A Nonlinear Black-Box Modelling Methodology for Neural Networks

Guy-Michel Cloarec, John Ringwood

A Nonlinear Black-Box Modelling Methodology for Neural Networks

Guy-Michel Cloarec, John Ringwood *Senior IEEE Member*

***Abstract*—The aim of this paper is to present a nonlinear black-box modelling methodology that, in a general context, introduces the use of various statistical techniques to improve the modelling performance. The improvements are obtained by addressing the main problem that any black-box technique is confronted with: the optimal choice of a model topology and the related parameters with respect to the complexity of the problem. The main idea is to efficiently combine statistical resampling and analysis with preprocessing, optimal topology determination, supervised parameter optimization and ensemble techniques.**

Nonlinear black-box modelling, statistical theory, preprocessing, curse of dimensionality, parameter optimization, ensemble techniques

I. Introduction

THIS paper will present a nonlinear black-box modelling methodology that aims to efficiently combine methods from different fields to improve the overall performance both in terms of time and accuracy of the model. The paper is organized as follows: In a Section II, we will introduce the main problems of nonlinear black-box modelling in a general context [1]. In Section III we will see how that black-box modelling is dependent on empirical data sets as its unique source of information, requiring an optimal analysis and use. Sections IV and V will address the most problematic issues in the modelling, i.e. the complexity issue and the model performance evaluation problems. Some preprocessing methods such as orthogonalization and clustering will be introduced and justified in Section VI. Section VII will examine the issued involved in the parameter optimization while Section VIII will present the benefits of ensemble methods. Finally, the modelling methodology will be put under test in Section IX with an application example.

II. Nonlinear Black-Box Models

The field of system identification can be classified into three main approaches : white-box, grey-box and black-box modelling. The *colour* refers to the growing uncertainty, either in term of physical insights or parameter accuracy, in the prior knowledge of the system. In practice, when the physical insights are not sufficient to define the model structure and identify the parameters from measured data, we have to resort to black-box modelling where no prior knowledge concerning either the structure or the causal inputs is available.

Control Systems Group, School of Electronic Engineering, Dublin City University Dublin, Ireland. E-mail cloarecg, ringwood@eeng.dcu.ie.

In such an approach, the model is obtained from the relationship between the past observation and the future states of the system by a functional operation called *mapping*. In that framework, a set of past observations, the regressors, are selected and transformed to provide the input space while the set of outputs is defined as the output space.

We will not detail the issues of the selection of the regressors as this topic has benefited from numerous contributions [2], [3]. However, in accordance with the statistical approach we have chosen, the selection of the optimal regressor is performed using statistical tests on the performance of linear models. This process can be implemented iteratively in the modelling process.

Both NARX and NOE regressor forms will be considered but the modelling methods described are in no way dependent on the regressor structure. For NARX models, we define the regression vector $\varphi(t)$ to be:

$$\varphi(t) = [u(t-1), \ldots, u(t-p), y(t-1), \ldots, y(t-q)] \quad (1)$$

where $u \in R^n$ and $y \in R^m$ are the system input and output respectively, while for NOE models, the regression vector $\hat{\varphi}(t)$ is:

$$\hat{\varphi}(t) = [u(t-1), \ldots, u(t-p), \hat{y}(t-1/\theta), \ldots, \hat{y}(t-q/\theta)] \quad (2)$$

where $\hat{y}(t-k/\theta)$ is the predicted output given the identified parameter vector θ.

The nonlinear model is then expressed by the equation:

$$y(t) = g_0(\varphi(t)) + v(t) \quad (3)$$

where $v(t)$ accounts for the fact that the next output will not be an exact function of the past and will be expected to be Gaussian noise with variance λ. g_0 is the hypothetical *true* function. In practice, g_0 is approximated by g, a nonlinear function defined by its finite-dimension parameter vector θ.

$$\hat{y}_g(t) = g(\varphi(t), \theta)) \quad (4)$$

g is searched within a family of functions characterized by θ. Since in black-box modelling no analytical methods to calculate θ are available, we will always deal with parameter estimates $\hat{\theta}$. As we will see later, that estimate will be conditioned to the data sets and the optimization method used.

Therefore, the main issues in nonlinear black-box modelling are to determine the optimal regressor vector φ , nonlinear mapping g and optimal parameter vector θ.

The nonlinear mapping function we will examine here are function expansions defined by:

$$g(\varphi) = \sum \alpha_k g_k(\varphi) \quad (5)$$

where g_k is the basis function. In most methods, the basis function is parameterized by a *motherbasis function* $k(x)$:

$$g_k(\varphi) = k(\varphi, \beta_k, \gamma_k) \tag{6}$$

Where β_k is the scale and γ_k the position of the basis function. The arbitrary choice of the motherbasis function leads to a taxonomy of methods. For example, trigonometric functions will lead to the Fourier expansion, Gaussian functions to *Radial Basis Networks*, functions obtained from a recursive equation to the *Wavelet Expansion* and the sigmoid functions to *Neural Network Systems*. It is also possible to extend that taxonomy to *Fuzzy Systems* although they are not usually perceived as functional expansions.

The functional expansion can be cascaded to obtain *connectionist models*. In such models, the expansion obtained from a *first layer system* is introduced to a second layer. Let $\varphi_k^{(2)}(t) = g_k(\varphi(t), \beta_k, \gamma_k)$ be the output of the first layer, where

$$\varphi^{(2)} = [\varphi_1^{(2)}(t), \ldots, \varphi_n^{(2)}(t)] \tag{7}$$

Instead of taking the linear combination of $\varphi^{(2)}$, we insert it into a second layer of basis function forming a second expansion. The second nonlinear expansion is defined by:

$$g(\varphi) = \sum_l \alpha_l g_l(\sum_k \alpha_{k,l} \varphi_k^{(2)}, \beta_l, \gamma_l) \tag{8}$$

In neural network terminology, such a model is a two-hidden layer neural network. The same process can be reiterated to obtain a multi-layered neural network. The parameter vector θ is then defined by :

$$\theta = \{\alpha_{k,l}, \beta_k, \gamma_k, \alpha_l, \beta_l, \gamma_l\} \tag{9}$$

The availability of various basis function leads to a great flexibility or capacity to approximate nonlinear mappings. On the other hand, that degree of freedom means that topological decisions must be made. The determination of the topology is related to the complexity of the problem as we will see in Section IV. Before considering that issue, we first have to examine the problems related to the use of empirical data.

