

Article

# Numerical Non-Linear Modelling Algorithm Using Radial Kernels on Local Mesh Support

Francisco José Navarro-González <sup>1</sup>, Yolanda Villacampa <sup>1,\*</sup>, Mónica Cortés-Molina <sup>1</sup> and Salvador Ivorra <sup>2</sup>

<sup>1</sup> Department of Applied Mathematics, University of Alicante, 03690 Alicante, Spain; francisco.navarro@ua.es (F.J.N.-G.); monica.cortes@ua.es (M.C.-M.)

<sup>2</sup> Department of Civil Engineering, University of Alicante, 03690 Alicante, Spain; sivorra@ua.es

\* Correspondence: villacampa@ua.es

Received: 24 August 2020; Accepted: 15 September 2020; Published: 17 September 2020



**Abstract:** Estimation problems are frequent in several fields such as engineering, economics, and physics, etc. Linear and non-linear regression are powerful techniques based on optimizing an error defined over a dataset. Although they have a strong theoretical background, the need of supposing an analytical expression sometimes makes them impractical. Consequently, a group of other approaches and methodologies are available, from neural networks to random forest, etc. This work presents a new methodology to increase the number of available numerical techniques and corresponds to a natural evolution of the previous algorithms for regression based on finite elements developed by the authors improving the computational behavior and allowing the study of problems with a greater number of points. It possesses an interesting characteristic: Its direct and clear geometrical meaning. The modelling problem is presented from the point of view of the statistical analysis of the data noise considered as a random field. The goodness of fit of the generated models has been tested and compared with some other methodologies validating the results with some experimental campaigns obtained from bibliography in the engineering field, showing good approximation. In addition, a small variation on the data estimation algorithm allows studying overfitting in a model, that it is a problematic fact when numerical methods are used to model experimental values.

**Keywords:** numerical modelling; algorithm; radial kernels; local mesh support; non-linear models

## 1. Introduction

Obtaining a numerical or analytical representation of a given unknown relationship is a frequent problem in the study and modelling of systems, with application in a wide variety of fields ranging from experimental and social sciences to engineering. In addition, in most cases, the only information available is a set of experimental data from the variables that define the relationship. In other words, without loss of generality, given

$$y = f(x^1, \dots, x^d) \quad (1)$$

with

$$\vec{x} = (x^1, \dots, x^d) \in \Omega \subset \mathbb{R}^d \quad (2)$$

usually only one set of experimental data is known for the variables involved in the relationship (1):

$$\left\{ (x_{[k]}^1, x_{[k]}^2, x_{[k]}^3, \dots, x_{[k]}^d) \right\}_{k=1,2,\dots,p} \quad (3)$$

These values are usually affected by several factors that induce variations between the real and the observed quantities. That is, for any  $k$ :

$$y_{[k]} = f(x_{[k]}^1 + \varepsilon_{[k]}^1, x_{[k]}^2 + \varepsilon_{[k]}^2, \dots, x_{[k]}^d + \varepsilon_{[k]}^d) + \epsilon_{[k]} = f(\vec{x}_{[k]} + \vec{\varepsilon}_{[k]}) + \epsilon_{[k]}. \tag{4}$$

If  $\Omega$  is a bounded domain of  $\mathbb{R}^d$ , and  $f \in C^1$ , developing the function  $f(\vec{x})$  in  $\vec{x}_{[k]} = (x_{[k]}^1, x_{[k]}^2, x_{[k]}^3, \dots, x_{[k]}^d)$ , from (4):

$$y_{[k]} = f(x_{[k]}^1, x_{[k]}^2, \dots, x_{[k]}^d) + \sum_i \frac{\partial f}{\partial x^i}(\xi_{[k]}^1, \xi_{[k]}^2, \dots, \xi_{[k]}^d) \cdot \varepsilon_{[k]}^i + \epsilon_{[k]} \tag{5}$$

where  $\xi_{[k]} = (\xi_{[k]}^1, \xi_{[k]}^2, \dots, \xi_{[k]}^d) = \vec{x}_{[k]} + \lambda \cdot \vec{\varepsilon}_{[k]}$ , with  $\lambda \in [0, 1]$ .

Defining a new value  $e_{[k]}$  that includes all perturbations:

$$e_{[k]} = \sum_i \frac{\partial f}{\partial x^i}(\xi_{[k]}^1, \xi_{[k]}^2, \dots, \xi_{[k]}^d) \cdot \varepsilon_{[k]}^i + \epsilon_{[k]} \tag{6}$$

Therefore,

$$y_{[k]} = f(x_{[k]}^1, x_{[k]}^2, \dots, x_{[k]}^d) + e_{[k]} \tag{7}$$

Expression (7) is frequently assumed implying that  $y$  is the only variable affected by the error, while the variables  $\{x^i\}_{i=1}^d$  correspond to the exact values. It is also frequent to include some considerations on the distribution of probability that generates the values of the perturbation, especially the assumption of normality.

Sometimes the form of the relations is known as occurs in linear regression where the equation corresponding to the studied system is:

$$y = \alpha + \lambda_1 x^1 + \lambda_2 x^2 + \dots + \lambda_d x^d \tag{8}$$

The problem to determine the values of lambda parameters, that give the best fit between the linear expression and the data, is now reduced minimizing a previously defined error function.

In real cases, the effects of non-linearity dominate the dynamics of the system and far from the equilibrium points linear relations as in (8) are not useful. In the cases when the relation is not linear but known:

$$y = z(x^1, x^2, \dots, x^d, \dots, \lambda_1, \lambda_2, \dots, \lambda_Q) \tag{9}$$

The problem consists in the calculation of the unknown parameters  $(\lambda_1, \lambda_2, \dots, \lambda_Q)$  using the data.

In (8) and (9), determining the values of the parameters of the model can be done by minimizing an error function defined over the parameters. One common method is to consider the mean square error (MSE) of the estimated values on the experimental points.

$$e(\lambda_1, \lambda_2, \dots, \lambda_Q) = \sum_{k=1}^p (y_{[k]} - z(x_{[k]}^1, \dots, x_{[k]}^d, \lambda_1, \lambda_2, \dots, \lambda_Q))^2 \tag{10}$$

However, in real practice, the expression for the relationship is unknown and the problem is to obtain an analytical expression or a numerical approximation to the relation (1).

The finite element method (FEM) is a numerical method used initially for solving structural problems in civil and aeronautical engineering that is applied in a wide variety of fields, see [1,2]. This method allows finding approximate solutions to differential equations and boundary problems in partial differential equations.

Given a differential equation defined through a differential operator,  $D$ :

$$D(f) = v \tag{11}$$

where  $f, v \in V$  with  $V$  a function space the finite elements method replaces  $V$  by a finite dimensional subspace  $V_h \subset V$  whose elements are continuous piecewise polynomial functions of degree  $K$  (this is a Sobolev Space). In addition, the domain of the problem,  $\Omega$  is divided in  $N_e$  parts called elements  $\Omega_i$ .

$$\Omega = \cup_{i=1}^{N_e} \Omega_i \tag{12}$$

The original Equation (11) changes now to:

$$D(f_h) = v_h \tag{13}$$

where  $f_h, v_h \in V_h$ .

If  $\dim V_h = Q$ , taking a basis of  $V_h$ :

$$V_h = \langle \varphi_1(x), \varphi_2(x), \dots, \varphi_Q(x) \rangle \tag{14}$$

the solution can be approximated as:

$$f_h(x) = \sum_{i=1}^Q u_i \cdot \varphi_i(x) \tag{15}$$

The values of  $u_i$  must be determined to get a good approximation for the solution to the differential equation.

The basis (14) is selected considering the behaviour of their functions in a set of  $Q$  points called nodes  $(\zeta_1, \zeta_2, \dots, \zeta_Q)$  that satisfy the following conditions:

$$\varphi_i(\zeta_j) = \delta_{ij} \tag{16}$$

In addition, an important property is that  $\varphi_i(x) = 0$  if the node  $\zeta_i$  does not form part of the element containing the point  $x$ .

The functions of the basis are named shape functions and they are used to interpolate in points different from the nodes.

The selection of the nodes is a crucial point for the precision of the results. The operation that construct the set of nodes and elements is called meshing and is determined by the set of elements and nodes  $(\{\zeta_r\}_{r=1}^Q, \{\Omega_i\}_{i=1}^{N_e})$ . The resulting mesh can be locally characterized by a parameter related with the size of the corresponding local element and the error of the approximated solution in the element.

The authors in previous papers, using different methodologies, have studied regression methods based on the FEM. In [3], the regression algorithm is based on a least squares approach comparing experimental data and the model generated using FEM. In [4], the model is built from a physical analogy that solves some problems on the linear system. One approach more on the line of the original FEM can be seen in [5] and applied by an example in [6,7]. Although different modifications of the initial algorithm have been developed, see [8] in order to improve the computing speeds for the regression model, the basic methodology can be resumed as follows.

The numerical regression model for the relation consists of the set formed by the nodal values  $\{u_i\}$ , which is called a representation model for the system. Then, the value of the relationship in any point can be estimated using the expression:

$$f_h(x) = \sum_{i=1}^Q u_i \varphi_i(x) \tag{17}$$

By (16), the sum is restricted to the nodes that determine the element which contains point  $x$ . For simplicity in the subsequent calculations the authors have normalised the data in the domain,  $[0, 1]^d$ , and used a mesh formed by regular hyper-cubic elements with edge length  $h$ . Introducing the complexity  $c$  as the number of elements in each dimension,  $h = 1/c$ , the total number of elements is  $c^d$  and the total number of nodes is  $(c + 1)^d$ .

The key point in FEM-based methods is the determination of the values  $\{u_i\}$  considering the definition of an error function and its minimization. That problem is equivalent to solve a linear system of size,  $(c + 1)^d$ . The results and precision have been checked with good performance for low dimensions, but there is a practical problem when the dimension of the problem becomes greater because of the order of the linear system solution's algorithms.

The problem with this approach is the conversion of a local problem, which is the determination of the function value in a point  $x_o$ , into a global question solving a system to obtain the  $(c + 1)^d$  nodal values.

As a conclusion, the direct geometric interpretation of FEM regression methods makes them very interesting compared to other approaches. The techniques related with finite elements are equivalent to the determination of the tangent subspace of the manifold defined by  $y = f(x)$  at the objective point  $x_0$ . The natural evolution of the methods previously considered in the introduction is the restriction of the calculations to the boundary of  $x_0$ , determined by the underlying FEM methodology. The only requirement is the determination of estimations on nodal points. These estimations can be as simple as the k-nearest neighbor (k-NN). However, k-NN presents the problem of determining the optimal value of k, to include as much representative values as possible, minimizing the influence of outliers. This can be solved by introducing the distance-based nearest neighbor.

To do this, the proposed algorithm is based on radial basis functions (and radial basis function networks) and the results on random fields, given that the regression problem on experimental data is equivalent to removing the influence of this stochastic term over the samples generated from the relationship under study.

After the introduction in Section 1, the paper is organised as follows. Section 2.1 considers the main characteristics and use of radial basis functions in problems of interpolation and regression, while Section 2.2 presents an introduction and some results on random fields. In Section 3, the method proposed by the authors is developed and applied to different problems. Section 4 corresponds to the conclusions and analysis of future investigations.

## 2. Materials and Methods

### 2.1. Radial Basis Functions

Radial functions are functions, which depend only on the distance between their arguments:

$$\theta(x, y) = \Phi(\|x - y\|) \tag{18}$$

These kinds of functions can be used as a basis of functional space (radial basis function), see [9], introducing an interpolation method for any function  $f$  on a set of pairs composed by points called centres and the values of the function on that points,  $\{(c_i, f_i)\}_{i=1}^P$  given by:

$$\hat{f}(x) = \sum_{i=1}^P \alpha_i \Phi(\|x - c_i\|) \tag{19}$$

The values of the alpha-coefficients are calculated imposing P conditions given by:

$$\hat{f}(c_i) = f_i \tag{20}$$

This expression can be used to approximate the solution of a differential equation, using by example a Galerking approach [10]. The determination of centres is a central part of the problem to consider,

because they are not fixed points as in other methods as finite elements, finite boundaries, etc., so these techniques receive the name of meshless methods. Several works have studied these approaches for different fields (fluid dynamics in [11]) or solving problems related with the corresponding systems as in [12,13].

