\text{sd}[\hat{f}(\mathbf{x})]},$$

$M = 5$		S3	S2	NHe	S1	BR	Grnd tr.
Training set	se_k	0.279 (0.008)	0.278 (0.007)	0.285 (0.008)	0.308 (0.008)	0.283 (0.007)	0.253 (—)
	e_k	0.027 (0.005)	0.027 (0.005)	0.031 (0.006)	0.052 (0.008)	0.046 (0.004)	—
	re_k	0.109 (0.019)	0.109 (0.019)	0.125 (0.024)	0.211 (0.032)	0.178 (0.015)	—
Testing set	se_k	0.302 (0.009)	0.302 (0.009)	0.309 (0.009)	0.335 (0.010)	0.294 (0.008)	0.275 (—)
	e_k	0.029 (0.006)	0.029 (0.006)	0.035 (0.007)	0.060 (0.009)	0.043 (0.004)	—
	re_k	0.111 (0.022)	0.111 (0.021)	0.131 (0.027)	0.230 (0.034)	0.157 (0.016)	—
$M = 10$		S3	S2	NHe	S1	BR	Grnd tr.
Training set	se_k	0.273 (0.007)	0.273 (0.007)	0.276 (0.007)	0.303 (0.008)	0.277 (0.007)	0.247 (—)
	e_k	0.027 (0.005)	0.027 (0.005)	0.029 (0.005)	0.053 (0.008)	0.047 (0.004)	—
	re_k	0.111 (0.020)	0.111 (0.020)	0.120 (0.022)	0.224 (0.032)	0.187 (0.015)	—
Testing set	se_k	0.296 (0.008)	0.296 (0.008)	0.299 (0.008)	0.330 (0.010)	0.288 (0.008)	0.267 (—)
	e_k	0.030 (0.006)	0.030 (0.006)	0.032 (0.006)	0.062 (0.009)	0.045 (0.004)	—
	re_k	0.115 (0.022)	0.114 (0.022)	0.126 (0.025)	0.246 (0.034)	0.168 (0.016)	—
$M = 20$		S3	S2	NHe	S1	BR	Grnd tr.
Training set	se_k	0.270 (0.007)	0.270 (0.007)	0.272 (0.007)	0.301 (0.008)	0.274 (0.007)	0.244 (—)
	e_k	0.027 (0.005)	0.027 (0.005)	0.028 (0.005)	0.054 (0.007)	0.048 (0.004)	—
	re_k	0.113 (0.020)	0.113 (0.020)	0.118 (0.021)	0.230 (0.031)	0.192 (0.015)	—
Testing set	se_k	0.292 (0.008)	0.292 (0.008)	0.294 (0.008)	0.327 (0.009)	0.284 (0.007)	0.263 (—)
	e_k	0.030 (0.006)	0.030 (0.006)	0.031 (0.006)	0.063 (0.009)	0.046 (0.004)	—
	re_k	0.117 (0.023)	0.117 (0.023)	0.123 (0.024)	0.254 (0.033)	0.173 (0.016)	—
$M = 100$		S3	S2	NHe	S1	BR	Grnd tr.
Training set	se_k	0.268 (0.007)	0.268 (0.007)	0.268 (0.007)	0.299 (0.008)	0.272 (0.007)	0.242 (—)
	e_k	0.027 (0.005)	0.027 (0.005)	0.028 (0.005)	0.055 (0.007)	0.049 (0.004)	—
	re_k	0.115 (0.020)	0.115 (0.020)	0.116 (0.020)	0.236 (0.031)	0.196 (0.015)	—
Testing set	se_k	0.290 (0.008)	0.290 (0.008)	0.290 (0.008)	0.325 (0.009)	0.282 (0.007)	0.261 (—)
	e_k	0.030 (0.006)	0.030 (0.006)	0.030 (0.006)	0.064 (0.008)	0.046 (0.004)	—
	re_k	0.120 (0.023)	0.120 (0.023)	0.121 (0.023)	0.260 (0.033)	0.178 (0.017)	—

Table 3: Results of the multi-dimensional synthetic experiment with heteroskedastic noise. Mean (standard deviation) of se_k , e_k and re_k .