III. Data Sets as Sources of Information

A. Definitions

As we mentioned in Section II, in black-box modelling, the only information available is within the data set of measured variables. That *raw* information must be processed to extract the behaviour of the system. That source of information can be corrupted by noise, may be non-causal or not correlated to causal input and/or may not be representative of the system behavior. Moreover, since that data will be the used throughout the modelling process, it is important to carefully analyze the data set.

Let D be the hypothetical *complete* data set of the mapping problem:

$$D = \{(\varphi(t), y(t)), t \in R\} \tag{10}$$

In practice, we deal with limited data sets, restrictions of D, defined by the distribution $\mathcal{F}$, which is usually unknown. The empirical data set available for the modelling is defined by:

$$D_{\mathcal{F}} = \{(\varphi(k), y(k)), k \in [1, M] \backslash (\varphi(k), y(k)) \hookrightarrow \mathcal{F}\} \tag{11}$$

where M is the number of samples in $D_{\mathcal{F}}$. Since M is finite, the *real* distribution of $D_{\mathcal{F}}$ is $\hat{\mathcal{F}}$. Knowledge to how much $D_{\mathcal{F}}$ is representative of D or how M relates to the complexity of the problem are generally unknown. Moreover, since the data set is the only source of information, the representativeness of $D_{\mathcal{F}}$ will condition each step of the modelling process, i.e. the model performance estimates, the topological determination and the parameter optimization.

The conditioning induced by the limited data sets is usually addressed by splitting the data set into three *independent* sets, the training, test and validation set respectively. The training set is used to performed the parameter optimization while the test set is used to evaluate the performance of the model. Finally, the validation set is used to evaluate the overall model. The main problem with that is because the complexity of the problem is unknown, we cannot define how independent and representative these sets are. Moreover, such restrictions reduce the information available for training which can be a real problem when the available information is scarce.

An important approach is issues from the statistical theory field and falls under the heading of *resampling techniques*. By resampling, we mean obtaining new data sets by drawing samples from the original data set. Let Q define the number of subsets drawn from $D_{\mathcal{F}}$. The resampled sets can be defined by:

$$D_{\hat{\mathcal{F}},q} = \{(\varphi(k), y(k)), k \in [1, N] \backslash (\varphi(k), y(k)) \hookrightarrow \hat{\mathcal{F}}\} \tag{12}$$

where

- q is the set index ($q \in [1, Q]$).
- N is the number of samples in the set ($N \leq M$).

Note that $\hat{\mathcal{F}}^*$ will be used to denote the real distribution of $D_{\hat{\mathcal{F}},q}$.

B. Cross-Validation

Cross-validation is a method using ideas from the statistical theory field to obtain better estimates of the model performance which are used in the topological and parameter optimization processes.

In Q-folded cross-validation, Q subsets containing $M-1$ samples each are created. Q estimates are then computed and the cross-validation estimates is obtained by averaging the subsets estimates. Let J be an estimate of the statistic t on the distribution $\mathcal{F}$. J is defined by:

$$J(\mathcal{F}) = \frac{1}{Q} \sum_{q=1}^{Q} t(D_{\hat{\mathcal{F}},q}) \tag{13}$$

The distribution of the $t(D_{\hat{\mathcal{F}},q})$ estimates can provide valuable information such as confidence limits on the estimates themselves. The main sources of error come from the fact that $D_{\mathcal{F}}$ may not be drawn uniformly from $\mathcal{F}$ (not representative) and the fact that the convergence of the estimates is obtained only for $Q \to \infty$.

It is important to notice that when the statistic t is quadratic, the distribution $\hat{\mathcal{F}}^*$ should be as Gaussian as possible.

C. Statistical Bootstrap

The averaging of statistics computed on subsets tends to unbias these estimates and make a better use of the data set. However, cross-validation requires subsets to contain $M-1$ samples which can generate heavy computation as the number of subsets Q increases (to ensure better confidence limits). Moreover, such resampling is not optimal in term of redundancy in the data set. That problem is also referred as the *Split Ratio* ($\gamma = \frac{N}{M}$) problem [8].

Statistical bootstrap is another resampling technique that uses uncorrelated subsets of size N. The subsets are obtained by drawing with replacement samples at random. Provided that Q and N are adequately chosen, the statistical bootstrap estimates will provide better estimates than cross-validation for a much smaller computation cost.

Throughout our method, we will use statistical bootstraps to estimate the model performance, i.e. the generalization, but also to create a small *uncorrelated* input subspaces on which parameter optimization is performed. However, as mentioned, the main problem with that method is the determination of the optimal Q and N. This issue is directly related to the complexity issue as we will see now and a solution to address the problem will be proposed in Section VI-C.

IV. Curse of Dimensionality

After the data sets are available for modelling, the nonlinear model g must be chosen. In connectionist models, this means determining the motherbasis functions and the dimension of the model. By dimension, we mean the dimension of the parameter space. The dimension of the model is related to its complexity, which, in the classification field, can be defined as the number of distinguishable patterns in the mapping.