Another frequent use of radial functions is in the problem of functional where an unknown function  $f(x)$  must be approximated from a set of pairs  $\{(x_k, y_k)\}_{k=1}^P$  provided by discrete measurements, where  $x$  corresponds to a vector and  $y$  are values related to the value of the function in the  $x$ -points. In addition, usually there exists errors between  $y$  and the value of  $f(x)$ , that is  $y_k = f(x_k) + e_k$ .

The direct use of radial basics function (RBF) as an interpolation method is not adequate here because of the need of exact interpolation on the available points. For solving this issue, some neural networks models are developed [14] with the addition of radial functions [15,16] because they are faster compared to the classical sigmoid neurons. This better behaviour is due to the improvements of the algorithm [17] or the determinations of radial centres [18,19] because it is capable of obtaining arbitrarily good approximations to the functions [20,21]. An additional advantage of these radial basis function networks (RBFN) is the more transparent interpretation of the results compared to the case of multilayer perceptron (MLP) with the non-linearities associated with the sigmoidal functions, [19,20].

Compared with other techniques there are different assets depending on the problem, from very good results [22] to other studies affirming its inferior behaviour [23]. Other authors have also cited the need of a greater number of points to obtain similar accuracies to sigmoid networks [20].

RBFN and variations on them have been used widely to study time series [24]; signal processing [25]; data fitting [26]; system modelling [16,27,28]; system control [29]; system identification [30]; faults analysis [31]; computer vision [32–34], and speak recognition [35].

In a neural network with  $n$  neurons, the output is:

$$f_o(x) = \sum_{i=1}^n f_i R_i(x) \tag{21}$$

Following [24], a necessary and sufficient condition for a function to act as an activation function in a RBFN is not to be an even polynomial. A typical but not the only, see [36], selection for these radial functions is a set of Gaussian functions given by:

$$R_i(x) = e^{-\left(\frac{\|x-c_i\|}{\sigma_i}\right)^2} \tag{22}$$

Figure 1 represents a diagram of this kind of network.

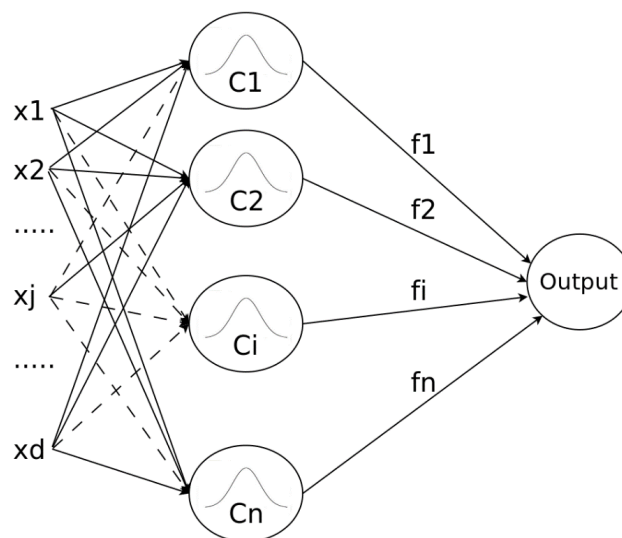


Figure 1. Radial basis function network as a set of Gaussian functions.

The use of these kinds of functions was initially applied in the study of some structures of the cerebral cortex [37].

The output is obtained from the parameters  $\{c_i, \sigma_i, f_i\}_{i=1}^n$  and several algorithms have been developed to obtain an optimum selection of those parameters (learning process). Some of them use a preliminary algorithm to determine the centres  $c_i$  (genetic algorithms [38,39] or classification trees [40] and the widths  $\sigma_i$  of each element of the network [19,31]. However, widths are not usually considered as important as the rest of the parameters, so some authors have focused on the development of algorithms to optimize them [41]. Some more recent papers have used a variety of algorithms to improve the RBFN learning capabilities using hybrid approaches [42], including techniques to reduce the dimensionality as the principal component analysis [43]. Between the modifications of the basic version of RBFN the most developed are the addition of a linear term to improve the behaviour on the regions that are out of the cloud of available points [26]. In addition, the use of a normalising operation defines the normalised radial function neural networks (NRBFN) [44,45], where the network output has the expression:

$$f_o(x) = \frac{\sum_{i=1}^n f_i R_i(x)}{\sum_{i=1}^n R_i(x)} \tag{23}$$

NRBFN present some good properties as the partition of unity, covering the whole input space and greater stability with respect to the selection of the network parameters [46].

The use of NRBFN has increased in the last times involving problems as electricity price forecasting [47], non-stationary time series [48], system control [49], system modelling [46], magnetic fields [50], and integral equations [51].

For a general study of the families of models related to (23), papers [52,53] can be of interest.

### 2.2. Random Fields

Given a domain  $\Omega$ , a random field defined on it is a collection of random variables defined on each point of  $\Omega$ .

Random fields are widely used in problems where local correlations exist on data perturbations as speech recognition, computer vision [54,55], statistical analysis of MEG images [56], and magnetic resonances imaging [57]. Some of these treatments are related with the problem of the determination of a threshold value, which has been generalized, as the determination of the frontier in a binary classification from a set of known points [58]. Additional applications are referenced in [59], etc., including particle physics, agriculture, economics, etc.

An interesting further step is the use of random fields and neural networks as considered in [60].

An important part of the investigations is devoted to specific types of random fields: Gaussian random fields, see [61], Markovian random fields, and conditional random fields used in structured learning problems where relationships exist between the regressed variables in different spatial-temporal points, by example, in the case of temporal data [62].

As pointed in [63], some of the considered problems can be seen as an incomplete data problem where part of the relevant data is hidden and must be estimated. Frequently, this estimation is made using maximum-likelihood related algorithms or using the mean field theory approach.

Following [64], let us consider a set of labelled points  $L \times \mathbb{R} = \{(x_1^1, \dots, x_1^d, y_1), \dots, (x_l^1, \dots, x_l^d, y_l)\}$  and a set of unlabeled points  $U = \{(x_1^1, \dots, x_1^d), \dots, (x_u^1, \dots, x_u^d)\}$ . The considered problem is to determine a real function defined on the set  $V = \{(x_1^1, \dots, x_1^d), \dots, (x_l^1, \dots, x_l^d), (x_{l+1}^1, \dots, x_{l+1}^d), \dots, (x_u^1, \dots, x_u^d)\}$ ,  $f : V \rightarrow \mathbb{R}$  that will be used to assign labels to the points in U, under the condition  $f(x_i) = y_i$  for  $x_i \in L$ . The introduction of a weight function

$$w_{ij} = \exp\left(-\sum_{k=1}^d \frac{(x_i^k - x_j^k)^2}{\sigma_k^2}\right) \tag{24}$$

to preserve ‘local smoothness’ and the definition of an energy function

$$E(f) = \frac{1}{2} \sum_{i,j} w_{ij} (f(x_i) - f(x_j))^2 \tag{25}$$

gives as a result the harmonic property for the minimum energy function calculated from  $E(f)$ , that is, the value of  $f$  at any unlabeled point can be written as an average of the rest of the points

$$f(x_j) = \frac{1}{\delta_j} \cdot \sum_{i \neq j} w_{ij} \cdot f(x_i) \tag{26}$$

In [65], the problem of calculating the regression of a random field from a set of observations is considered. Under the conditions of a finite domain and distribution of the points over that domain, the authors suggest considering the problem as a continuous case.

Other studies of local regression of random fields are considered in [66], where given a realization of a random field  $\{(x_i, y_i) : i \in \mathbb{Z}^N\}$  the problem is the determination of the spatial regression function given by  $f : x \mapsto f(x) := E[y_i | x_i = x]$ . An approach to the nonparametric regression can be seen in [67,68], where the regression problem is determining the function  $f$  defined on a lattice domain and given by the expression  $y_i = f(x_i) + e_i$ . The estimation expression for the function  $f$  on the domain  $[0, 1]^d$  is related with the expression:

$$f_N(x) = \frac{\sum_i y_i \cdot K\left(\frac{x-x_i}{h_N}\right)}{\sum_i K\left(\frac{x-x_i}{h_N}\right)} \tag{27}$$

where  $K$  is a kernel function and  $N$  is a parameter that determines the number of points in the lattice and the value of the parameter  $h_N$ .

Under some general conditions, the difference between the expected value of  $f_N(x)$  and the value of the function,  $|E[f_N(x)] - f(x)|$  is bounded by the value of  $h_N$ . Moreover, when  $h_N$  is fixed as a determined expression of  $N$ , the upper bound for the difference  $|f_N(x) - f(x)|$  is related to  $(\ln N)^\alpha / N^d$ . These results are very interesting, but they are obtained using a dataset defined on a lattice.

### 3. Results

#### 3.1. Radial Kernel Weighted Average

**Definition 1.** A parameterized radial function is a function  $\Phi : \mathbb{R}_+ \times \mathbb{R}_+ \rightarrow \mathbb{R}_+$  characterized by a parameter  $\omega$  that accomplishes the conditions:

$$\left. \begin{aligned} \forall \omega \neq 0 \quad \lim_{r \rightarrow \infty} \Phi(r, \omega) &= 0 \\ \forall r \neq 0 \quad \lim_{\omega \rightarrow 0} \Phi(r, \omega) &= 0 \end{aligned} \right\} \tag{28}$$

Following the definition, an example of parameterized radial function is the Gaussian:

$$\Phi(r, \omega) = e^{-\left(\frac{r}{\omega}\right)^2} \tag{29}$$

**Definition 2.** Given a function  $f(x)$  defined on a domain,  $\Omega$ , the  $\Phi$ -weighted average of  $f(x)$  in a point  $x_o$ ,  $c_*(x_o)$ , is a number implicitly defined as:

$$\int_{\Omega} [c_*(x_o) - f(x)] \Phi(\|x - x_o\|, \omega) dx = 0 \tag{30}$$

where  $\Phi(.,.)$  is a parameterized radial function.

**Definition 3.** The  $J$ -th moment of the radial function  $\Phi$  is defined as:

$$W^J(x_0) = \int_{\Omega} \|x - x_0\|^J \cdot \Phi(\|x - x_0\|, \omega) dx \tag{31}$$

Using this expression, Equation (30) can be written as:

$$c_*(x_0) \cdot W^0(x_0) = \int_{\Omega} f(x) \cdot \Phi(\|x - x_0\|, \omega) dx \tag{32}$$

**Definition 4.** Given a domain  $\Omega$ , the  $J$ -th core of width  $\chi$  (represented as  $K_{\Omega}^J(\omega, \chi)$ ), is the subdomain  $K_{\Omega}^J(\omega, \chi) \subset \Omega$  where  $\forall x_0 \in K_{\Omega}^J(\omega, \chi)$ , the function  $W^J(x_0)$  satisfies:

$$0 \leq W^J - W^J(x_0) \leq \chi \tag{33}$$

where  $W^J = \int_{\mathbb{R}^d} \|x\|^J \cdot \Phi(\|x\|, \omega) dx$ .

**Definition 5.** In the same conditions of Definition 4, the axial core of width  $\chi$  (represented as  $AK_{\Omega}(\omega, \chi)$ ), is the subdomain  $AK_{\Omega}(\omega, \chi) \subset \Omega$  where  $\forall x_0 \in AK_{\Omega}(\omega, \chi)$  and for every  $i = \{1, 2, \dots, d\}$ :

$$| \int_{\Omega} (x - x_0)^i \cdot \Phi(\|x - x_0\|, \omega) dx | \leq \chi \tag{34}$$

For practical effects, Expression (33) implies considering  $W^J(x)$  as the constant value  $W^J$  for points belonging to  $K_{\Omega}^J(\omega, \chi)$  within the precision given by  $\chi$ . Whereas, (34) is equivalent to using a zero value for the integral inside  $AK_{\Omega}(\omega, \chi)$  with the same consideration with respect to the precision, properties that will be used to obtain an upper bound for the error of the estimation calculated with the algorithm on points belonging to  $K_{\Omega}^0(\omega, \chi) \cap AK_{\Omega}(\omega, \chi)$ .