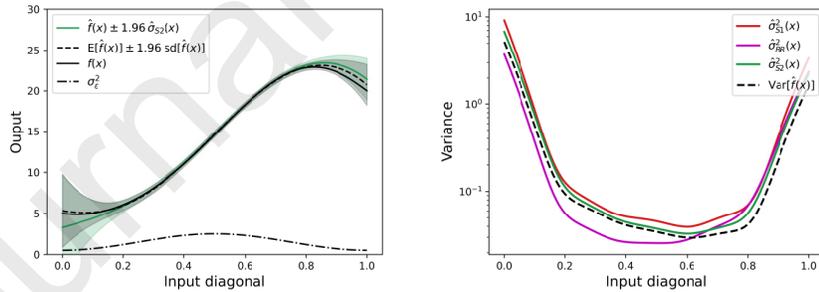


Figure 2: Multi-dimensional heteroskedastic synthetic experiment : (left) In green, a single estimation of $\hat{f}(x)$ with $M = 5$ and its estimated ± 1.96 standard-error bands based on $\hat{\sigma}_{S_2}^2$. In black dashed line, the mean of 10'000 ensembles for $M = 5$, with ± 1.96 standard-error bands. The true $f(x)$ is displayed in full black line and the noise variance in dash-dotted black line; (right) The **averaged variance estimations** for $\hat{\sigma}_{S_2}^2$ in green, $\hat{\sigma}_{S_1}^2$ in red and $\hat{\sigma}_{BR}^2$ in purple, based on 1'000 replications of the experiment. **Black dashed line indicates** the variance computed from the 10'000 ensembles, considered as ground truth. The diagonal distance from the origin on the x-axis is measured in maximum norm. **Note the logarithmic scale on the y-axis.**

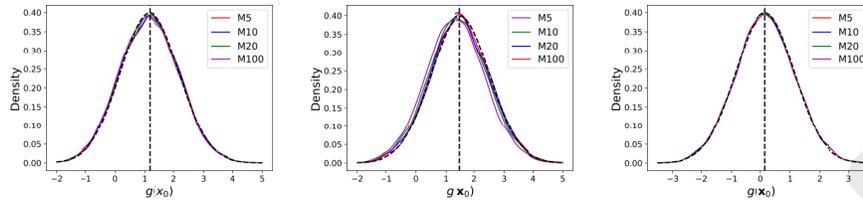


Figure 3: Kernel density estimates of $g(\mathbf{x}_0)$; (left) for the one-dimensional case, at $x_0 = \pi/4$, (middle) for the multi-dimensional case with homoskedastic noise, at $\mathbf{x}_0 = 1/2 \cdot (1, 1, 1, 1, 1)$, (right) at the same point with regularized ELM ensemble of $N = 300$ neurons. In black dashed line, Gaussian distributions with unit variance and mean (vertical dashed line) determined by the ratio between the bias and the standard deviation of $\hat{f}(\mathbf{x}_0)$

is unknown. However, for our simulated case studies, remark that it is very close to a Gaussian distribution. Kernel density estimates — based on the 10'000 replications done in previous sections — are shown in Figure 3 at some example points at which $\hat{f}(\mathbf{x})$ exhibits significant bias. Figure 3 (left) displays the distribution of $g(x_0)$ at $x_0 = \pi/4$ for the one-dimensional case. Figure 3 (middle) visualizes the distribution of $g(\mathbf{x}_0)$ at $\mathbf{x}_0 = 1/2 \cdot (1, 1, 1, 1, 1)$ for the Friedman dataset with homoskedastic noise, which corresponds to the center of the 5-dimensional hypercube and the middle point of the input diagonal. Distribution of $g(\mathbf{x}_0)$ behaves similarly for the Friedman dataset with heteroskedastic noise (not shown) and other experiments were conducted with noise from Student laws showing the same behaviour. This suggests that for ELM ensembles, $g(\mathbf{x})$ and $\hat{f}(\mathbf{x})$ may asymptotically follow a Gaussian distribution. However, dependencies exist between the components of \mathbf{z}_m , and also between the members of the ELM ensemble. Therefore, the classical central limit theorem is not directly applicable, and it seems hard to straightforwardly conclude to Gaussianity in case of large sample size n or large M . In spite of knowing if one can prove or disprove this conclusion, let us assume that the distribution of $g(\mathbf{x})$ is (asymptotically) Gaussian in the remainder of this section. As a matter of fact, $g(\mathbf{x})$ has a unit variance but it is not centred, due to the bias of $\hat{f}(\mathbf{x})$. Its mean is given by

$$\mathbb{E}[g(\mathbf{x})] = \frac{\text{Bias}[\hat{f}(\mathbf{x})]}{\text{sd}[\hat{f}(\mathbf{x})]},$$

and is reported in Figure 3 as a vertical dashed black line. Obviously, this quantity is unknown in practice but necessary to build a reliable CI for $f(\mathbf{x})$. However, if the bias of $\hat{f}(\mathbf{x})$ is negligible relatively to its variance, then $g(\mathbf{x})$ is close to centred, and approximate point-wise CI can be derived based solely on an estimation of the variance of $\hat{f}(\mathbf{x})$. That is, if $g(\mathbf{x})$ is close to centred, then the estimated ± 1.96 standard-error around $\hat{f}(\mathbf{x})$ define an approximate point-wise 95% CI for $f(\mathbf{x})$.