For binary problems, the issue is adequately expressed by the VC-dimension [9]:

Theorem 1 (VC-dimension): Suppose $\mathcal{A}$ is a family of measurable functions mapping X into $\{0,1\}$. A set $S = \{x_1, \cdots, x_n\}$ is said to be shattered by $\mathcal{A}$ if each of the 2^n functions mapping S into $\{0,1\}$ is the restriction of S of some function in $\mathcal{A}$. Equivalently, S is shattered by $\mathcal{A}$ if for every subset $A \subseteq S$, there exists a corresponding function $a_A(.) \in \mathcal{S}$ such that :

$$\begin{aligned} a_A(x_i) &= 1 \quad if\ x_i \in \mathcal{A} \\ a_A(x_i) &= 0 \quad if\ x_i \notin \mathcal{A} \end{aligned} \tag{14}$$

The Vapnik-Chervonenkis dimension of $\mathcal{A}$, denoted VC-dim($\mathcal{A}$) is the largest integer n such that there exists a set of cardinality n that is shattered by $\mathcal{A}$.

The VC-dimension is defined for families of binary valued functions. The corresponding notion families of $[0,1]$-valued function is referred as the Pollard-dimension.

Theorem 2 (Pollard Dimension): Suppose $\mathcal{A}$ is a family of measurable function mapping X into $[0,1]$. A set $S = \{x_1, \cdots, x_n\}$ is said to be P-shattered by $\mathcal{A}$ if there exists a vector $C \in [0,1]^n$ such that, for every binary vector $e \in [0,1]^n$, there exists a corresponding function $a_e \in \mathcal{A}$ such that:

$$\begin{aligned} a_e(x_i) &\geq c_i \quad if\ e_i = 1 \\ a_e(x_i) &< c_i \quad if\ e_i = 0 \end{aligned} \tag{15}$$

The P-dimension, or *pseudo-dimension* of $\mathcal{A}$ denoted by P-dim($\mathcal{A}$) is the largest integer n such that there exists a set of cardinality n that is P-shattered by $\mathcal{A}$.

The P-dimension corresponds to the complexity of the problem and should also correspond to the complexity of the model used. It can be shown that the performance of the model is directly related to the match between the complexity of the output space and the complexity of the model, given an input data set:

$$\varepsilon_g(d) = \left(\frac{c}{d}\right)^{\alpha} + \varepsilon_{min} \tag{16}$$

where

- c is a measure of the target complexity,
- d is a measure of the complexity of the model structure,
- $\varepsilon_g(d)$ is the best generalization error achievable in model class $\mathcal{H}_d$,
- ε_{min} is the degree of unrealizability of the target with respect to the input and model structure, and
- α is a real number between 1 and 2.

It is possible to approximate the complexity of the target c by the first absolute moment of the Fourier magnitude distribution of the target function [12].

In equation (16), the performance of the model is quantified by a generalization error. In practice, neither c or d are available so that the model performance must be defined using numerical estimates. Under the assumption of stationarity, those estimates will quantify the performance of the model on data sets not used for the parameter optimization process. In Section V-B, we will examine generalization estimates based on statistical bootstraps.

It is important to match c and d to avoid the main problem of nonlinear black-box modelling: *the curse of dimensionality*. When the model offers too much degrees of freedom (too much parameters), idiosyncrasies in the training patterns are learned which downgrades the model generalization performance. When iterative parameter optimization methods are used, these problems are referred as *overtraining*. In the opposite case, the model does not have enough capacity to express the behaviour of the system.

We will see in Section VI and VII that preprocessing and statistical analysis techniques will be used to determine the optimal topology and avoid overtraining.

V. Model Performance Evaluation

A. Objective Function

Since the black-box modelling is based on the transformation of empirical data, the performance of the model must be evaluated at each step. The sources of performance downgrading can be identified as:

- Uncertainty due to the limited and eventually not representative data sets.
- Systematic error due to model defect.
- Model selection uncertainty.
- Parameter determination errors.

All these sources contribute to the observed prediction error. However, since modelling is an iterative process, the effects of some of these sources can be isolated and addressed

independently. The main techniques are based on the analysis of the statistical characteristics of the prediction error (analysis of residual, while others are based on the objective functions, such as MSE, evaluated on the prediction error. As the first approach has benefited from extensive study [2], we will examine in more detail the latter.

The model performance is quantified by a criterion $J(g)$:

$$J(g) = \int\int l(\varphi, y, g)p(\varphi, y)d\varphi dy \tag{17}$$

where

- $l(\varphi, y, g)$ is the risk that measures *how close* the model is to the data.
- $p(\varphi, y)$ is an unknown but fixed (stationary system) joint probability.

B. Generalization

In a probabilistic approach [1], [5], [11], the estimates are conditioned to the data set and equation (17) can be expressed by:

$$J(g, D_{\mathcal{F}}) = E_{\mathcal{F}}[l(\varphi, y, g)] \tag{18}$$

Where $E_{\mathcal{F}}$ is the expectation on the distribution $\mathcal{F}$.

The generalization criterion can be defined as the performance of g independently of $D_{\mathcal{F}}$. In our framework, it will be defined as the expectation of the prediction risk over all possible data sets D drawn from $\mathcal{F}$ of length M.

In the widely used case of quadratic objective functions, equation (18) is expressed by:

$$J(g, D_{\mathcal{F}}) = \lambda + E\|g_0(\varphi) - g_{D_{\mathcal{F}}}(\varphi, \theta)\|^2 \tag{19}$$

C. Bias-Variance Tradeoff

If we define the generalization estimate to be $J_*(g) = E(J(g, D_{\mathcal{F}}))$. We define two parameter vectors to be respectively the ideal parameter vector on D and on $D_{\mathcal{F}_*}$:

$$\begin{array}{rcl} \theta_\star & = & arg(minJ_*(g)) \\ \theta_{D_{\mathcal{F}}} & = & arg(minJ(g, D_{\mathcal{F}})) \end{array} \tag{20}$$

Then, according to [1], [5], the generalization estimate can be approximated by :

$$J_\star(g) \simeq \lambda + E\|g_0(\varphi) - g(\varphi, \theta_\star)\|^2 + E\|g(\varphi, \theta_D) - g(\varphi, \theta_{D_{\mathcal{F}}})\|^2 \tag{21}$$

Hence, equation (21) representing the MSE criterion can be decomposed into three parts : the noise, the *bias error* and the *variance error* respectively. The main reason for performing that decomposition is that each component is characteristic of a different source of uncertainty [5].