### 3.2. Interpolation over a Mesh of Finite Elements

Let the decomposition of the domain in  $N_e$  elements  $\Omega = \cup_{i=1}^{N_e} \Omega_i$  and  $Q$  nodes  $\{\zeta_r\}_{r=1}^Q$  with the associated shape functions  $\{N_r(x)\}_{r=1}^Q$ . Then, the following relations are true:

$$\left. \begin{aligned} \sum_{r=1}^Q N_r(x) &= 1 \\ N_r(\zeta_S) &= \delta_{rS} \end{aligned} \right\} \tag{35}$$

Given a point  $x_0 \in \Omega_l \subset \Omega$  with nodes  $\{\zeta_r\}_{r=1}^Q$  and a function  $f(x)$ , the interpolated value of the function in the point  $\hat{f}(x_0)$  is defined as:

$$\hat{f}(x_0) = \sum_{r=1}^Q N_r(x_0) \cdot f(\zeta_r) \tag{36}$$

**Definition 6.** Given a function  $f(x)$  defined on  $\Omega$ , and a point  $x_0 \in \Omega_l$  the regression estimation of  $f(x)$  on  $x_0$  associated to the radial kernel  $\Phi$  and the mesh  $(\{\zeta_r\}_{r=1}^Q, \{\Omega_i\}_{i=1}^{N_e})$  is defined as the interpolation of the weighted averages of  $f$  on the nodes of  $\Omega_l$ :

$$\hat{z}(x_0) = \sum_{r=1}^Q N_r(x_0) \cdot c_*(\zeta_r) \tag{37}$$

This expression is closely related with the approaches presented in (23)–(25) and (27).



**Definition 7.** Given a function  $f(x)$  defined on a domain  $\Omega$  and a random field  $e(x) \sim N(0, \sigma(x))$  defined on  $\Omega$ , an experimental realisation of  $f$  associated with a sample  $e^S(x)$  of  $e(x)$  is a function  $y^S : \Omega \rightarrow \mathbb{R}$ , given by:

$$y^S(x) = f(x) + e^S(x) \tag{38}$$

**Proposition 1.** The expected value (denoted as  $E[\dots]$ ) of the weighted average corresponding to the values  $y^S$  in any point  $x_o$  is:

$$E[c_*^S(x_o)] = c_*(x_o) \tag{39}$$

**Proof.** Following Expression (30), the regression of the experimental realisation on each node is:

$$\int_{\Omega} [c_*^S(x_o) - y^S(x)] \cdot \Phi(\|x - x_o\|, \omega) dx = 0 \tag{40}$$

Using (38), the last expression can be written as:

$$\int_{\Omega} [c_*^S(x_o) - f(x)] \cdot \Phi(\|x - x_o\|, \omega) dx = \int_{\Omega} e^S(x) \cdot \Phi(\|x - x_o\|, \omega) dx \tag{41}$$

Therefore, by (30) and using the definition (31) of  $W^0(x_o)$ :

$$(c_*^S(x_o) - c_*(x_o)) \cdot W^0(x_o) = \int_{\Omega} e^S(x) \cdot \Phi(\|x - x_o\|, \omega) dx \tag{42}$$

Under the conditions of Definition 7,  $E[e^S(x)] = 0$ , and the expected value of (42) is:

$$E[c_*^S(x_o) - c_*(x_o)] \cdot W^0(x_o) = 0 \tag{43}$$

Therefore,

$$E[c_*^S(x_o)] = c_*(x_o) \tag{44}$$

□

**Proposition 2.** Let  $e(x)$  be a distribution not correlated for different points of  $\Omega$ ,  $E[e^S(x) \cdot e^S(y)] = \sigma^2(x) \cdot \delta(x - y)$ . Then, the variance of the regression corresponding to the values  $y^S$  is given by:

$$Var(c_*^S(x_o) - c_*(x_o)) \cdot (W^0(x_o))^2 = \int_{\Omega} \sigma^2(x) \cdot \Phi^2(\|x - x_o\|, \omega) dx \tag{45}$$

**Proof.** Taking (42) to the square and calculating the expected values:

$$E[(c_*^S(x_o) - c_*(x_o))^2] \cdot (W^0(x_o))^2 = \int_{\Omega} \int_{\Omega} E[e^S(x) \cdot e^S(y)] \cdot \Phi(\|x - x_o\|, \omega) \cdot \Phi(\|y - x_o\|, \omega) dx dy = \int_{\Omega} \sigma^2(x) \cdot \Phi^2(\|x - x_o\|, \omega) dx \tag{46}$$

That is:

$$Var(c_*^S(x_o) - c_*(x_o)) \cdot (W^0(x_o))^2 = \int_{\Omega} \sigma^2(x) \cdot \Phi^2(\|x - x_o\|, \omega) dx \tag{47}$$

□

**Proposition 3.** (Upper bound of the error on inner points). Let  $f$  be a function  $f \in C^2(\Omega)$  defined on  $\Omega \subset \mathbb{R}^d$ ,  $x_0 \in K_\Omega^0(\omega, \chi) \cap AK_\Omega(\omega, \chi)$  and  $y^S$  a Gaussian experimental realisation of  $f$ . Then:

$$E[|\hat{z}^S(x_0) - f(x_0)|] \leq M \cdot d \cdot h^2 + \sqrt{\frac{2}{\pi}} \cdot \sigma \tag{48}$$

**Proof.** From Equation (37):

$$W^0(x_0) \cdot [\hat{z}^S(x_0) - f(x_0)] = W^0(x_0) \cdot \left[ \sum_{r=1}^Q N_r(x_0) \cdot c_r^S(\zeta_r) - f(x_0) \right] \tag{49}$$

If the element containing  $x_0$  accomplishes that  $\Omega_i \subset K_\Omega^0(\omega, \chi)$  up to precision  $\chi$ ,  $W^0(x) \approx W^0(\zeta_r) \approx W^0$ , and using (32) and (41):

$$\begin{aligned} W^0(x_0) \cdot [\hat{z}^S(x_0) - f(x_0)] &= W^0(\zeta_r) \cdot \left[ \sum_{r=1}^Q N_r(x_0) \cdot c_r^S(\zeta_r) - f(x_0) \right] = \\ &= \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega [y^S(x) - f(x_0)] \cdot \Phi(\|x - \zeta_r\|, \omega) dx = \sum_{r=1}^Q N_r(x_0) \cdot \\ &= \int_\Omega [f(x) - f(x_0)] \cdot \Phi(\|x - \zeta_r\|, \omega) dx + \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega e^S(x) \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{50}$$

Developing now  $f(x)$  around  $x = x_0$ , the last expression takes the form:

$$\begin{aligned} W^0(x_0) \cdot (\hat{z}^S(x_0) - f(x_0)) &= \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega (\nabla f(x_0) \cdot (x - x_0) + \frac{1}{2} \cdot D^\alpha f(p_r(x)) \cdot \\ &= (x - x_0)^\alpha) \cdot \Phi(\|x - \zeta_r\|, \omega) dx + \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega e^S(x) \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{51}$$

where  $\alpha$  is a two-dimensional multi-index. Writing  $x - x_0 = x - \zeta_r + \zeta_r - x_0$ , the first part of the integral takes the form:

$$\begin{aligned} \int_\Omega \nabla_i f(x_0) \cdot (x - x_0)^i \cdot \Phi(\|x - \zeta_r\|, \omega) dx &= \nabla_i f(x_0) \cdot \int_\Omega (x - \zeta_r)^i \cdot \\ &= \Phi(\|x - \zeta_r\|, \omega) dx + \nabla_i f(x_0) \cdot \int_\Omega (\zeta_r - x_0)^i \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{52}$$

If that  $\zeta_r \in AK_\Omega(\omega, \chi)$  the first integral is approximately zero, while for the second:

$$\begin{aligned} \sum_{r=1}^Q N_r(x_0) \cdot \nabla_i f(x_0) \cdot (\zeta_r - x_0)^i \cdot W^0(\zeta_r) &= W^0(x_0) \cdot \nabla_i f(x_0) \cdot \sum_{r=1}^Q N_r(x_0) \cdot \\ &= (\zeta_r - x_0)^i = 0 \end{aligned} \tag{53}$$

That is:

$$\begin{aligned} W^0(x_0) \cdot (\hat{z}^S(x_0) - f(x_0)) &= \sum_{r=1}^Q N_r(x_0) \cdot \frac{1}{2} \cdot \int_\Omega D^\alpha f(p_r(x)) \cdot (x - x_0)^\alpha \cdot \\ &= \Phi(\|x - \zeta_r\|, \omega) dx + \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega e^S(x) \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{54}$$

Taking the absolute value:

$$\begin{aligned} |W^0(x_0) \cdot (\hat{z}^S(x_0) - f(x_0))| &\leq \frac{1}{2} \cdot \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega \|D^\alpha f(p_r(x))\| \cdot \|x - x_0\|^\alpha \cdot \\ &= \Phi(\|x - \zeta_r\|, \omega) dt + \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega |e^S(x)| \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{55}$$

If  $M$  is an upper bound for the norm of the hessian matrix:

$$\begin{aligned} W^0 \cdot |\hat{z}^S(x_0) - f(x_0)| &\leq \frac{1}{2} \cdot \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega M \cdot \|x - x_0\|^2 \cdot \Phi(\|x - \\ &= \zeta_r\|, \omega) dt \leq W^0 \cdot \frac{1}{2} \cdot M \cdot d \cdot h^2 + \sum_{r=1}^Q N_r(x_0) \cdot \int_\Omega |e^S(x)| \cdot \Phi(\|x - \zeta_r\|, \omega) dx \end{aligned} \tag{56}$$

For the last integral, the expected value is:

$$E\left[\int_{\Omega} |e^s(x)| \cdot \Phi(\|x - \zeta_r\|, \omega) dx\right] = \int_{\Omega} \int_0^{\infty} |\epsilon| \cdot \pi(x, |\epsilon|) \cdot \Phi(\|x - \zeta_r\|, \omega) d\epsilon dx \tag{57}$$

where  $\pi(x, |\epsilon|)$  is the probability of an error  $\epsilon$  in the point given by  $x$ . In the case when this probability is independent of the point  $\pi(x, |\epsilon|) = \pi(|\epsilon|)$ :

$$E\left[\int_{\Omega} |e^s(x)| \cdot \Phi(\|x - \zeta_r\|, \omega) dx\right] = W^0(\zeta_r) \cdot \int_0^{\infty} |\epsilon| \cdot \pi(|\epsilon|) d\epsilon = W^0 \cdot \int_0^{\infty} |\epsilon| \cdot \pi(|\epsilon|) d\epsilon \tag{58}$$

In the case of a Gaussian distribution of  $\epsilon$ :

$$E\left[|\hat{z}^S(x_o) - f(x_o)|\right] \leq \frac{1}{2} \cdot M \cdot d \cdot h^2 + \sqrt{\frac{2}{\pi}} \cdot \sigma \tag{59}$$

□

### 3.3. Computational Algorithm

A natural selection for the parameter  $\omega$  of the radial function is the value corresponding to the element width  $h$ . In addition, in real cases, complete sets of values of  $y^S(x)$  are not available, so one must use just a subset of  $P$  points. Then, from this subsample, the integrals are calculated using finite sums, so:

$$\int_{\Omega} \Phi(\|x - x_o\|, \omega) dx \cong \frac{\mu(\Omega)}{P} \cdot \sum_{k=1}^P \Phi(\|x_k - x_o\|, h) \tag{60}$$

$$\int_{\Omega} y^s(x) \cdot \Phi(\|x - x_o\|, \omega) dx \cong \frac{\mu(\Omega)}{P} \cdot \sum_{k=1}^P y_k^s \cdot \Phi(\|x_k - x_o\|, h) \tag{61}$$

Now, the finite sample version of (40) is:

$$\sum_{k=1}^P [c_S(x_o) - y_k^S] \cdot \Phi(\|x_k - x_o\|, h) = 0 \tag{62}$$

Following Expression (62), the proposed algorithm can be condensed in the next schema (Algorithm 1):

---

**Algorithm 1.**

---

1. **Begin**
  2. **Step 1: for** i =1:P **do**
    3. Initialize the point Q\_1 with X[i].
    4. Initialize estimation[i] with 0.
    5. **for** k = 1:2<sup>dimension</sup> **do**
      6. Initialize the point Q\_node with Node[k].
      7. Initialize dist and nodeEstim with 0.
      6. **for** j = 1:P **do**
        7. Initialize the point Q\_2 with X[j].
        8. Increment dist with Radial\_Kernel(Q\_node, Q\_2)
        9. Increment nodeEstim with Z[j] \* Radial\_Kernel(Q\_node, Q\_2).
      10. **end for** j
    11. **end for** k
    12. Update the estimate node as nodeEstim = nodeEstim/dist.
    13. Increment estimation[i] with nodeEstim \* ShapeFunction(k, Q\_1).
  14. **end for** i
  15. **End.**
-

The calculation of radial kernels in lines 9 and 10 for a radial function as that in Equation (29), dimension  $d$  implies the sums of the  $d$ -coordinate differences  $r^i = x^i - x_k^i$ . The loop from 7 to 10 iterates over the number of points  $P$ , so given that these steps are repeated in the  $2^d$  nodes of the finite element for each objective points, the computational time of the algorithm is  $t_c \sim O(P^2 \cdot 2^d \cdot d)$ . The memory complexity corresponds to saving the experimental points, that is,  $m \sim O(P)$ . An analysis of the relation between execution times and number of points will be considered in the applications section.