Figure 4 plots for some of the previous experiments the coverage probability of the approximate point-wise 95% CI, i.e. the proportion of time that the

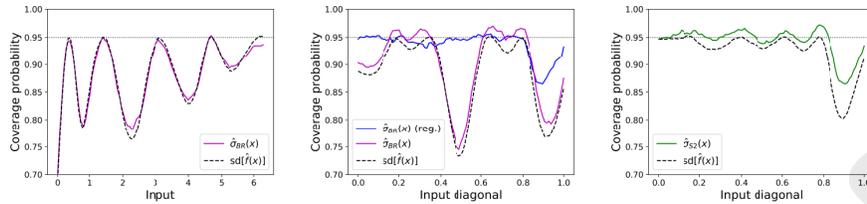


Figure 4: Coverage probabilities for the one-dimensional case with $M = 10$ (left), for the multi-dimensional case with $M = 5$, with homoskedastic noise (middle), and with heteroskedastic noise (right). The true coverage probability is fixed at 95 % (dotted line). The actual coverage probability is reported for $\hat{f}(x) \pm 1.96 \hat{\sigma}_{BR}$ (in purple) and $\hat{f}(x) \pm 1.96 \hat{\sigma}_{S2}$ (in green). It is also shown for $\hat{f}(x) \pm 1.96 sd[\hat{f}(x)]$ in black dashed line, for comparison purpose. In the homoskedastic multi-dimensional case, the actual coverage probability is also reported for $\hat{f}(x) \pm 1.96 \hat{\sigma}_{BR}$ for ensemble of $M = 5$ regularized ELM of $N = 300$ neurons (in blue).

estimated confidence interval actually contains the true $f(\mathbf{x})$ among all the 1'000 experiment repetitions. Figure 4 (left) shows the one-dimensional case for $M = 10$. The black dashed line indicates the proportion of time that $f(x)$ lies within $\hat{f}(x) \pm 1.96 sd[\hat{f}(x)]$, computed on the basis of the 10'000 simulations
 355 baseline. Observe that around each point where the bias vanishes — seen on Figure 1 (left) — this proportion comes closer to the true coverage probability of 0.95 defined by the confidence level. Conversely, for instance at $x_0 = \pi/4$ — which is near the point with the smallest variance, see Figure 1 (right) — the bias is high relatively to the variance, then $g(\mathbf{x}_0)$ is far from centred, which implies
 360 a wrong construction of the CI, leading to a bad result. Obviously, $sd[\hat{f}(x)]$ is not available in practice, and looking at the actual coverage probability of the estimated CI $\hat{f}(x) \pm 1.96 \hat{\sigma}_{BR}(x)$ is more interesting. However, note that it reproduces quite fairly the same behaviour, as expected.

The non-regularized multi-dimensional experiment for $M = 5$ is also displayed in Figure 4 for homoskedastic case, also estimated with $\hat{f}(x) \pm 1.96 \hat{\sigma}_{BR}(x)$
 365 (middle), and for heteroskedastic case, estimated with $\hat{f}(x) \pm 1.96 \hat{\sigma}_{S2}(x)$ (right). For both, the actual coverage probability is globally greater than the proportion based on the theoretical CI build with the true $sd[\hat{f}(x)]$. This is partially explained by the overestimation of the $\hat{f}(\mathbf{x})$ standard deviation — see section 4.2.
 370 In some low bias regions, this results in slightly conservative CI, i.e. the actual coverage probability is greater than the true coverage probability of 95%. Observe that around $\mathbf{x}_0 = 1/2 \cdot (1, 1, 1, 1)$ the actual coverage probability is especially bad for the homoskedastic case, while it is quite reasonable for the heteroskedastic case. This is explained by the fact that the heteroskedastic noise
 375 variance around the center of the hypercube is up to five times more than the homoskedastic variance, which implies an increase of the variance of $\hat{f}(\mathbf{x})$ and a decrease of $\mathbb{E}[g(\mathbf{x})]$ around \mathbf{x}_0 . Finally, the actual coverage probability is quite satisfying for the heteroskedastic case.

Clearly, the effectiveness of the CI estimation for $f(\mathbf{x})$ is highly dependant

		$N = 91$	Grnd tr.	$N = 300$ (reg.)	Grnd tr.
Training set	MSE	0.441 (0.030)	—	0.361 (0.025)	—
	RE	0.017 (0.001)	—	0.014 (0.001)	—
	se_k	0.273 (0.009)	0.255 (—)	0.297 (0.010)	0.294 (—)
	e_k	0.018 (0.008)	—	0.009 (0.006)	—
	re_k	0.070 (0.032)	—	0.031 (0.022)	—
Testing set	MSE	0.679 (0.037)	—	0.682 (0.041)	—
	RE	0.028 (0.002)	—	0.028 (0.002)	—
	se_k	0.282 (0.010)	0.264 (—)	0.312 (0.011)	0.307 (—)
	e_k	0.018 (0.008)	—	0.009 (0.007)	—
	re_k	0.069 (0.030)	—	0.031 (0.022)	—

Table 4: Comparison between non-regularized model with $N = 91$ neurons and regularized model with $N = 300$ and $\alpha = 10^{-6}$, on the multi-dimensional synthetic experiment with homoskedastic noise. For both model, $M = 5$ and the variance is estimated with $\hat{\sigma}_{BR}^2$. Mean (standard deviation) of MSE , se_k , e_k and re_k .