The bias error is related to the default of complexity in g and the limited number of samples. As $N \to \infty$, $\theta_* \to \theta$ and the criterion express the difference between the *true* model g_0 and the best possible approximation g given the structure and the topology.

The variance error is the result of the uncertainty on the vector parameter due to the limited number of samples and the optimization method. It is shown in [1] that the increase in the number of parameters induces an increase in the uncertainty on $\theta_{D_{\mathcal{F}}}$, which subsequently increases the variance error.

It appears then that the two components of the criterion have opposite behaviours with respect to the complexity of the model. As a consequence, a tradeoff will have to be found so that the final model will have an optimal topology and an optimal parameter vector, as we will see in Section VII. However, we can already say that this tradeoff will be performed during the parameter optimization using generalization estimates, as shown in Figure 1.

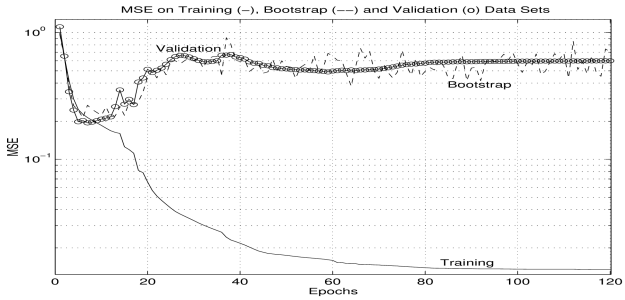

Fig. 1. Evaluation of the MSE during Training on Training, Bootstrap and Validation Data Sets.

D. Statistical Bootstrap Estimates of Generalization

In Section V-C, we had generalization estimates that used the hypothetical data set D. In practice, we deal with restricted and eventually non-uniformly distributed samples so that the predictions obtained are biased by what is defined in Section VII as the *conditioning of the estimates*.

The solution proposed to reduce the counter-effect of that conditioning will be the use of statistical bootstrap estimates [13]. The basic idea is to obtain estimates obtained on statistical bootstraps for different topology and optimization parameters. In most cases, the distribution of these estimates will be approximated by a Gaussian distribution, the center of which will give a better estimate (the *bootstrap estimate*).

Before giving an expression of the generalization estimate, we have to define the notation. Let $\hat{\theta}_{D_{\hat{\mathcal{F}},q}}$ be the parameter vector estimated from the optimization on the bootstrap $D_{\hat{\mathcal{F}},q}$. To express the properties of the samples, we use the indicator $\delta_{k,q}$:

$$\delta_{k,q} = \begin{cases} 1 & \text{if } k \notin D_{\hat{\mathcal{F}},q} \\ 0 & \text{if } k \in D_{\hat{\mathcal{F}},q} \end{cases} \tag{22}$$

We define now $J_{D_{\hat{\mathcal{F}},q}}(\hat{\theta}_{D_{\hat{\mathcal{F}},q\prime}})$ to be the performance of the model determined on bootstrap $q\prime$ and evaluated on bootstrap q.

$$J_{D_{\hat{\mathcal{F}},q}}(\hat{\theta}_{D_{\hat{\mathcal{F}},q\prime}}) = \frac{1}{N}\sum_{k=1}^{N}(1-\delta_{k,q})(y(k) - g_{D_{\hat{\mathcal{F}},q\prime}}(\varphi(k), \hat{\theta}_{D_{\hat{\mathcal{F}},q\prime}}))^2 \tag{23}$$

If we define Γ, the generalization estimates to be the average of $J_{D_{\hat{\mathcal{F}},q}}(\hat{\theta}_{D_{\hat{\mathcal{F}},q\prime}})$ on all possible bootstraps of size N, according to [13],it can be approximated by $\hat{\Gamma}$ given by the following equations.

We define β to be the average number of examples in the test set, i.e. the probability that a specific input-output pattern is not in a resample of size N.

$$\beta = (1 - \frac{1}{N})^N \tag{24}$$

It is important to note that $\beta \to e^{-1}$ when $N \to \infty$. To simplify the notation, we define $\hat{\gamma}(J)$ to be a *local* estimate of the criterion in the bootstrap world.

$$\hat{\gamma}(J) = \frac{J_{D_{\mathcal{F}}}(\hat{\theta}_q) - (1-\beta) J_{D_{\tilde{\mathcal{F}},q}}(\hat{\theta}_q)}{\beta} \tag{25}$$

The generalization statistical bootstrap estimate $\hat{\Gamma}(J)$ and its variance $S^2_{\hat{\Gamma}(J)}$ are defined as:

$$\begin{array}{rcl} \hat{\Gamma}(J) & \simeq & E\hat{\gamma}_q(J) \\ S^2_{\hat{\Gamma}(J)} & \simeq & E\|\hat{\Gamma}(J) - \hat{\gamma}_q(J)\|^2 \end{array} \tag{26}$$

The generalization statistical bootstrap estimate are used for different purposes at different stages of modelling. First, a bootstrap generalization estimate of linear models will be used for the regressor vector selection. It will be used to supervise an *early stop training* method (Section VII) and to provide training subsets for the network committee (Section VIII). Bootstraps estimates will be used in the evaluation of the model parameter distribution using an *evolutionary* optimization technique (Section VII-C).

The main drawback of the method is the choice of Q and N and is related to the complexity issue. Several statistical techniques such as the analysis of the statistics of performance criterion on linear models or complexity evaluation such as Fourier analysis can be used to give broad estimates.

In Section VI-C, we will see that Q and N can be reduced by supervising the draw of the bootstraps. Such reduction will decrease the computational load and will enable the use of more intensive optimization techniques.

VI. Clustering and other Preprocessing

A. Preprocessing Techniques

The term preprocessing covers a large set of techniques aiming at reducing the uncertainties in the input data set. In the black-box modelling approach, it involves determining the optimal inputs given the available information, i.e. the selection of the inputs, the regressors and eventually the transformations to operate on the regressor space. These transformations can be filtering (subsampling), oversampling or for multiscale problems, Wavelet based transformations. Of course, the preprocessing is very much dependent on the nature of the problem. The lack of quantitative criteria upon which the selection of the preprocessing can be based will involve *trial and errors* approaches as well as experience or basic understanding of the process.