The analysis developed above shows clearly the improvement of the current algorithm compared with those presented in [4,5] where the dominant computational orders were  $O((c + 1)^{3d})$  and  $O((c + 1)^d)$ . Remembering that the complexity is related with the precision of the estimation (without considering overfitting problems), the previous methodologies were highly limited in their application by the dimensionality of the problem. However, this limit disappears in this new approach, with the number of points being the main limiting factor.

Let us consider a normalised dataset. Then, the models are determined by the parameter  $h = \frac{1}{c}$  that represents the length of the edge of the hypercubes that form the discretization of the domain  $\Omega$ . Each value of  $c$  (complexity) determines a unique model, but following (37), when  $h \rightarrow 0$ , the regression estimation accomplishes that:

$$\hat{z}(x_o) \rightarrow c_*^s(x_o) \tag{63}$$

In that limit,  $\hat{z}(x_o)$  corresponds to a weighted average of the  $K$ -nearest points to  $x_o$ , with  $K$  depending on the parameter  $\omega$  in  $\Phi$ , which has been fixed as  $\omega = h$ .

Let us consider the case of an estimation on a point  $(x_0^1, x_0^2, \dots, x_0^d)$ , based on the model defined by a dataset  $\{(x_k^1, x_k^2, \dots, x_k^d; y_k)\}_{k=0}^P$  containing  $x_o$ . For commodity, also let the dataset be ordered with respect to the distance to  $x_o$ , with  $\Phi_i = \Phi(\|x_i - x_o\|, h)$  the parameterized radial function. The weighted average regression is then obtained as:

$$c_*^s(x_o) \cdot (1 + \sum_{k=1}^P \Phi_k) = y_0^S + \sum_{k=1}^P y_k^S \cdot \Phi_k \tag{64}$$

However, if the estimation is calculated without using the self  $x_o$ , the restricted weighted average regression  $c_{r*}^s$  takes the form:

$$c_{r*}^s(x_o) \cdot \sum_{k=1}^P \Phi_k = \sum_{k=1}^P y_k^S \cdot \Phi_k \tag{65}$$

Therefore, there exists a relationship between both expressions:

$$c_*^s(x_o) \cdot (1 + \sum_{k=1}^P \Phi_k) = y_0^S + c_{r*}^s(x_o) \cdot \sum_{k=1}^P \Phi_k \tag{66}$$

If the model dataset has a low density, the sum vanishes compared with the constant terms, obtaining:

$$c_*^s(x_o) \approx y_0^S \tag{67}$$

Joining (63) and (67), in the limit  $h \rightarrow 0$ , the result of the estimation tends to  $\hat{z}(x_o) \rightarrow y_0^S$ , so there will be a trend to the overfitting as the complexity grows in the full model.

The restricted model obtained from Equation (65) does not incorporate the information about the experimental value, if present, on the objective point, so the model is not using all the available information and it is obviously under fitted.

### 3.4. Applications

To test the algorithm behaviour, several problems have been selected from the UCI Machine Learning Datasets [69] to compare the results with those presented in other papers obtained using different techniques. UCI is a well-known repository and widely used data source for problems related to machine learning, regression, and time series prediction.

The fit of a model can be measured using different parameters, where  $y_i$  is the observed and  $\tilde{y}_i$  is the estimated value:

1.  $R^2$  coefficient: The strict use of  $R^2$  is limited to the linear regression. Out of this case, the value of  $R^2$  can vary in the interval  $]-\infty, 1]$ . However, the expression that defines  $R^2$  is related to the other family of quality indicators as Legates-McCabe, Nash and Sutcliffe, and Willmott:

$$R^2 = 1 - \frac{\sum_{i=1}^P (y_i - \tilde{y}_i)^2}{\sum_{i=1}^P (y_i - E[y])^2}$$

2. Mean squared error (MSE): This indicator is associated to the second moment of the error, incorporating information about its variance and bias. Its principal problem is the overweighting of outliers, so usually, other parameters as MAE are preferred:

$$MSE = \frac{1}{P} \cdot \sum_{i=1}^P (y_i - \tilde{y}_i)^2$$

3. Root mean squared error (RMSE): Squared root of MSE. Its main advantage, with respect to it, is that it is measured in the same units of the predicted variable. However, it suffers of the same problems with respect to the outlier influence:

$$RMSE = \sqrt{\frac{1}{P} \cdot \sum_{i=1}^P (y_i - \tilde{y}_i)^2}$$

4. Mean absolute error (MAE): The best characteristic of MAE compared to RMSE is the equal weight of the deviations on the error. The ratio RMSE/MAE serves as an indication of the outlier presence on the sample:

$$MAE = \frac{1}{P} \cdot \sum_{i=1}^P |y_i - \tilde{y}_i|$$

5. Mean absolute percentage error (MAPE): This parameter is frequently used by its simple interpretation as a relative error. However, it presents a bias that tends to select the model with smaller forecasts. One option could be the symmetric MAPE, but is not so widely used:

$$MAPE = \frac{100}{P} \cdot \sum_{i=1}^P \left| \frac{y_i - \tilde{y}_i}{y_i} \right|$$

6. Regression error characteristic (REC) curve: It is a curve that obtains plotting the error tolerance on the X-axis versus the percentage of points predicted within the tolerance on the Y-axis.

The proposed methodology is a deterministic algorithm and the obtained model (and in consequence the quality parameters) depends only on the available experimental data, so there is no difference in them between several runnings of the model.

### 3.4.1. Air foil Self Noise

The study of self-noise is an active topic in aerodynamics with a great number of research papers devoted to its prediction or improving the design of air foils and other structures. The phenomena related to this problem are complex involving vortex, turbulence, and flows in the linear-limit.

The NASA "Air foil Self Noise" dataset [70,71] was obtained from a series of aerodynamic and acoustic tests of two and three-dimensional air foil sections conducted in an anechoic wind tunnel.

The data set comprises different size NACA 0012 air foils (with sizes from 2.5 to 61 cm) at various wind tunnel speeds (up to Mach 0.21) and angles of attack (from 0 to 25.2°). The span of the air foil and the observer position were the same in all of the experiments. The measurements were made in a wind tunnel.

The data is composed of 1503 rows with five independent variables to determine the sound pressure level as indicated in Table 1.

**Table 1.** Input and output variables for the air foil self noise.

Input Variables	Output Variable
1. Frequency (Hz).	Scaled sound pressure level (dB)
2. Angle of attack (deg).	
3. Chord length (m).	
4. Free-stream velocity (m/s).	
5. Suction side displacement thickness (m).	

Table 2 shows the parameters of quality for the models obtained in previous research using this dataset. The techniques are linear regression (LR), decision tree (DT), support vector machine regression (SVMR), random forest regression (RF), AdaBoost regression (ABR), XGBoost regression (XGBR), several types of neural networks (NN), and Stochastic Gradient Tree Boosting (SGTB).

**Table 2.** Parameters of quality for models of air foil self noise dataset in previous papers.

Research	Model	R <sup>2</sup>	MSE	RMSE	MAE
[71]	LR	0.558	22.129	4.704	3.672
	DT	0.871	6.457	2.541	1.788
	SVMR	0.751	12.483	3.533	2.674
	RF	0.935	3.283	1.812	1.313
	ABR	0.725	13.796	3.714	3.078
	XGBR	0.946	2.707	1.645	1.074
	Baseline NN	0.676	16.236	4.029	3.078
	Deeper NN	0.802	9.916	3.149	2.474
[72]	RF	0.929			
	SGTB	0.969			
[73]	NN	0.952 (R <sup>2</sup> of linear regression predicted/real)			
[74]	NN	0.894			
[75]	NN	0.943/0.838		(several models)	

The first step is to study the behaviour of a cross validation using (80% of the data as train and 20% as test sets) and 100 iterations. The result for different values of the complexity is shown in Figure 2.

Apart from the expected result of a smaller error in the train subsets, an important behaviour can be observed in the MAE corresponding to the test subsamples with a stable value further than complexity 90 (with a mean MAE of 1.958). This value corresponds to a complexity close to 30 on the train subsamples.

The next step is the study of the models generated by the full dataset to test if this trend is verified. To do this, for each complexity two estimations have been done for the points of the dataset, corresponding to the full and the restricted estimations introduced in Equations (64) and (65). The results of MAE, MAPE, and R<sup>2</sup> coefficients are shown in Figure 3.

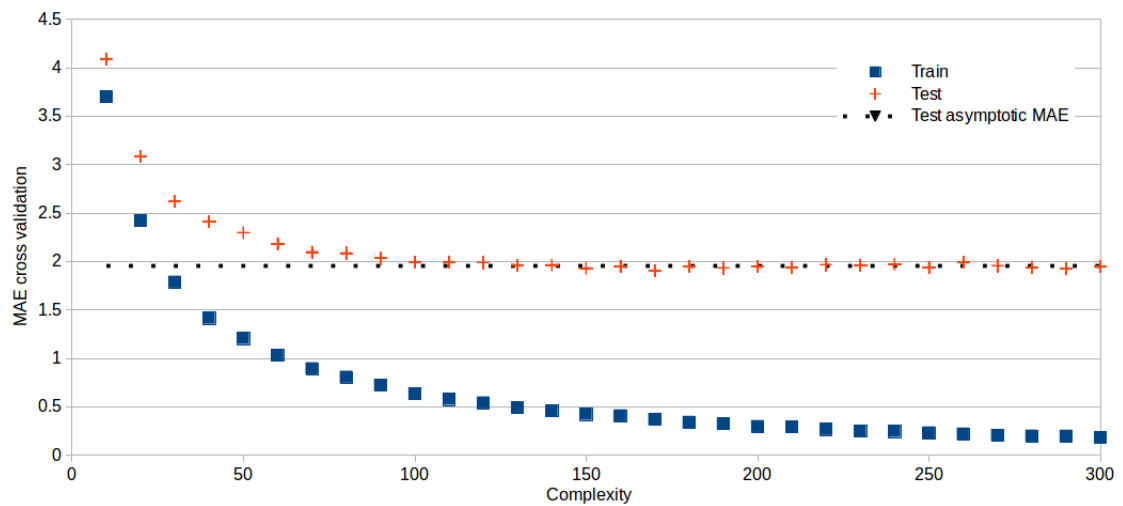


Figure 2. Mean absolute error (MAE) of train and test samples on cross validations with a train size of 80% for complexities between 10 and 300.