380 on the dataset at hand and significant bias of $\hat{f}(\mathbf{x})$ relatively to its variance can lead to highly permissive and bad CI for $f(\mathbf{x})$. However, it is possible to identify potential paths to overcome this problem. Firstly, note that even if the bias of $\hat{f}(\mathbf{x})$ is too important to be ignored, the estimated ± 1.96 standard-error bands around $\hat{f}(\mathbf{x})$ still provides a reliable CI for $\mathbb{E}[\hat{f}(\mathbf{x})]$. Secondly, the bias could be
385 estimated. Thirdly, a manner of reducing the bias is to smooth $\hat{f}(\mathbf{x})$ slightly less than what would be appropriate [32], for instance through regularization. For the latter, one provides here an example for the multi-dimensional case with homoskedastic noise.

Ensemble of $M = 5$ regularized ELM is trained with $N = 300$. Selecting voluntarily a bigger number of neurons increases the model complexity, then reduces the model bias. But increasing complexity also implies increasing the model variability, which puts the model in an overfitting situation that the regularization mechanism controls at the expense of the introduction of an additional bias. The general cross-validation estimate results in a Tikhonov factor of 10^{-4} which introduces too much bias. Then, $\alpha = 10^{-6}$ is empirically set to decrease the amount of smoothing hence alleviating the bias, at the expense of a higher variance. To measure predictive performance of the model, the mean squared error (MSE) and relative mean squared error (RE) are defined on the training set by

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i))^2 \quad \text{and} \quad RE = \frac{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{f}(\mathbf{x}_i))^2}{\frac{1}{n} \sum_{i=1}^n (y_i - \frac{1}{n} \sum_{i=j}^n y_j)^2},$$

390 and similar measures are defined on the testing set. Generally speaking, lower values of MSE and RE are better. A value higher than 1 for RE indicates that the model performs worse than the mean [33]. Note also that RE can be interpreted as an estimation of the ratio between the residual variance and the data variance.

395 Table 4 shows the quantitative results of the regularized model with $N = 300$ compared to the non-regularized model done previously with $N = 91$. In average

among the 1'000 experiments, the testing MSE is slightly better for $N = 91$. However, the testing RE shows that in both cases it represents 2.8% of the data variance and no significant difference is identifiable. As expected, the se_k of the true variance — and its estimation — is greater for $N = 300$. The variance is better estimated, as shown by e_k and re_k . This is likely due to the model bias reduction, which probably implies a decrease of the bias of the noise estimation, and therefore of the bias of the variance estimates — see section 3.2. Figure 3 (right) visualizes the distribution of $g(\mathbf{x}_0)$ at the center of the 5-dimensional hypercube. Comparing with the first model $N = 91$ — Figure 3 (middle) — the distribution of $g(\mathbf{x}_0)$ is far closer to a centered Gaussian. The coverage probability is also shown and compared in Figure 4 (middle), where results are considerably improved, especially at $\mathbf{x}_0 = 1/2 \cdot (1, 1, 1, 1, 1)$. Summarizing, while the CI are then globally correctly estimated, the predictive performance are almost the same.

5. Conclusion

This paper discussed variance of (regularized) ELM under general hypothesis and its estimation through small ensembles of retrained ELMs under homoskedastic and heteroskedastic hypothesis. As ELM is nothing more than a linear regression in a random feature space, analytical results can be derived by conditioning on the random input weights and biases. In particular, the variance of $\hat{f}(\mathbf{x}_0)$ knowing input data has been decomposed into additive terms, supporting the identification and the interpretation of the contribution of different variability sources. Based on these formulas, several variance estimates independent of the noise distribution were provided for homoskedastic and heteroskedastic cases, for which a Python implementation was provided. Formulas and estimate-related theoretical results are supported by numerical simulations and empirical findings. Bias-reduced estimate $\hat{\sigma}_{BR}^2$ is likely uniformly better than $\hat{\sigma}_{NH_o}^2$ in the homoskedastic case and should be preferred. In the heteroskedastic case, $\hat{\sigma}_{S_2}^2$ and $\hat{\sigma}_{S_3}^2$ are empirically shown to be better than other proposed estimates. Although these estimates are close to each other, $\hat{\sigma}_{S_2}^2$ is computationally more efficient than $\hat{\sigma}_{S_3}^2$.