In this paper, we will address two techniques which are supposed to address several problems encountered in black-box modelling but also general enough to be used in a variety of problems. The first technique called *Principal Component Analysis* is a method based on orthogonalization and reduction of the data space. The second set of techniques are defined as *clustering techniques* where clustering of the output space is used to infer partitions of the input space with a causal correlation. That clustering will be used as a basis for local modelling but also for the construction of more efficient statistical bootstraps.

B. Principal Component Analysis

1) Introduction: Principal Component Analysis (PCA) [7] is a statistical technique used to perform the analysis or the preprocessing of the input space. It is particularly useful for correlated input spaces obtained with regression models. The basic idea is to perform orthogonalization of the input space, where the axes of the rotation defines the *principal directions* on which the variance is maximal. For that reason, the rotated samples are called *principal components*. The analyzis of their characteristics can be used to perform subspace reduction, i.e. eliminating dimensions related to small variance or non-causal sources (such as noise or spurious regressors).

In the modelling process, the PCA method will be used in two different ways. The first one concerns itself with transformation of the input space. By extracting linear features and producing uncorrelated inputs, parameter optimization will be facilitated. Moreover, since we deal with a NARX structure, the inputs will be correlated giving PCA the potential to greatly reduce the dimension of the input space. This will reduce the complexity of the input space, will require less complex models, decrease the computation load and facilitate input space distribution analysis (see Section VI-C).

PCA will also be used in the regularization process where the analysis of the spaces defined by the output of the layers will be used to determine the optimal number of elements, a topological characteristic of the model. This will be examined in Section VII.

2) Definitions: Practically, the method consists of calculating the eigenvalues of the covariance matrix of the input space φ defined in equation (4) (the number of regressors is p). Let S be the covariance matrix of the input space defined by φ, also called the data matrix. To ensure that the transformation is isotropic, the input space is first normalized. If the input space is formed by p regressors, the orthogonalization can be described as follow:

$$S = E(\varphi\varphi^T) \tag{27}$$

We define $\tilde{S}$ to be the diagonal representation of S in the orthonormal base defined by the base vector $A = (a_1, \ldots, a_p)$ so that we have :

$$\tilde{S} = ASA^T \tag{28}$$

$$\tilde{S} = \begin{pmatrix} \tilde{s}_1^2 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{s}_p^2 \end{pmatrix} \tag{29}$$

From the decomposition, two results will be used : the orthogonalized vectors and the corresponding eigenvalues. We

define $\varphi^* = (\varphi^*(1), \ldots, \varphi^*(p))$ to be the orthogonalized input space, the matrix of the principal components.

$$\varphi^* = A\varphi \tag{30}$$

The components being mutually orthogonal, the corresponding signals are uncorrelated. The eigenvalues of S are the variance of the principal components, which is proportional to the *information* contained within the component. Because of uncertainties in modelling, it is likely that redundant or spurious information was introduced in the input space. As it is uncorrelated with the causal information, it can be isolated and extracted by eliminating the corresponding components. This is referred as the subspace reduction.

The main problem is then to find a criterion to distinguish the spurious components from the causal ones. One method used consists of evaluating the generalization performance of a linear model using exhaustive combination of component in the input space. Another, more often used, consists assumes that the dimensionality reduction prevails on the causality determination and performs the selection on the basis of a statistical criterion called *proportion of variance explained*, P_v.

$$\forall k \in [1, p],\ P_v(k) = \frac{\sum_{i=1}^{k} \tilde{s}_i^2}{\sum_{i=1}^{p} \tilde{s}_i^2} \tag{31}$$

A variance explained threshold, P_{v_0}, is arbitrarily defined (for example 95%) as the component selection criterion. Let l be the number of retained components:

$$l = \max_k (P_v(k) \leq P_{v_0}) \tag{32}$$

The resulting input space will be $\varphi^*_{red} = (\varphi^*(1), \ldots, \varphi^*(l))$. The transformed space will have a higher sample density and a smaller dimensionality. Such improvement will have effects on the clustering analysis expressed in Section VI-C and in turn on the topology and parameter determination, as we will see in Section VII.

C. Causal Information and Clustering Techniques

1) Introduction: In Section IV, the complexity of the problem, when expressed by the Pollard dimension, lead to the interpretation that the mapping problem lends itself to the concept of partition of the input and output spaces. The problem can be subdivided in two different ways.

The first one consists of reducing the complexity of the problem by using local models. A black-box model is then obtained by the combination of the local models. Local models are approximations of the system behaviour in certain *regions* of the output space. In our technique, these regions are expressed by partitions of the output space that we will define as *clusters*. As we will see in Section VI-C3, the parameter optimization of these local models requires that the sample space used is *related* to that partition. The partitions will be assumed to correspond to causality properties, a bijection in the mapping.

Another approach is to consider that the function approximation problem is related to a classification problem. The complexity is then only a matter of relating *input patterns* to *decision patterns*. As we deal with high dimensional continuous spaces, the input and decision patterns are clusters that must be identified. In Sections VI-C3 and VI-C4, we will see how clustering can be used to improve the performance of our modelling techniques.

2) Clustering Techniques: In the field of system identification, clustering techniques are mainly used by fuzzy modelling methods [17]. The clusters that partitions the input and output spaces are defined as fuzzy sets and the correlation between input and output clusters are used to construct the fuzzy rules.