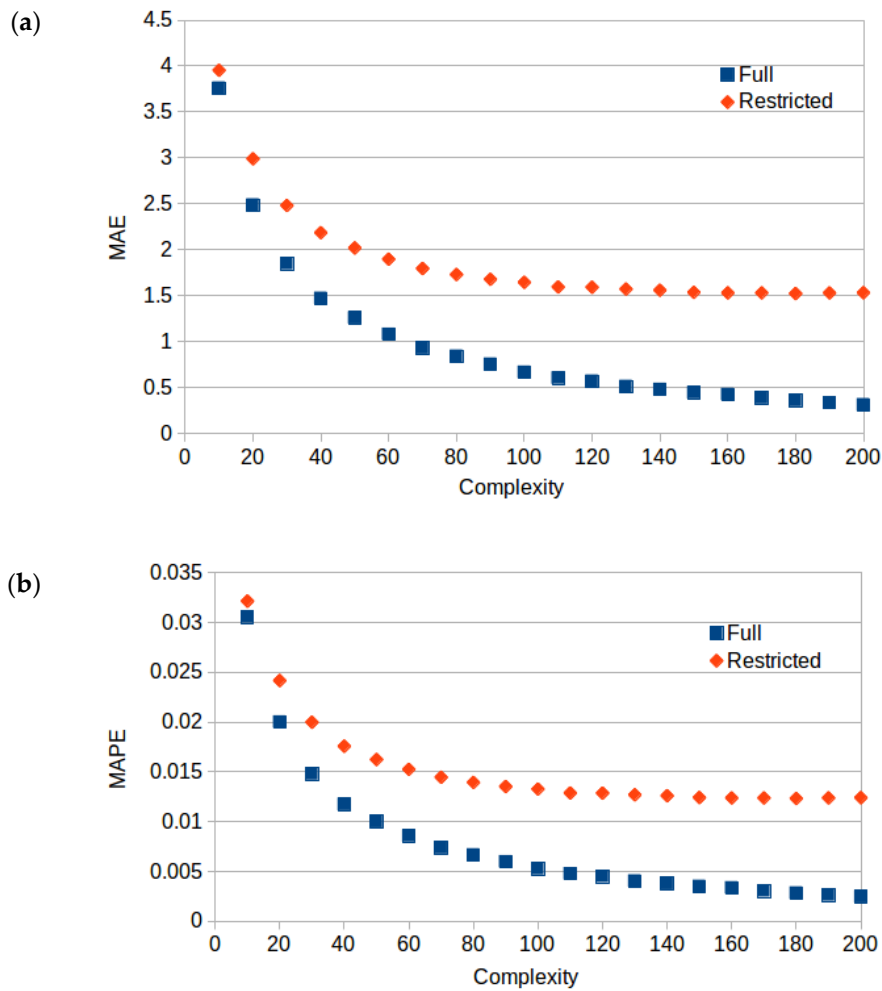
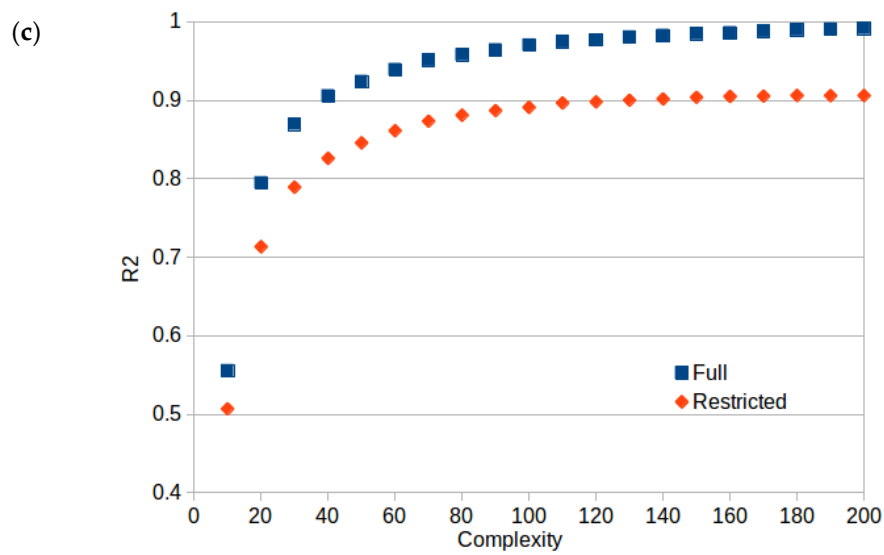


Figure 3. Cont.

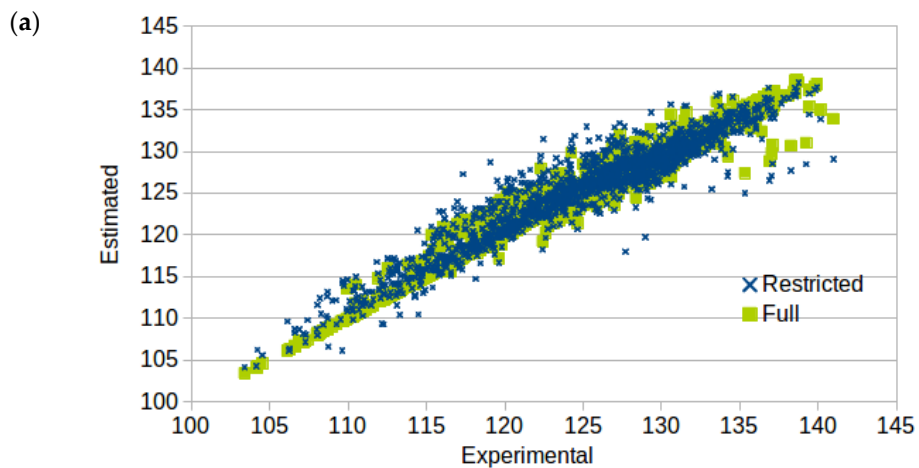


**Figure 3.** Quality parameters for the full and restricted models on the dataset, for different complexities from 10 to 200. (a) MAE. (b) Mean absolute percentage error (MAPE). (c)  $R^2$ .

The behaviour is consistent with the one observed in Figure 2, with a stable value in MAE around complexity 100 and MAPE from complexity 80. In addition, it is clear how the full model has diminishing values for MAE and MAPE coefficients when the complexity grows (making evident in that case the existence of overfitting because improvements in the full model are not parallel to improvements in the values of the restricted model).

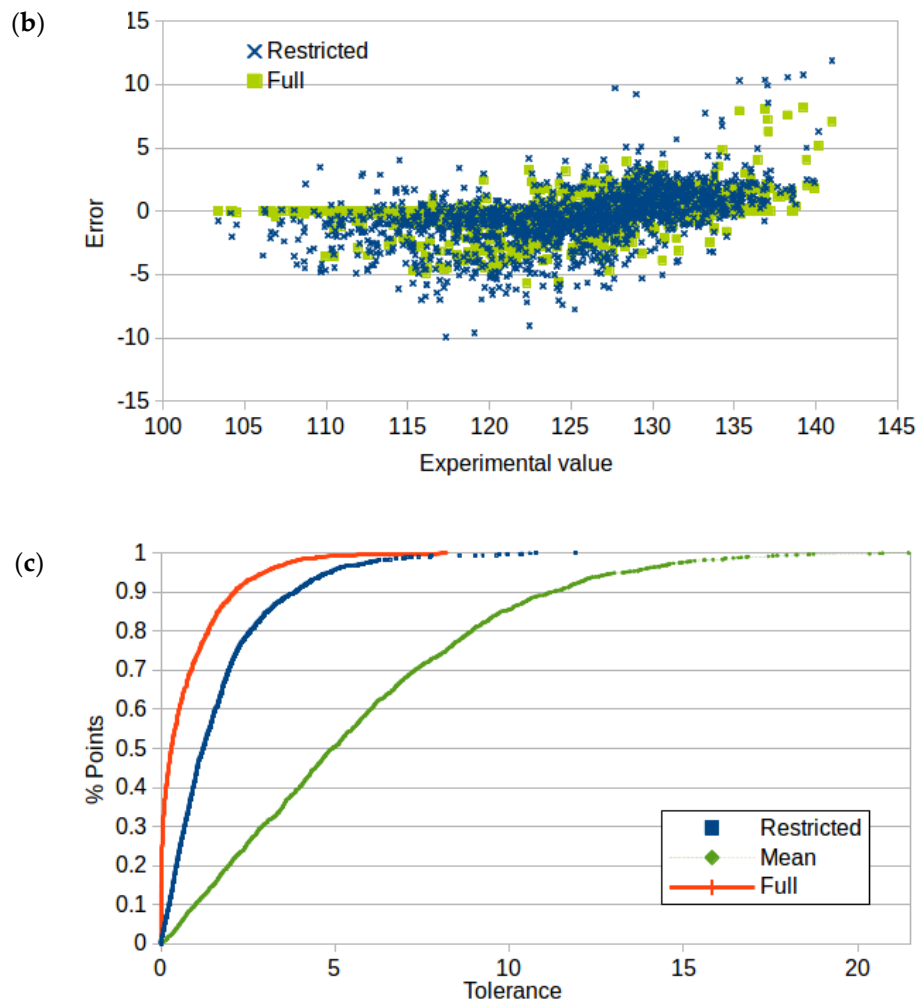
Let us then study the characteristics of the model with complexity 90.

Figure 4 shows the errors as pairs observed/estimated for the full and restricted models for complexity 90.



**Figure 4.** Cont.





**Figure 4.** Results for full (solid) and restricted models (x-marks) for complexity 90. (a) Estimated vs. experimental values. (b) Errors vs. experimental. (c) Regression error characteristic (REC) curves of full, restricted, and null (mean) models.

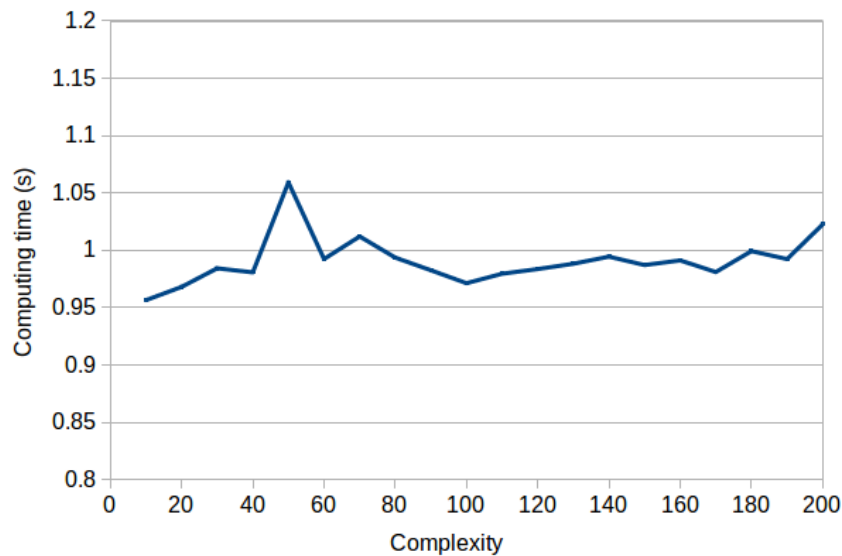
The parameters of quality for this model are shown in Table 3.

**Table 3.** Parameters of the model with complexity 90 for the air foil self noise dataset.

Parameter	Full Model	Restricted Model
R <sup>2</sup>	0.964	0.887
MAE	0.755	1.679
MAPE	0.60%	1.35%
MSE	1.711	5.387
RMSE	1.308	2.321

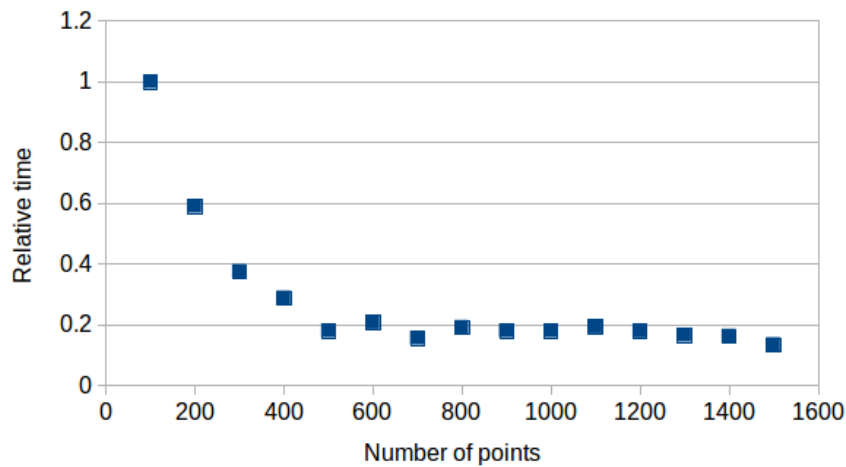
The results are comparable to those presented in Table 2 corresponding to previous papers.

The computation time over 10 iterations has been measured for different complexities, obtaining a result shown on Figure 5. Given the expression of the computational order  $O(P^2 \cdot 2^d \cdot d)$  no dependence on complexity is expected, a result that is ratified by the plot results.



**Figure 5.** Computing time (in seconds) as a function of the complexity for the air foil self noise problem.

To confirm the quadratic dependence on the number of points  $O(P^2 \cdot 2^d \cdot d)$ , a study of the computing time for different values of  $P$  in the dataset, for a high complexity (in this study complexity 100 has been selected) has been realised. The result for the value of the variable has been obtained, the computing time divided by the square of the number of points is shown in Figure 6.