The paper also showed the possibility of constructing accurate CI for $f(\mathbf{x}_0)$ and $\mathbb{E}[\hat{f}(\mathbf{x}_0)]$ despite the non-parametric, non-linear, and random nature of ELM. It provided a detailed explanation of the bias/variance contribution in CI estimation and highlighted that bias must be carefully considered to achieve satisfactory performances, especially in the regularized case which introduces significant bias. In particular, bias was traded against variance which can be estimated while preserving the predicting performance of the modelling, leading to credible uncertainty estimation. Also, as the variance estimates are distribution-free, it is reasonable to think that CI could be built with non-Gaussian noise distributional assumptions.

Several aspects of ELM uncertainty quantification still need to be investigated. From a theoretical perspective, (asymptotical) normality of ELM (ensembles) should be proved or disproved. More generally, having an analytical

440 expression for the distribution of H could be very useful to develop estimation based on a single ELM. Additionally, random matrix theory — which already provided theoretical results for ELM [34] — should be investigated in the uncertainty quantification context.

445 Practically, prediction variance estimation is straightforward by adding $\hat{\sigma}_\varepsilon^2$ to the variance estimate in the homoskedastic case, while the noise variance could be estimated in the heteroskedastic case, e.g. with a second model [8, 9]. Prediction interval can also be constructed, assuming convenient noise distribution. Future studies could also involve dependant data, e.g. by adapting heteroskedasticity and autocorrelation consistent (HAC) estimations of the full noise covariance matrix, in the temporal, spatial **or spatio-temporal** cases [13, 35, 36, 37].

6. Appendix

Proof of equation (4). Recall that W_1, \dots, W_M are i.i.d.. Reusing Eq. (1), one have

$$\begin{aligned} \mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W_1, \dots, W_M \right] &= \frac{1}{M} \sum_{m=1}^M \mathbb{E} \left[\hat{f}_m(\mathbf{x}_0) \mid W_m \right] \\ &= \frac{1}{M} \sum_{m=1}^M \mathbf{h}_m^T H_m^\alpha \mathbf{f}. \end{aligned} \quad (23)$$

The law of total expectation yields

$$\begin{aligned} \mathbb{E} \left[\hat{f}(\mathbf{x}_0) - f(\mathbf{x}_0) \right] &= \mathbb{E} \left[\mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W_1, \dots, W_M \right] \right] - f(\mathbf{x}_0) \\ &= \frac{1}{M} \sum_{m=1}^M \underbrace{\mathbb{E} \left[\mathbf{h}_m^T H_m^\alpha \mathbf{f} \right]}_{= \text{cst}} - f(\mathbf{x}_0) \\ &= \mathbb{E} \left[\mathbf{h}^T H^\alpha \mathbf{f} \right] - f(\mathbf{x}_0), \end{aligned}$$

and the bias still unchanged. Computing the first term of the law of total variance, one get

$$\begin{aligned} \text{Var} \left[\hat{f}(\mathbf{x}_0) \mid W_1, \dots, W_M \right] &= \text{Var} \left[\frac{1}{M} \sum_{m=1}^M \mathbf{h}_m^T H_m^\alpha \mathbf{y} \mid W_1, \dots, W_M \right] \\ &= \frac{1}{M^2} \sum_{m,l=1}^M \text{Cov} \left[\mathbf{h}_m^T H_m^\alpha \mathbf{y}, \mathbf{h}_l^T H_l^\alpha \mathbf{y} \mid W_m, W_l \right] \\ &= \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{h}_m^T H_m^\alpha \text{Var}[\mathbf{y}] H_l^{\alpha T} \mathbf{h}_l \\ &= \frac{1}{M^2} \sum_{m,l=1}^M \mathbf{h}_m^T H_m^\alpha \Sigma H_l^{\alpha T} \mathbf{h}_l, \end{aligned}$$

which implies

$$\begin{aligned}
 \mathbb{E} \left[\text{Var} \left[\hat{f}(\mathbf{x}_0) \mid W_1, \dots, W_M \right] \right] &= \frac{1}{M^2} \sum_{m=1}^M \underbrace{\mathbb{E} \left[\mathbf{h}_m^T H_m^\alpha \Sigma H_m^{\alpha T} \mathbf{h}_m \right]}_{= \text{cst}} \\
 &+ \frac{1}{M^2} \sum_{m \neq l} \mathbb{E} \left[\mathbf{h}_m^T H_m^\alpha \right] \Sigma \mathbb{E} \left[H_l^{\alpha T} \mathbf{h}_l \right] \\
 &= \frac{1}{M} \mathbb{E} \left[\mathbf{h}^T H^\alpha \Sigma H^{\alpha T} \mathbf{h} \right] \\
 &+ \frac{M-1}{M} \mathbb{E} \left[\mathbf{h}^T H^\alpha \right] \Sigma \mathbb{E} \left[H^{\alpha T} \mathbf{h} \right],
 \end{aligned} \tag{24}$$