Clustering techniques can be divided into two main groups, depending on the type of selection of the cluster. In the first group of techniques, clustering is performed by imposing the number of clusters. The describing parameters of which are tuned to minimize the criterion J_m. Let X be the variable associated with a set of clusters (classes).

$$\min_{U,V} \{J_m(U, V, X) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{i,k})^m \|x_k - V_i\|_A^2\} \tag{33}$$

where

- c is the number of clusters,
- $V = (V_1, \ldots, V_c)$ is the vector of clusters centers,
- $U = (u_{i,k})$ is the matrix of degrees of membership of the sample X_k with respect to cluster C_i,
- $m \in [1, \infty]$ is a weighting exponent, and
- $\| \cdot \|_A$ is any inner product norm,

Given n and m, a numerical method is used to obtain the *optimal* (U^*, V^*) that locally minimizes J_m. The matrix U is constrained to comply with a probabilistic interpretation by:

$$\forall k,\ \sum_{i=1}^{c} u_{i,k} = 1 \tag{34}$$

Equation (34) relates the fact that the $u_{i,k}$ can be interpreted as the probability of sample X_k belonging to class C_i.

The *Fuzzy Clustering Method* (FCM) [14] is a widely used method to find the local minimum using an iterative method. In order for the clustering to be optimal, the parameters c and m should be varied and the process performed a few times in order to reduce the dependencies, i.e. the non-deterministic nature of the parameter determination leading to various solution for different initial conditions, or the constrains on the number and the shape of the clusters.

The FCM can be related to other techniques such as the Gustafson-Kessel or Fuzzy C Variety (FCV) techniques [15]. They also search for a solution within a fixed framework (the number of clusters is also arbitrarily defined) and they mainly differ in the shape of the clusters employed. While FCM will search spherical or ellipsoidal clusters, the Gustafson-Kessel method will look for hyperellipsoidal clusters and the FCV for linear variety clusters.

In the second group of techniques, the data space is not arbitrarily divided into a fixed number of clusters, but new clusters are added iteratively. A set of methods derived from the *Mountain Clustering* method [16], considers the density of the samples in variable space. The highest density locus defines the center of a cluster and membership functions are

defined by the *distance* of the samples from that cluster center. The density of each sample X_k is given by:

$$D(X_k) = \sum_{i=1}^{N} \exp(-\frac{\|X_k - X_i\|^2}{(r_a/2)^2}) \tag{35}$$

where r_a is the radius of the neighborhood outside which the data do not contribute significantly to the density measure. The center of the cluster X_{V_1} is defined by $\max_k D(X_k)$. The next cluster is defined similarly but on a sample space from which the samples with high membership have been removed. The new density is obtained by:

$$D^{new}(X_k) = D^{old}(X_k) - D^{old}(X_{V_1}) \exp(-\frac{\|X_k - X_{V_1}\|^2}{(r_b/2)^2}) \tag{36}$$

where r_b is another radius constant taken smaller than r_a to ensure that the first density center has a reduced new density and will not be retained. The process is reiterated until no significant density center is found. The membership matrix U, is then defined by:

$$u_{i,k} = \exp(-\frac{\|X_k - X_{V_i}\|^2}{(r_a/2)^2}) \tag{37}$$

Of course, the number of the clusters will partially depend on the *threshold distance* that decreases with the number of clusters. The main advantage of that method over FCM is that the clusters are relatively independent of the sample distribution. In such an approach, the samples on the *border* are not underestimated because of the more numerous samples in the central region. The clusters are then more representative of the functional distribution rather than the empirical distribution. The drawback compared to FCM is that it is more computationally intense.

3) Causality and Cluster Projections: As already mentioned, the clustering techniques are used to build fuzzy rules by performing the *projection* of output clusters on the input space. Using the probabilistic approach where $P(\varphi, y)$ defines the joint probability of having φ when y is observed, we extend the concept to clusters. Locally, the joint probability refers to the causality of a given φ_k to a given y_k (φ_k being past observations).

Given that the task of the model is to express that joint probability, it is possible to subdivide the task using local models. The local models will represent the behavior of the system in different regions.

The main problem with local models is to extract the relevant corresponding subsamples in the output and input spaces. This will be performed by *projection*. Let $C_i^{(y)}$ be a cluster on the output space and $u_{i,k}^{(y)}$ the element of the matrix U defining membership function of the samples y_k to the cluster $C_i^{(y)}$. The projection of that cluster on the input space is defined as the cluster $C_i^{(\varphi)}$. The membership of the input samples on the projected cluster $u_{i,k}^{(\varphi)}$ is defined by:

$$u_{i,k}^{(\varphi)} = u_{i,k}^{(y)} \tag{38}$$

The problem is then to identify and approximate the projected cluster. The various uncertainties in the model will, in general, lead to *complex* projections. For inputs which are not causal nor correlated to causal input, the projection will not be coherent and appear as noise. Such a phenomenon can be used to further reduce the dimension of the input space. For others, the different levels of causality will produce clusters more or less difficult to identify. The complexity may lead to numerous clusters that will eventually have to be merged.

It is important to note that the use of preprocessing techniques such as PCA will enhance the performance of the clustering when performed on the input space. If the inputs are ordered according to their variance, the first inputs will have the highest correlations with output and the highest densities. As the component order increases, that correlation decreases and the identification is more difficult. The reduced dimensionality also contributes to a reduction in the computational load.

Once the clustering has been performed, the results can be used in different ways to improve the model performance.

4) Local Models and Bootstrap Construction: Local models are used to reduce the complexity of the modelling by performing local approximations. The simplest local models are sets of linear approximations. Depending on the mapping problem, such an approach can provide better performance and almost always a reduced model complexity (smaller number of parameters) overall. We will see in Section VIII that local models can also be nonlinear.

Whatever the model structure chosen for local models, it is important to perform the parameter identification with data sets which are relevant to the local approximation. With the projection approach, it is possible to resample the data set on the basis of the correlation with an output locus. If $C_i^{(y)}$ is the output cluster related to the output set point then we can define and D_{C_i} to be the data corresponding to the cluster C_i. The selection of the input-output samples is defined by a threshold on the probability:

$$P((\varphi_k, y_k) \in D_{C_i}) = P(\varphi_k \in C_i^{(\varphi)}) P(y_k \in C_i^{(y)}) \tag{39}$$

The identification of the local model is then performed using D_{C_i}.