**Figure 6.** Relative computing time over the squared number of points (time over  $100^2$  points as unit) as a function of the problem size for complexity 100.

The constant behaviour at a greater number of points of the last figure certifies the hypothesis introduced on the dependence  $t_c \sim P^2$ .

### 3.4.2. Combined Cycle Power

The problem of effective prediction of power output for electric power plants is of great importance in order to optimize the economic profit by the generated megawatt. For a gas turbine, the relationship between the power available under full load conditions and variables as temperature, pressure, relative humidity, and the exhaust vacuum measured from steam turbine, is complex enough to be studied using machine learning methods. Specifically, given the diversity in the values of the different periods (summer and winter) regression and prediction methods based on locality will have a better performance. This makes this dataset a good benchmark for the current methodology.

The dataset contains 9568 data points collected from a combined Cycle Power Plant over six years (2006–2011), when the power plant was set to work with a full load. The variables that correspond to the hourly average values are presented in Table 4.

**Table 4.** Input and output variables for the combined cycle power dataset.

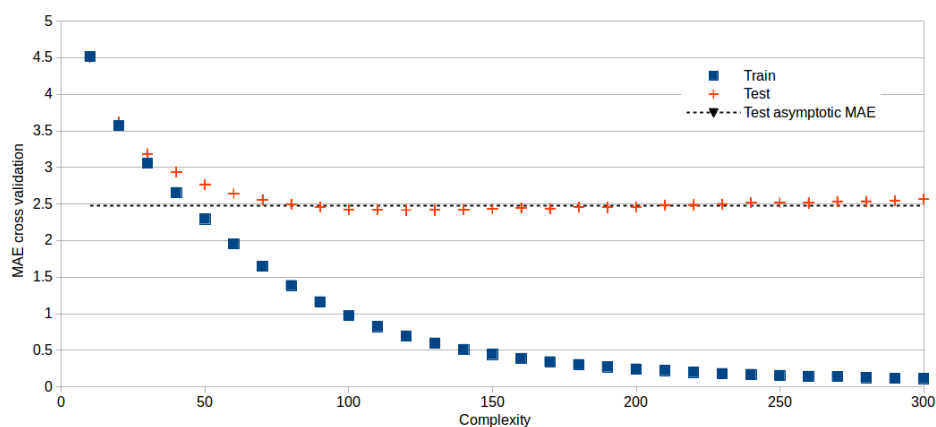
Input Variables	Output Variable
1. Temperature	Net hourly electrical energy output
2. Ambient pressure	
3. Relative Humidity	
4. Exhaust Vacuum	

For additional details, consider the original papers [76,77]. The models considered in [76] are: (i) Least median square (LMS); (ii) support vector poly kernel regression (SMOReg); (iii) KStar (K \*); (iv) bagging REP tree (BREP); (v) model trees rules (M5R); (vi) model trees regression (M5P) and; (vii) reduced error pruning (REP) trees. The results presented in Table 5 for [77] are: (i) k-nearest neighbour (k-NN) smoother + feedforward error backpropagating ANN and; (ii) the best of several combinations of K-means clustering + feedforward error backpropagating ANN.

**Table 5.** Parameters of quality for the models of combined cycle power dataset in previous research.

Research	Model	MSE	RMSE	MAE
[76]	LMS	20.903	4.572	3.621
	SMOReg	20.821	4.563	3.620
	K *	14.907	3.861	2.882
	BREP	14.341	3.787	2.818
	M5R	17.040	4.128	3.172
	M5P	16.704	4.087	3.140
	REP	17.733	4.211	3.133
[77]	k-NN + ANN	19.990		
	k-means + ANN	15.430		

Following the steps done in the first example, the cross validation results can be seen in Figure 7.



**Figure 7.** MAE of train and test samples on cross validations with a train size of 80% for complexities between 10 and 300.

Now, the value of the stabilised MAE for the test subsamples (calculated from complexities above 70) is 2.479, corresponding to a complexity of 50 in the train subsample. Running the full and restricted models for complexities between 10 and 200, it can be seen that the stable behaviour for MAE and

MAPE starts before that value, and can be fixed around complexity 70. Figure 8 shows the comparative of the quality parameters (MAE, MAPE, and  $R^2$ ) for the full and restricted models on the dataset, for different complexities from 10 to 200.

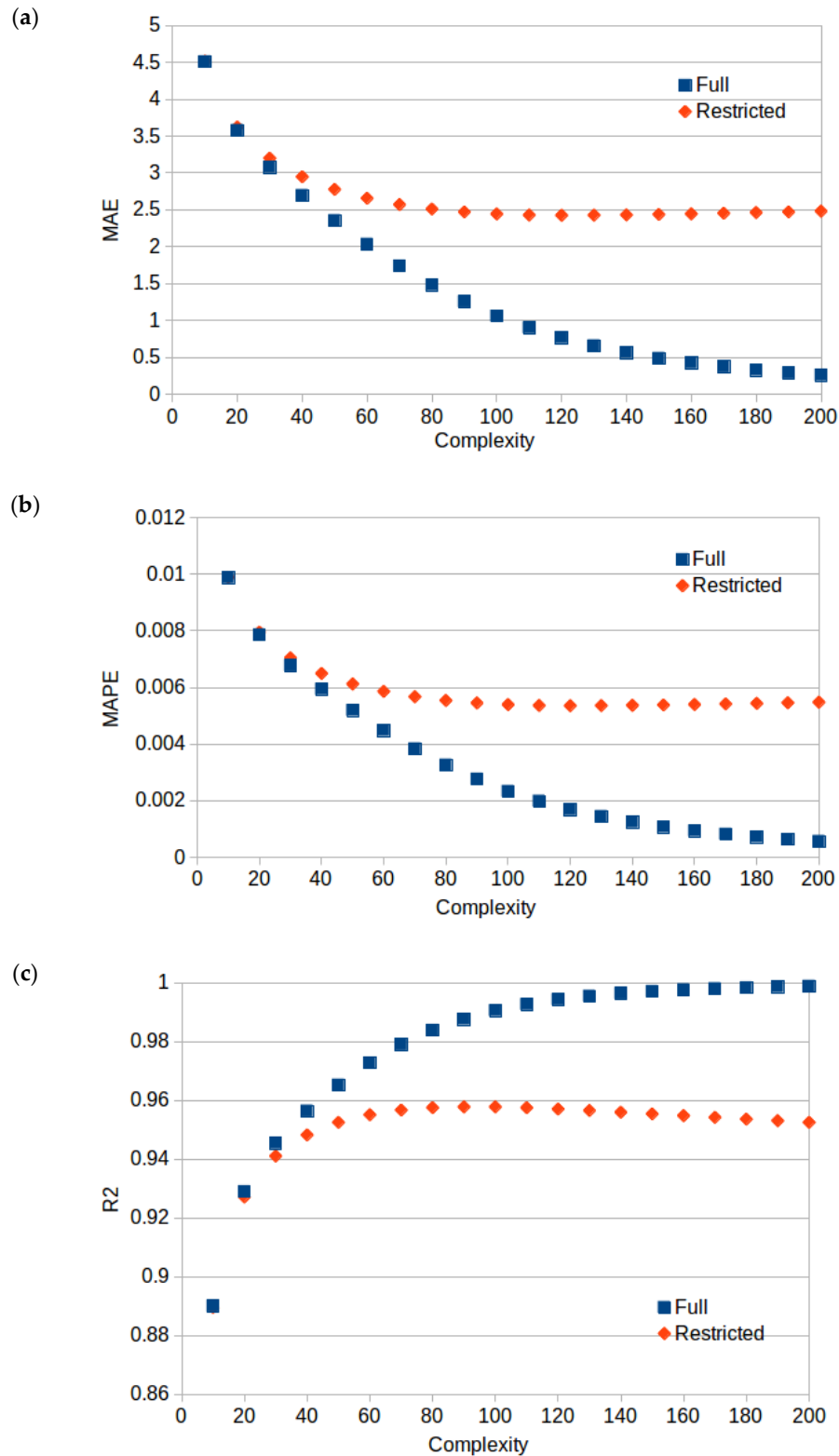
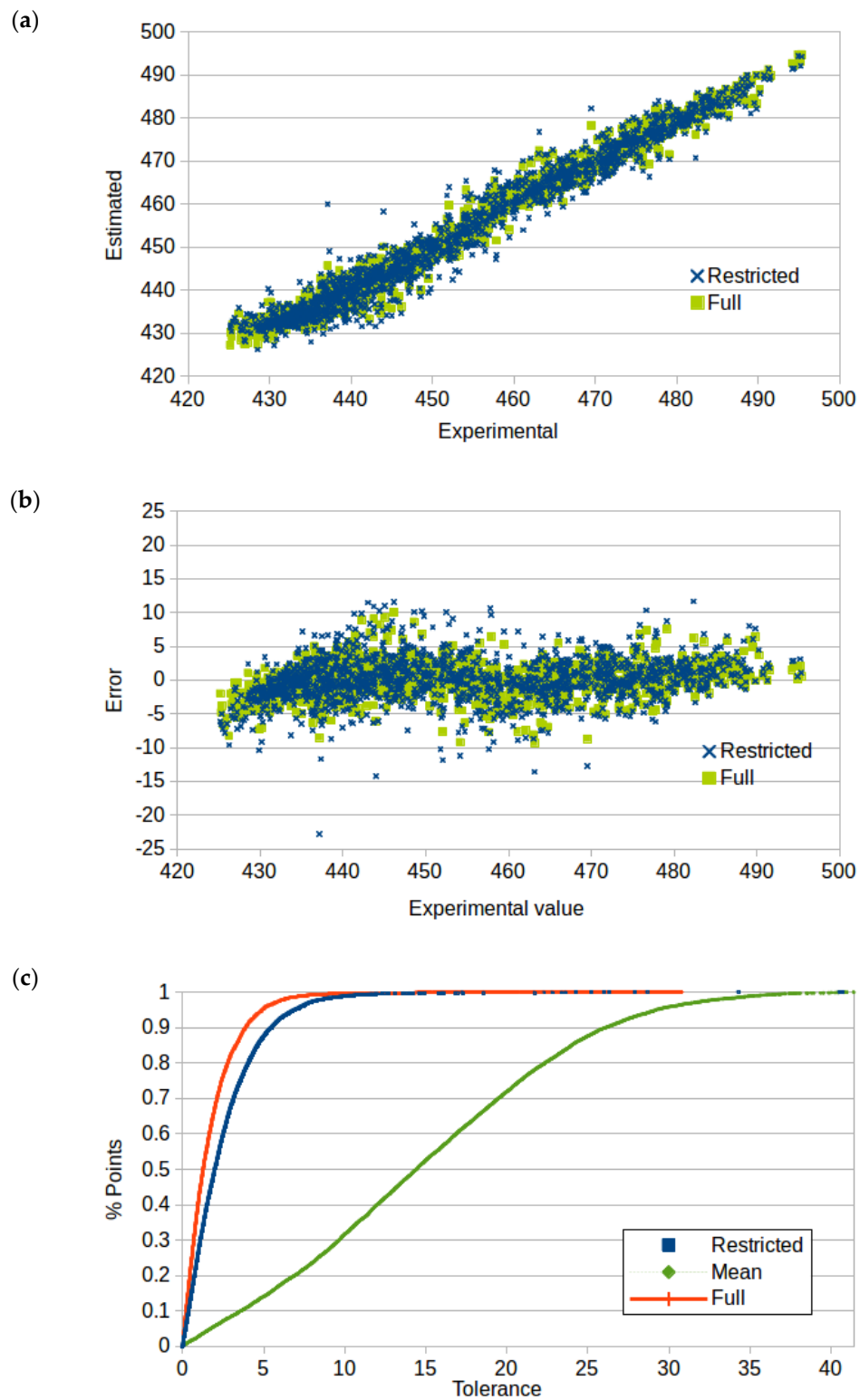


Figure 8. Quality parameters for the full and restricted models on the dataset, for different complexities from 10 to 200. (a) MAE. (b) MAPE. (c)  $R^2$ .

To be conservative with respect to the possibility of overfitting, complexity 60 is selected to run the final model. The corresponding plot of estimated versus experimental values can be seen in Figure 9.