where one used the fact that the input weights and biases are drawn independently. Using (23), the second term of the law of total variance is

$$\begin{aligned}
 \text{Var} \left[\mathbb{E} \left[\hat{f}(\mathbf{x}_0) \mid W_1, \dots, W_M \right] \right] &= \frac{1}{M^2} \sum_{m,l=1}^M \text{Cov} \left[\mathbf{h}_m^T H_m^\alpha \mathbf{f}, \mathbf{h}_l^T H_l^\alpha \mathbf{f} \right] \\
 &= \frac{1}{M^2} \sum_{m=1}^M \underbrace{\text{Var} \left[\mathbf{h}_m^T H_m^\alpha \mathbf{f} \right]}_{= \text{cst}} \\
 &= \frac{1}{M} \text{Var} \left[\mathbf{h}^T H^\alpha \mathbf{f} \right],
 \end{aligned} \tag{25}$$

as $\text{Cov} \left[\mathbf{h}_m^T H_m^\alpha \mathbf{f}, \mathbf{h}_l^T H_l^\alpha \mathbf{f} \right]$ vanishes when $m \neq l$, thanks again to the i.i.d. assumption on the weights. By summing Eq. (24) and Eq. (25), the result is obtained. \square

Proof of equation (19). First, note that for all $m, l, k = 1, \dots, M$,

$$\begin{aligned}
 \text{Cov} \left[\mathbf{r}_m^T \mathbf{r}_m, \mathbf{z}_k^T \mathbf{z}_l \right] &= \text{Cov} \left[\mathbb{E} \left[\mathbf{r}_m^T \mathbf{r}_m \mid W_m \right], \mathbb{E} \left[\mathbf{z}_k^T \mathbf{z}_l \mid W_l, W_k \right] \right] \\
 &+ \mathbb{E} \left[\text{Cov} \left[\mathbf{r}_m^T \mathbf{r}_m, \mathbf{z}_k^T \mathbf{z}_l \mid W_m, W_l, W_k \right] \right] \\
 &= \text{Cov} \left[\mathbf{f}^T \mathbf{b}_{w,m}, \mathbf{z}_k^T \mathbf{z}_l \right],
 \end{aligned} \tag{26}$$

where for the first equality uses the law of total covariance, and the second equality uses equation (11), the fact that expectation has no effect on a constant and that covariance between a random variable and a constant is null. Also, using the covariance definition and the independence of weights between models, one has for all $k \neq l$,

$$\begin{aligned}
 \text{Cov} \left[\mathbf{f}^T \mathbf{b}_{w,k}, \mathbf{z}_k^T \mathbf{z}_l \right] &= \mathbb{E} \left[\mathbf{f}^T \mathbf{b}_{w,k} \mathbf{z}_k^T \mathbf{z}_l \right] - \mathbb{E} \left[\mathbf{f}^T \mathbf{b}_{w,k} \right] \mathbb{E} \left[\mathbf{z}_k^T \mathbf{z}_l \right] \\
 &= \mathbb{E} \left[\mathbf{f}^T \mathbf{b}_{w,k} \mathbf{z}_k^T \right] \boldsymbol{\mu} - \mathbb{E} \left[\mathbf{f}^T \mathbf{b}_{w,k} \right] \mathbb{E} \left[\mathbf{z}_k^T \right] \boldsymbol{\mu} \\
 &= \text{Cov} \left[\mathbf{f}^T \mathbf{b}_{w,k}, \mathbf{z}_k^T \boldsymbol{\mu} \right],
 \end{aligned} \tag{27}$$

Then, using equations (10), (12), (26) and (27), one obtains

$$\begin{aligned}
\text{Cov} \left[\hat{\sigma}_\varepsilon^2, \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] \right] &= \frac{1}{M^2(M-1)(n-N)} \sum_{m=1}^M \sum_{k \neq l} \text{Cov} [\mathbf{r}_m^T \mathbf{r}_m, \mathbf{z}_k^T \mathbf{z}_l] \\
&= \frac{1}{M^2(M-1)(n-N)} \sum_{m=1}^M \sum_{k \neq l} \text{Cov} [\mathbf{f}^T \mathbf{b}_{w,m}, \mathbf{z}_k^T \mathbf{z}_l] \\
&= \frac{2}{M^2(M-1)(n-N)} \sum_{k \neq l} \text{Cov} [\mathbf{f}^T \mathbf{b}_{w,k}, \mathbf{z}_k^T \mathbf{z}_l] \\
&= \frac{2}{M^2(M-1)(n-N)} \sum_{k \neq l} \text{Cov} [\mathbf{f}^T \mathbf{b}_{w,k}, \mathbf{z}_k^T \boldsymbol{\mu}] \\
&= \frac{2}{M(n-N)} \text{Cov} [\mathbf{f}^T \mathbf{b}_w, \mathbf{z}^T \boldsymbol{\mu}],
\end{aligned}$$