The clustering of the input and output spaces can also be used in another resample method, the statistical bootstrap, examined in Section III-C. The main difficulty there is to obtain small but representative sets. We have already mentioned the methods based on statistical analysis and model performance criteria. All these methods are based on bootstraps obtained by drawing samples at random. Because of the empirical distribution available, this lead to uncertainty on the distribution of the bootstraps.

An improved method uses the clusters to improve the representativeness of the distribution. Rather than selection by random draw, the bootstrap can be obtained by drawing at random within each cluster, each cluster being evenly represented in the bootstrap. Such a construction leads to smaller bootstraps and requires less of them which, in turn, reduces the computation load required. This is particularly important in the parameter optimization processes.

VII. Parameter Optimization

A. Topology Determination

Topology determination is probably the most difficult and yet the most important issue of modelling. We have already mentioned in Section IV that it is important to match the complexity of the model to the complexity of the data. Having selected the optimal regressor and performed preprocessing, the complexity of the input space is greatly reduced, giving a reduction in the complexity of the mapping. This enables a less complex parameter vector to be identified but also reduces the uncertainties associated with the identification of high dimension parameter vectors.

The concepts of optimal topology as well as the optimal parameter vector can be related to a probabilistic problem under certain conditioning. In fact, the topology is evaluated using estimates of the generalization properties of the identified model. The estimation of the topology will therefore be biased by the parameter optimization. The main approaches in system identification are *trial and errors* or *regularization methods*. The approach to our methodology relates to the latter.

Another important point is that the conditioning of the topological issue may lead to not one but a set of answers, all relevant in certain contexts. This will lead to the concept of *most likely topology* from which a distribution of models with different topologies will be derived. Practically, the topology determination is performed in two distinct stages.

Our method consists of using models with known over-capacity and then analyzing the redundancies within it. The over-sized model is obtained by verifying that it can be trained to convergence. If not, the complexity must be increased until convergence is obtained.

When an initial over-sized model is defined, we train to convergence a set of models using different statistical bootstraps. The optimal topology is obtained by pruning elements on the basis of the redundancies detected by the PCA method. We examine here the general case of multi-layered models.

In Section II, the output of the first layer is defined by $\varphi^{(2)}$. Let $\tilde{S}_1$ be the covariance matrix of $\varphi^{(2)}(t)$ in the orthonormal base defined by A_1.

$$\tilde{S}_1 = A_1 E(\varphi^{(2)}\varphi^{(2)t})A_1^t \tag{40}$$

The optimal number of elements in the first layer K_1 will be defined by the number of elements necessary to *transmit* 99% of the variance contained in $\varphi^{(2)}$.

$$K_1 = \max_{k_1}(P_{V,\tilde{S}_1}(k_1) \leq 99\%) \tag{41}$$

Once K_1 is determined, the model with an optimal first layer is trained to convergence. We define $\varphi^{(3)}$ to be the output of the second layer.

$$\varphi_l^{(3)} = g_l(\sum_{k=1}^{K_1} \alpha_{k,l}\varphi_k^{(2)}, \beta_l, \gamma_l) \tag{42}$$

$$K_2 = \max_{k_2}(P_{V,\tilde{S}_2}(k_2) \leq 99\%) \tag{43}$$

Where $\tilde{S}_2$ is the covariance matrix of $\varphi^{(3)}$ in the orthonormal base defined by A_2. The process is reiterated on all the layers. When the process is performed on a set of statistical bootstraps, the *optimal topology* is not constant but follows a distribution. The characteristics of that distribution (in the case of a Gaussian distribution, the mean and the standard deviation) are used to find an upper bound on the topology.

B. Local and Global Optimization

There are numerous numerical methods to perform the global minimization of a model performance criterion. Among these, we can emphasize methods such as *Simulated Annealing* (SA), *Evolutionary Algorithm* (EA),*Cascade Correlation with Multi-Starts* (CASCOR-MS), *Gradient Descent with Multi-Starts* (GRAD-MS) or *Truncated Newton's Method with Multi-Starts* (TN-MS) [18].

The parameter optimization is a part of the topological determination. In general, the parameter identification cannot be performed analytically so that numerical search procedures from the field of nonlinear optimization must be used to minimize a performance criterion $\hat{V}$, usually a model fit measure. Using our notation, $\hat{V}$ can be defined by:

$$\overline{V}(\theta) = E\|g_0(\varphi(t) - g(\varphi(t), \theta)\|^2 \tag{44}$$

For stationary, or supposed so, processes we have:

$$\overline{V}(\theta) = \lim_{N\to\infty} \sum_{t=1}^{N} \|g_0(\varphi(t)) - g(\varphi(t), \theta)\|^2 \tag{45}$$

Let $D_{\hat{\mathcal{F}},b}$ be the statistical bootstrap used for the parameter optimization, then the optimization estimates is obtained with :

$$\min_\theta V(\theta, D_{\hat{\mathcal{F}},b}) = \min_\theta(\frac{1}{N}\sum_{t=1}^{N} \|y(t) - g(\varphi(t), \theta)\|^2) \tag{46}$$

θ is related to the mapping function $\mathcal{A}$ of complexity class $\mathcal{H}_d$, so that the parameter vector θ^* will be conditioned by $\mathcal{C} = \{\mathcal{H}_d, \mathcal{A}, D_{\hat{\mathcal{F}},b}, \varphi\}$. In the case of non-deterministic numerical optimization, the solution found will depend on the optimization method and its parameterization, especially parameters such as initial conditions in gradient-descent based method. If we define $\mathcal{L}$ as the optimization method, $V_\mathcal{L}$ its parameters and θ_0 the initial conditions, the model estimates are then expressed by:

$$\hat{\theta} = \arg(\min_\theta V(\mathcal{H}_d, \mathcal{A}, D_{\hat{\mathcal{F}},b}, \varphi, \mathcal{L}, V_\mathcal{L}, \theta_0)) \tag{47}$$

Equation (47) expresses the conditioning of the solution. As a consequence, $\hat{\theta}$ will be a local minimum of $V(\theta)$. To simplify the notation, we will refer to $\mathcal{C} = \{\mathcal{H}_d, \mathcal{A}, D_{\hat{\mathcal{F}},b}, \varphi, \mathcal{L}, V_\mathcal{L}, \theta_0\}$ as the conditioning of the solutions. The more severe the conditioning, the less general the model.