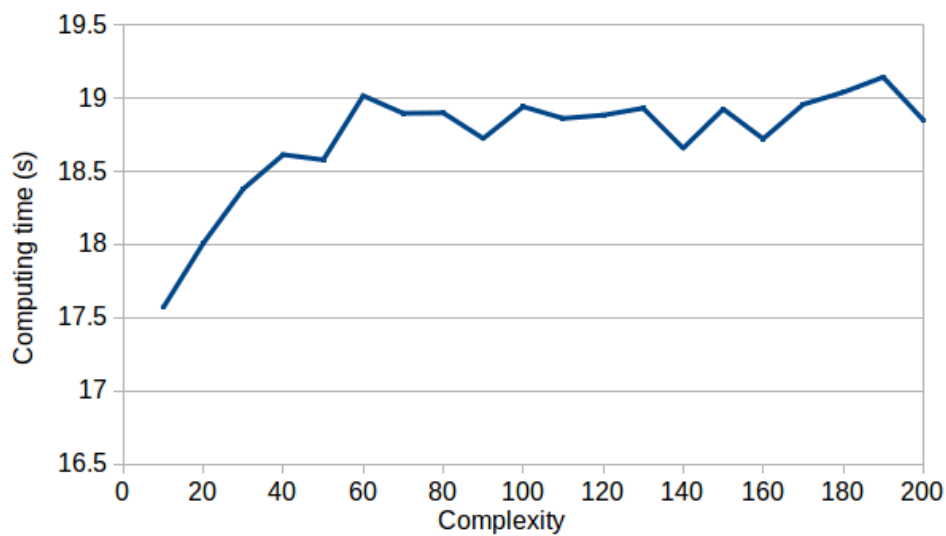
**Figure 9.** Results for full (solid) and restricted models (x-marks) for complexity 70. (a) Estimated vs. experimental values. (b) Errors vs. experimental. (c) REC curves of full, restricted, and null (mean) models.

The parameters of quality for this model are shown in Table 6.

**Table 6.** Parameters of the model with complexity 70 for the combined cycle power dataset.

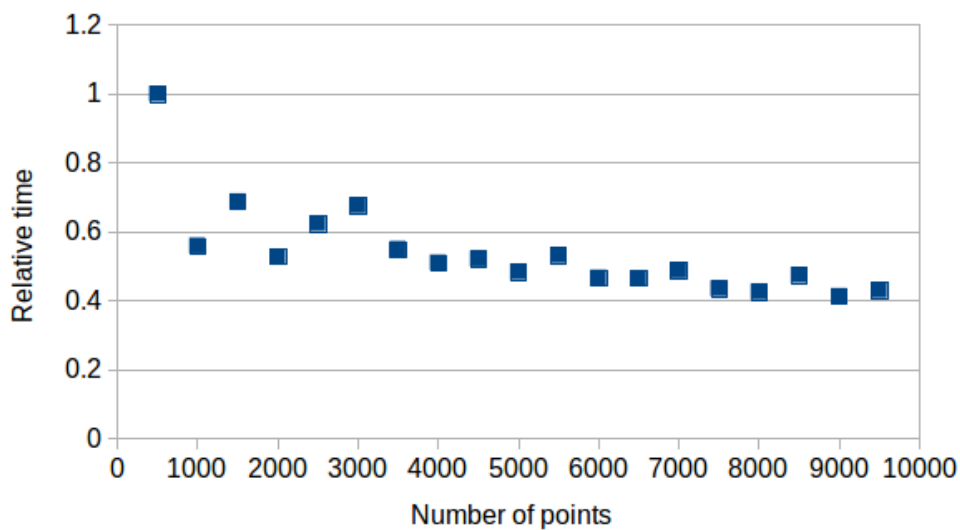
Parameter	Full Model	Restricted Model
R <sup>2</sup>	0.979	0.957
MAE	1.741	2.572
MAPE	0.39%	0.57%
MSE	6.102	12.592
RMSE	2.470	3.548

The computation time has been also obtained from 10 iterations for complexities between 10 and 200, showing its independence in Figure 10.



**Figure 10.** Computing time (in seconds) as a function of the complexity for complexities between 10 and 200.

Again, a representation of  $t_c/P^2$  for complexity 100 is shown in Figure 11.



**Figure 11.** Relative computing time over the squared number of points (time over 500<sup>2</sup> points as unit) as a function of the problem size for complexity 100.

#### 4. Discussion

Several questions must be considered in this step. Firstly, the presented algorithm continues the line of work developed by the authors in previous papers, see [3,4,78], solving the common bottleneck caused by the dependence of the time of computation on the complexity of the model at orders  $O((c+1)^{3d})$  and  $O((c+1)^d)$ . Now, the dependence on the dimension has an expression of  $d \cdot 2^d$ . This represents an important advance, allowing the treatment of a greater number of problems.

Secondly, given that for complexities less than 100, (for the most part of the studied problems this value is usually around 70–80), there exists a direct relationship between the complexity of the model and the precision of the estimation, the independence of the computing time of current algorithm from this parameter is an important result. However, the problem has been now displaced to the number of points, which contribution to the total computing time was hidden in the previous methodologies. Therefore, the two new challenges for future investigations will be the use of more efficient algorithms and data structures to improve the dependence of the number of points and the dimensionality of the problem that is present as  $d \cdot 2^d$ .

On the other side, the characteristics of the methodology allow a cheap parallel calculation of the estimated values for points belonging to the sample that determines the model in two different ways: The full and the restricted models. The study of their behaviour for different complexities allows the inclusion of a benchmark to capture the overfitting in the model. Overfitting of models is a very common problem that appears where the model tends to treat in-model and out-of-model points in a different way causing the estimations to be useless. Following in this direction, considering the development of algorithms for automatic detection of overfitting, and the selection of optimum complexity will be another pending task for future investigations. Considering that the process of selecting the most adequate complexity for the model is equivalent to optimising the error of a weighted combination of both models, its automation seems to be a feasible problem.

However, at the moment, obtaining an adequate combination of the full and restricted models introduced in 4.3 is a complex problem. This makes the comparisons with other available methodologies diffuse because the quality parameters of the optimal model would be an intermediate case between those corresponding to the full and restricted cases.

Another characteristic that could be considered as a distinctive aspect of the methodology is its ‘liquidity’, in the sense of the absence of a model as such, that is, the methodology does not generate any additional structure beyond the estimations and the optimum value for the  $h$  parameter. Thus, it can also be viewed as a positive aspect.

Finally, the proposed methodology has been used on well-known problems in order to compare the results with those obtained in other papers and methods.

For the Air Foil Self Noise dataset, the quality parameters of the restricted model, considered as the worst prediction are outperformed by decision trees, XGBoost regression, and some models of NN. However, the full model, considered as the most optimistic model (in occasions presenting overfitting) obtain results similar to the best available models.

In the case of Combined Cycle Power dataset, both models (full and restricted) obtain better quality parameters than the methodologies previously considered. This very interesting result should be studied more in depth to understand it, especially in the case of the restricted model.

In addition, a wider use of the regression method must be faced by increasing the number of studied datasets, and comparing the results with other available methods, especially those based on radial basis functions, to compare their performance.

**Author Contributions:** Conceptualization, F.J.N.-G. and Y.V.; methodology, Y.V. and F.J.N.-G.; software, F.J.N.-G. and M.C.-M.; validation, Y.V. and F.J.N.-G.; formal analysis, F.J.N.-G., Y.V., M.C.-M. and S.I.; investigation, F.J.N.-G., Y.V., M.C.-M. and S.I.; data curation, F.J.N.-G. and Y.V.; writing—original draft preparation, F.J.N.-G., Y.V., M.C.-M. and S.I.; writing—review and editing, F.J.N.-G., Y.V., M.C.-M. and S.I.; supervision, Y.V. and S.I.; project administration, Y.V. and S.I.; funding acquisition, S.I. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research has been partially funded by the Spanish Ministry of Science, Innovation and Universities, grant number RTI2018-101148-B-I00.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Gallagher, R.H. Finite element analysis: Fundamentals. *Int. J. Numer. Methods Eng.* **1975**, *9*, 732. [[CrossRef](#)]
- Brenner, S.; Scott, R. *The Mathematical Theory of Finite Element Methods*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
- Navarro-González, F.J.; Villacampa, Y. A new methodology for complex systems using n-dimensional finite elements. *Adv. Eng. Softw.* **2012**, *48*, 52–57. [[CrossRef](#)]
- Navarro-González, F.J.; Villacampa, Y. Generation of representation models for complex systems using Lagrangian functions. *Adv. Eng. Softw.* **2013**, *64*, 33–37. [[CrossRef](#)]
- Navarro-González, F.J.; Villacampa, Y. A finite element numerical algorithm for modelling and data fitting in complex systems. *Int. J. Comput. Methods Exp. Meas.* **2016**, *4*, 100–113. [[CrossRef](#)]
- Palazón, A.; López, I.; Aragonés, L.; Villacampa, Y.; Navarro-González, F.J. Modelling of Escherichia coli concentrations in bathing water at microtidal coasts. *Sci. Total Environ.* **2017**, *593–594*, 173–181. [[CrossRef](#)]
- Aragonés, L.; Pagán, J.I.; López, I.; Navarro-González, F.J. Galerkin's formulation of the finite elements method to obtain the depth of closure. *Sci. Total Environ.* **2019**, *660*, 1256–1263. [[CrossRef](#)]
- Migallón, V.; Navarro-González, F.; Penadés, J.; Villacampa, Y. Parallel approach of a Galerkin-based methodology for predicting the compressive strength of the lightweight aggregate concrete. *Constr. Build. Mater.* **2019**, *219*, 56–68. [[CrossRef](#)]
- Buhmann, M.D. Radial Basis Functions. *Acta Numer.* **2000**, *9*, 1–38. [[CrossRef](#)]
- Mai-Duy, N.; Tran-Cong, T. Approximation of function and its derivatives using radial basis function networks. *Appl. Math. Model.* **2003**, *27*, 197–220. [[CrossRef](#)]
- Kansa, E.J. Multiquadrics-A scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput. Math. Appl.* **1990**, *19*, 147–161. [[CrossRef](#)]
- Kansa, E.J.; Hon, Y.C. Circumventing the ill-conditioning problem with multiquadric radial basis functions: Applications to elliptic partial differential equations. *Comput. Math. Appl.* **2000**, *39*, 123–137. [[CrossRef](#)]
- Schaback, R.; Wendland, H. Adaptive greedy techniques for approximate solution of large RBF systems. *Numer. Algorithms* **1999**, *24*, 239–254. [[CrossRef](#)]
- Broomhead, D.S.; Lowe, D. Multivariable Functional Interpolation and Adaptive Networks. *Complex Syst.* **1988**, *2*, 321–355.
- Orr, M.J.L. *Introduction to Radial Basis Function Networks*; Center for Cognitive Science, Univ. of Edinburgh: Edinburgh, UK, 1996.
- Elanayar, S.; Shin, Y.C. Radial Basis Function Neural Network for Approximation and Estimation of Nonlinear Stochastic Dynamic Systems. *IEEE Trans. Neural Netw.* **1994**, *5*, 594–603. [[CrossRef](#)]
- Saha, A.; Keeler, J.D. Algorithms for better representation and faster learning in radial basis function networks. In *Advances in Neural Information Processing Systems*, 2nd ed.; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1990; pp. 482–489.
- Wettschereck, D.; Dietterich, T. Improving the performance of radial basis function networks by learning center locations. In *Advances in Neural Information Processing Systems*; Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA, 1992; pp. 1133–1140.
- Gomm, J.B.; Yu, D.L. Selecting radial basis function network centers with recursive orthogonal least squares training. *IEEE Trans. Neural Netw.* **2000**, *11*, 306–314. [[CrossRef](#)]
- Park, J.; Sandberg, I.W. Universal Approximation Using Radial-Basis-Function Networks. *Neural Comput.* **1991**, *3*, 246–257. [[CrossRef](#)]
- Poggio, T.; Girosi, F. Regularization algorithms for learning that are equivalent to multilayer networks. *Science* **1990**, *247*, 978–982. [[CrossRef](#)]
- Howell, A.J.; Buxton, H. Learning identity with radial basis function networks. *Neurocomputing* **1998**, *20*, 15–34. [[CrossRef](#)]