where covariances vanish for $m \neq k, l$ in the third equality. By using the Cauchy-Schwarz inequality on the last equation, one gets

$$\left| \text{Cov} \left[\hat{\sigma}_\varepsilon^2, \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}} - \frac{1}{M} \text{Tr}[\hat{Q}] \right] \right| \leq \frac{2}{M(n-N)} \sqrt{\text{Var} [\mathbf{f}^T \mathbf{b}_w] \text{Var} [\mathbf{z}^T \boldsymbol{\mu}]}$$

455 from which equation (19) is obtained. □

Software Availability

UncELMe — Uncertainty quantification of Extreme Learning Machine ensemble — is a Python package proposed on PyPI and GitHub (<https://github.com/fguignard/UncELMe>).
460 It allows interested users to compute all variance estimates for Extreme Learning Machine ensemble discussed in the present paper. It is built within the scikit learn estimator framework, which enable the use of all convenient functionalities of scikit-learn [38]. Noise estimation are also returned to enable building of prediction intervals.

465 Author Contributions

F.G. conceived the main conceptual ideas, conduct investigations, developed the theoretical formalism and the methodology, performed the calculations, interpreted the computational results, wrote the original draft, and developed the Python software. M.K. carried out the supervision, project administration and funding acquisition. F.G., F.A. and M.K. discussed the results, provided critical
470 feedback, commented, reviewed and edited the original manuscript, corrected the final version of the paper, and gave final approval for publication.

The original idea of this article stems in a conference paper [24] presented at the 28th European Symposium on Artificial Neural Networks, Computational

475 Intelligence and Machine Learning (ESANN 2020) in Bruges, Belgium, from 2 to
4 October 2020. More precisely, in [24] equations (3) and (4) and naive estimates
were presented for the non-regularized case only, with very few justifications and
without any details. All other work presented in this paper is the result of an
original research.

480 Acknowledgments

This work is supported by the National Research Programme 75 “Big Data”
(PNR75) of the Swiss National Science Foundation (SNSF), project no. 167285.
The authors are grateful to Prof. Dr. David Ginsbourger from Idiap Research
Institute and University of Bern, Switzerland, for providing relevant insights on
485 probability calculus. They also thank Dr. Mohamed Laib, Dr. Sylvain Robert
and Dr. Jean Golay for the profitable discussions.

Declaration of interests

The authors declare that they have no known competing financial interests or
personal relationships that could have appeared to influence the work reported
490 in this paper.

References

- [1] T. Heskes, Practical confidence and prediction intervals, in: M. C. Mozer, M. I. Jordan, T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9*, MIT Press, 1997, pp. 176–182.
495 URL <http://papers.nips.cc/paper/1306-practical-confidence-and-prediction-intervals.pdf>
- [2] R. Tibshirani, A comparison of some error estimates for neural network models, *Neural Computation* 8 (1) (1996) 152–163. [arXiv:https://doi.org/10.1162/neco.1996.8.1.152](https://doi.org/10.1162/neco.1996.8.1.152), doi:10.1162/neco.1996.8.1.152.
500 URL <https://doi.org/10.1162/neco.1996.8.1.152>
- [3] R. Dybowski, S. J. Roberts, Confidence intervals and prediction intervals for feed-forward neural networks, *Clinical applications of artificial neural networks* (2001) 298–326.
- [4] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, Vol. 2, 2004, pp. 985–990 vol.2. doi:10.1109/IJCNN.2004.1380068.
- [5] C. Chatfield, Model uncertainty, data mining and statistical inference, *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 158 (3) (1995) 419–444.
510