23. Scholkopf, B.; Sung Kah-Kay Burges, C.J.; Girosi, F.; Niyogi, P.; Poggio, T.; Vapnik, V. Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers. *IEEE Trans. Signal Process.* **1997**, *45*, 2758–2765. [[CrossRef](#)]
24. Chng, E.S.; Chen, S.; Mulgrew, B. Gradient radial basis function networks for nonlinear and nonstationary time series prediction. *IEEE Trans. Neural Netw.* **1996**, *7*, 190–194. [[CrossRef](#)]
25. Chen, S. Nonlinear time series modelling and prediction using Gaussian RBF networks with enhanced clustering and RLS learning. *Electron. Lett.* **1995**, *31*, 117–118. [[CrossRef](#)]
26. Li, M.M.; Verma, B. Nonlinear curve fitting to stopping power data using RBF neural networks. *Expert Syst. Appl.* **2016**, *45*, 161–171. [[CrossRef](#)]
27. Han, H.G.; Chen, Q.L.; Qiao, J.F. An efficient self-organizing RBF neural network for water quality prediction. *Neural Netw.* **2011**, *24*, 717–725. [[CrossRef](#)] [[PubMed](#)]
28. Yilmaz, I.; Kaynar, O. Multiple regression, ANN (RBF, MLP) and ANFIS models for prediction of swell potential of clayey soils. *Expert Syst. Appl.* **2011**, *38*, 5958–5966. [[CrossRef](#)]
29. Fabri, S.; Kadiramanathan, V. Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Trans. Neural Netw.* **1996**, *7*, 1151–1167. [[CrossRef](#)]
30. Chen, S.; Billings, S.A.; Grant, P.M. Recursive hybrid algorithm for non-linear system identification using radial basis function networks. *Int. J. Control.* **1992**, *55*, 1051–1070. [[CrossRef](#)]
31. Yu, D.L.; Gomm, J.B.; Williams, D. Sensor fault diagnosis in a chemical process via RBF neural networks. *Control Eng. Pract.* **1999**, *7*, 49–55. [[CrossRef](#)]
32. Yang, F.; Paindavoine, M. Implementation of an RBF Neural Network on Embedded Systems: Real-Time Face Tracking and Identity Verification. *IEEE Trans. Neural Netw.* **2003**, *14*, 1162–1175. [[CrossRef](#)]
33. Tsai, Y.-T.; Shih, Z.-C. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.* **2006**, *25*, 967–976. [[CrossRef](#)]
34. Cho, S.-Y.; Chow, T.W.S. Neural computation approach for developing a 3D shape reconstruction model. *IEEE Trans. Neural Netw.* **2001**, *12*, 1204–1214.
35. Oglesby, J.; Mason, J.S. Radial basis function networks for speaker recognition. In Proceedings of the 1991 International Conference on Acoustics, Speech, and Signal Processing, Toronto, ON, Canada, 14–17 April 1991; pp. 393–396.
36. Schilling, R.J.; Carroll, J.J.; Al-Ajlouni, A.F. Approximation of nonlinear systems with radial basis function neural networks. *IEEE Trans. Neural Netw.* **2001**, *12*, 1–15. [[CrossRef](#)] [[PubMed](#)]
37. Jang, J.-S.; Sun, C.-T. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. Neural Netw.* **1993**, *4*, 156–159. [[CrossRef](#)] [[PubMed](#)]
38. Billings, S.A.; Zheng, G.L. Radial basis function network configuration using genetic algorithms. *Neural Netw.* **1995**, *8*, 877–890. [[CrossRef](#)]
39. Chen, S.; Wu, Y.; Luk, B.L. Combined genetic algorithm optimization and regularized orthogonal least squares learning for radial basis function networks. *IEEE Trans. Neural Netw.* **1999**, *10*, 1239–1243. [[CrossRef](#)] [[PubMed](#)]
40. Kubat, M. Decision trees can initialize radial-basis function networks. *IEEE Trans. Neural Netw.* **1998**, *9*, 813–821. [[CrossRef](#)]
41. Benoudjit, N.; Archambeau, C.; Lendasse, A.; Lee, J.A.; Verleysen, M. Width optimization of the Gaussian kernels in Radial basis function networks. In Proceedings of the ESANN 2002, 10th Euroean Symposium on Artificial Neural Networks, Bruges, Belgium, 24–26 April 2002.
42. González, J.; Rojas, I.; Ortega, J.; Pomares, H.; Fernandez, F.J.; Diaz, A.F. Multiobjective evolutionary optimization of the size, shape, and position parameters of radial basis function networks for function approximation. *IEEE Trans. Neural Netw.* **2003**, *14*, 1478–1495. [[CrossRef](#)] [[PubMed](#)]
43. Er, M.J.; Wu, S.; Lu, J.; Toh, H.L. Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Netw.* **2002**, *13*, 697–710.
44. Bugmann, G. Normalized Gaussian radial basis function networks. *Neurocomputing* **1998**, *20*, 97–110. [[CrossRef](#)]
45. Hofmann, S.; Treichl, T.; Schroder, D. Identification and observation of mechatronic systems including multidimensional nonlinear dynamic functions. In Proceedings of the 7th International Workshop on Advanced Motion Control, Maribor, Slovenia, 3–5 July 2002; pp. 285–290.

46. Saha, P.; Tarafdar, D.; Pal, S.K.; Srivastava, A.K.; Das, K. Modelling of wire electro-discharge machining of TiC/Fe in situ metal matrix composite using normalized RBFN with enhanced k-means clustering technique. *Int. J. Adv. Manuf. Technol.* **2009**, *43*, 107–116. [[CrossRef](#)]
47. Mori, H.; Awata, A. Data mining of electricity price forecasting with regression tree and normalized radial basis function network. In Proceedings of the 2007 IEEE International Conference on Systems, Man and Cybernetics, Montreal, QC, Canada, 7–10 October 2007; pp. 3743–3748.
48. Deco, G.; Neuneier, R.; Schümann, B. Non-parametric data selection for neural learning in non-stationary time series. *Neural Netw.* **1997**, *10*, 401–407. [[CrossRef](#)]
49. Deng, H.; Li, H.-X.; Wu, Y.-H. Feedback-linearization-based neural adaptive control for unknown nonaffine nonlinear discrete-time systems. *IEEE Trans. Neural Netw.* **2008**, *19*, 1615–1625. [[CrossRef](#)] [[PubMed](#)]
50. Ranković, V.; Radulović, J. Prediction of magnetic field near power lines by normalized radial basis function network. *Adv. Eng. Softw.* **2011**, *42*, 934–938. [[CrossRef](#)]
51. Golbabai, A.; Seifollahi, S.; Javidi, M. Normalized RBF networks: Application to a system of integral equations. *Phys. Scr.* **2008**, *78*, 15008. [[CrossRef](#)]
52. Nelles, O. Axes-oblique partitioning strategies for local model networks. In Proceedings of the 2006 IEEE Conference on Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, 4–6 October 2006; pp. 2378–2383.
53. Hartmann, B.; Nelles, O. On the smoothness in local model networks. In Proceedings of the 2009 American Control Conference, St. Louis, MO, USA, 10–12 June 2009; pp. 3573–3578.
54. Brett, M.; Penny, W.D.; Kiebel, S. *Human Brain Function II, an Introduction to Random Field Theory*; Academic Press: London, UK, 2003.
55. Pieczynski, W.; Tebbache, A.-N. Pairwise Markov random fields and segmentation of textured images. *Mach. Graph. Vis.* **2000**, *9*, 705–718.
56. Pantazis, D.; Nichols, T.E.; Baillet, S.; Leahy, R.M. A comparison of random field theory and permutation methods for the statistical analysis of MEG data. *NeuroImage* **2005**, *25*, 383–394. [[CrossRef](#)]
57. Zhang, Y.; Brady, M.; Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. *IEEE Trans. Med. Imaging* **2001**, *20*, 45–57. [[CrossRef](#)]
58. De Oliveira, V. Bayesian prediction of clipped Gaussian random fields. *Comput. Stat. Data Anal.* **2000**, *34*, 299–314. [[CrossRef](#)]
59. Adler, R.J.; Taylor, J.E. *Random Fields and Geometry*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
60. Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Vineet, V.; Su, Z.; Du, D.; Huang, C.; Torr, P.H. Conditional random fields as recurrent neural networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1529–1537.
61. Abrahamsen, P. A Review of Gaussian Random Fields and Correlation Functions. 1997. Available online: [https://www.researchgate.net/profile/Petter\\_Abrahamsen/publication/335490333\\_A\\_Review\\_of\\_Gaussian\\_Random\\_Fields\\_and\\_Correlation\\_Functions/links/5d68cc0692851c154cc5b9bd/A-Review-of-Gaussian-Random-Fields-and-Correlation-Functions.pdf](https://www.researchgate.net/profile/Petter_Abrahamsen/publication/335490333_A_Review_of_Gaussian_Random_Fields_and_Correlation_Functions/links/5d68cc0692851c154cc5b9bd/A-Review-of-Gaussian-Random-Fields-and-Correlation-Functions.pdf) (accessed on 17 September 2020). [[CrossRef](#)]
62. Radosavljevic, V.; Vucetic, S.; Obradovic, Z. Continuous Conditional Random Fields for Regression in Remote Sensing. In Proceedings of the ECAI 2010-19th European Conference on Artificial Intelligence, Lisbon, Portugal, 16–20 August 2010; pp. 809–814.
63. Zhang, J. The mean field theory in EM procedures for Markov random fields. *IEEE Trans. Signal Process.* **1992**, *40*, 2570–2583. [[CrossRef](#)]
64. Zhu, X.; Ghahramani, Z.; Lafferty, J.D. Semi-supervised learning using gaussian fields and harmonic functions. In Proceedings of the Twentieth International Conference Machine Learning (ICML 2003), Washington, DC, USA, 21–24 August 2003; pp. 912–919.
65. Cohen, A.; Jones, R.H. Regression on a random field. *J. Am. Stat. Assoc.* **1969**, *64*, 1172–1182. [[CrossRef](#)]
66. Hallin, M.; Lu, Z.; Tran, L.T. Local linear spatial regression. *Ann. Stat.* **2004**, *32*, 2469–2500. [[CrossRef](#)]
67. Carbon, M.; Francq, C.; Tran, L.T. Kernel regression estimation for random fields. *J. Stat. Plan. Inference* **2007**, *137*, 778–798. [[CrossRef](#)]
68. El Machkouri, M. Nonparametric regression estimation for random fields in a fixed-design. *Stat. Inference Stoch. Process.* **2007**, *10*, 29–47. [[CrossRef](#)]

69. Dua, D.; Graff, C. UCI Machine Learning Repository. 2019. Available online: [https://archive.ics.uci.edu/ml/citation\\_policy.html](https://archive.ics.uci.edu/ml/citation_policy.html) (accessed on 19 December 2019).
70. Brooks, T.F.; Pope, D.S.; Marcolini, M.A. *Airfoil Self-Noise and Prediction*; NASA: Washington, DC, USA, 1989.
71. NASA-Airfoil Noise, Revisiting Machine Learning Datasets. 2018. Available online: <https://www.simonwenkel.com/2018/11/06/revisiting-ml-NASA-airfoil-noise.html> (accessed on 20 September 2006).
72. Patri, A.; Patnaik, Y. Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise. *Procedia Comput. Sci.* **2015**, *46*, 109–121. [[CrossRef](#)]
73. The NASA Data Set, Machine Learning Examples. Predict the Noise Generated by Airfoil Blades. 2018. Available online: <https://www.neuraldesigner.com/learning/examples/airfoil-self-noise-prediction> (accessed on 19 December 2019).
74. Gonzalez, R.L. *Neural Networks for Variational Problems in Engineering*. Ph.D. Dissertation, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 2009.
75. Errasquin, L. *Airfoil Self-Noise Prediction Using Neural Networks for Wind Turbines*. Ph.D. Dissertation, Virginia Tech, Blacksburg, VA, USA, 2009.
76. Tüfekci, P. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *Int. J. Electr. Power Energy Syst.* **2014**, *60*, 126–140. [[CrossRef](#)]
77. Kaya, H.; Tüfekci, P.; Gürgen, S.F. Local and Global Learning Methods for Predicting Power of a Combined Gas & Steam Turbine. In *Proceedings of the International Conference on Emerging Trends in Computer and Electronics Engineering (ICETCEE 2012)*, Dubai, UAE, 24–25 March 2012.
78. Navarro-González, F.J.; Compañ, P.; Satorre, R.; Villacampa, Y. Numerical determination for solving the symmetric eigenvector problem using genetic algorithm. *Appl. Math. Model.* **2016**, *40*, 4935–4947. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).