- [6] E. Soria-Olivas, J. Gomez-Sanchis, J. D. Martin, J. Vila-Frances, M. Martinez, J. R. Magdalena, A. J. Serrano, Belm: Bayesian extreme learning machine, *IEEE Transactions on Neural Networks* 22 (3) (2011) 505–509. doi:10.1109/TNN.2010.2103956.
- 515 [7] K. Ning, M. Liu, M. Dong, A new robust elm method based on a bayesian framework with heavy-tailed distribution and weighted likelihood function, *Neurocomputing* 149 (2015) 891–903.
- [8] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, K. P. Wong, Probabilistic forecasting of wind power generation using extreme learning machine, *IEEE Transactions on Power Systems* 29 (3) (2014) 1033–1044. doi:10.1109/TPWRS.2013.2287871.
- 520 [9] A. Akusok, Y. Miche, K.-M. Björk, A. Lendasse, Per-sample prediction intervals for extreme learning machines, *International Journal of Machine Learning and Cybernetics* 10 (5) (2019) 991–1001. doi:10.1007/s13042-017-0777-2.
525 URL <https://doi.org/10.1007/s13042-017-0777-2>
- [10] M. Leuenberger, M. Kanevski, Extreme learning machines for spatial environmental data, *Computers & Geosciences* 85 (2015) 64–73.
- [11] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: A review, *Neural Networks* 61 (2015) 32–48.
- 530 [12] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, *Neurocomputing* 70 (1-3) (2006) 489–501.
- [13] R. Davidson, J. G. MacKinnon, et al., *Econometric theory and methods*, Vol. 5, Oxford University Press New York, 2004.
- 535 [14] A. C. Davison, *Statistical Models*, Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press, 2003. doi:10.1017/CBO9780511815850.
- [15] W. Deng, Q. Zheng, L. Chen, Regularized extreme learning machine, in: 2009 IEEE symposium on computational intelligence and data mining, *IEEE*, 2009, pp. 389–395.
- 540 [16] A. Lendasse, A. Akusok, O. Simula, F. Corona, M. van Heeswijk, E. Eirola, Y. Miche, Extreme learning machine: A robust modeling technique? yes!, in: *International Work-Conference on Artificial Neural Networks*, Springer, 2013, pp. 17–35.
- 545 [17] W. W. Piegorsch, *Statistical data analytics: Foundations for data mining, informatics, and knowledge discovery*, John Wiley & Sons, 2015.
- [18] S. Boyd, L. Vandenberghe, *Introduction to applied linear algebra: vectors, matrices, and least squares*, Cambridge university press, 2018.

- [19] N. Liu, H. Wang, Ensemble based extreme learning machine, *IEEE Signal Processing Letters* 17 (8) (2010) 754–757. 550
- [20] A. M. Mathai, S. B. Provost, *Quadratic forms in random variables: theory and applications*, Dekker, 1992.
- [21] G. A. Seber, *Multivariate observations*, Vol. 252, John Wiley & Sons, 2009.
- [22] T. J. Hastie, R. J. Tibshirani, *Generalized additive models*, Vol. 43, CRC press, 1990. 555
- [23] G. H. Golub, M. Heath, G. Wahba, Generalized cross-validation as a method for choosing a good ridge parameter, *Technometrics* 21 (2) (1979) 215–223.
- [24] F. Guignard, M. Laib, M. Kanevski, Model variance for extreme learning machine, *ESANN*, 2020. 560
- [25] J. G. MacKinnon, H. White, Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties, *Journal of econometrics* 29 (3) (1985) 305–325.
- [26] H. Nyquist, Applications of the jackknife procedure in ridge regression, *Computational Statistics & Data Analysis* 6 (2) (1988) 177–183. 565
- [27] H. White, A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity, *Econometrica: journal of the Econometric Society* (1980) 817–838.
- [28] D. V. Hinkley, Jackknifing in unbalanced situations, *Technometrics* 19 (3) (1977) 285–292. 570
- [29] S. D. Horn, R. A. Horn, D. B. Duncan, Estimating heteroscedastic variances in linear models, *Journal of the American Statistical Association* 70 (350) (1975) 380–385.
- [30] F. Cribari-Neto, Asymptotic inference under heteroskedasticity of unknown form, *Computational Statistics & Data Analysis* 45 (2) (2004) 215–233. 575
- [31] J. H. Friedman, Multivariate adaptive regression splines, *Ann. Statist.* 19 (1) (1991) 1–67. doi:10.1214/aos/1176347963.
URL <https://doi.org/10.1214/aos/1176347963>
- [32] P. Hall, Effect of bias estimation on coverage accuracy of bootstrap confidence intervals for a probability density, *The Annals of Statistics* (1992) 675–694. 580
- [33] J. Golay, M. Leuenberger, M. Kanevski, Feature selection for regression problems based on the morisita estimator of intrinsic dimension, *Pattern Recognition* 70 (2017) 126–138.

- 585 [34] C. Louart, Z. Liao, R. Couillet, et al., A random matrix approach to neural networks, *The Annals of Applied Probability* 28 (2) (2018) 1190–1248.
- [35] W. K. Newey, K. D. West, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, Tech. rep., National Bureau of Economic Research (1986).
- 590 [36] H. H. Kelejian, I. R. Prucha, Hac estimation in a spatial framework, *Journal of Econometrics* 140 (1) (2007) 131–154.
- [37] F. Amato, F. Guignard, S. Robert, M. Kanevski, A novel framework for spatio-temporal prediction of environmental data using deep learning, *Scientific Reports* 10 (1) (2020) 1–11.
- 595 [38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, *the Journal of machine Learning research* 12 (2011) 2825–2830.