A set of techniques, combined together, are used in order to reduce the conditioning. The idea is to use an *ensemble* approach. A set of solutions obtained by different conditioning will be distributed *around* the global minima. We define $\hat{\theta}_i$ to be the solution obtained with conditioning $\mathcal{C}_i$.

Let $\mathcal{F}_\mathcal{C}$ be the distribution of the possible condition. The parameter distribution $\mathcal{F}_\theta$ is directly conditioned by $\mathcal{F}_\mathcal{C}$. The dimension of $\mathcal{C}$ is too large to to work with an exhaustive

condition set. In order to obtain a representative parameter distribution, we will work on a restriction defined by $\hat{\mathcal{F}}_{\mathcal{C}}$. The computation limitation requires us to reduce the dimension of the conditioning. This is obtained using a combination of techniques already described in previous sections.

The conditioning introduced by the gradient descent parameter optimization algorithms will be reduced by multi-start optimization and early stop methods supervised by statistical bootstrap generalization estimates (that addresses the bias-variance trade-off problem). The topological conditioning will be overcome using elements of a topological distribution obtained using regularization methods such as PCA, neuron pruning and statistical weight pruning methods. Finally, the data set conditioning is reduced by using model ensembles obtained on statistical bootstraps. The representativeness of the bootstraps is ensured using clustering methods.

All these techniques contribute to a representative realization of the parameter distribution. That realization is defined to be $\hat{\mathcal{F}}_{\theta}$. If the realization is representative enough, the center of the distribution will be the optimal parameter θ^*, the global minimum of the model performance surface. When such a distribution is obtained, two approaches are possible. The first one builds an *averaged model* obtained by estimating the parameter averaged over the distribution. Unfortunately, since we deal with topological distribution, such an averaging is not possible. The second approach combines the predictions of the model distribution, rather that the parameters. If the model/parameter distribution realization $\hat{\mathcal{F}}_{\theta}$ is uniform then the model error will be uniformly distributed as well. Such approach is called the *ensemble* or *committee* method and is detailed in Section VIII.

Basically, the method used attempts a hypothetical global minimum by applying statistical methods to local optimization techniques. This approach is not as computationally intensive as global minimum search algorithms already mentioned. However, as we will see in Section VII-C, the parameter distributions can be used to implement more efficiently global optimization methods.

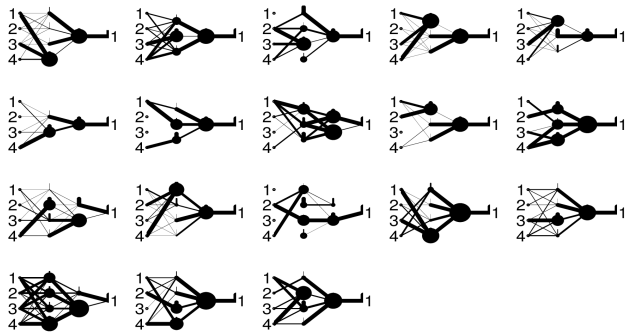

Fig. 2. Committee of Local Models.

C. Evolutionary Optimization

Evolutionary Algorithms (EA) are mainly used in nonlinear, multiobjective optimization problems [19]. In our black-box modelling methodology, EA methods optimized for neural network architectures are used to tune the sub-committee models with respect to an increasing horizon of prediction criteria. This approach is made possible by the hybrid structure which reduces the complexity of the individual models and consequently the dimension of the parameter space. Statistical methods are used to obtain broad estimate of the parameter distribution which is used to optimize the evolution procedure.

Practically, the general exploratory features of EAs, the cause of the computationnal load, are reduced by assigning different cross-over and mutation probabilities to the parameters on the basis of the parameter distribution and architectural features such as *Critical Paths*. Such determinism greatly accelerate the convergence of the algorithm. The probability of creating relevant outspring is increased by targeting the relevent parameters but also by quantifying the level of the modification to each parameter. As a consequence, the size of the population required to ensure the convergence of the algorithm is reduced as well as the computation times.

VIII. Ensemble Methods

A. Introduction

In the previous sections, we have seen that black-box modelling requires the determination of an *optimal model*, i.e. the best possible model given the model conditioning. Because of the conditioning, the resulting model is a *local* solution (in the parameter space). Using some results from *Bayesian modelling* [10], it is possible to consider that under certain conditions these local solutions are members of a solution population having an unknown but continuous distribution. In that approach, the optimal model is the most probable.

Figure 2 represents a set of neural network models obtained with various conditioning. An important result is that when the dependencies are minimized, i.e. when the effects of the conditioning are reduced and uniformly distributed so that the local solution are in the optimal solution neighborhood, the hypothetical continuous model distribution leads to local solution uniformly distributed around the optimal solution. In turn, the model estimates are also uniformly distributed around the optimal estimate. When the topology and the parameter role are fixed, it is possible to obtain a better model by *averaging* the models [20]. For various reasons, starting with a topological distribution in the set of solution models, we will perform the combination of estimates to improve the model performance.

B. Combination of Estimates

In previous sections, we have described how various techniques can be used to improve the generalization performance of the model. The outcome is a set of models which are local solutions to the problem, that must be combined to produce estimates. [20] shows how linear combinations and the statistical analysis of the set of estimates can produce better estimates and gives an upper bound for the confidence limits on these estimates.

When neural networks are used, the resulting set of models is called a *Network Committee*. The estimates are nonlinearly combined using a *neural network combiner*. The committee

structure enables further improvements such as the local models described in Section VI-C4. With local models, the complexity of the problem is divided and distributed. This leads to smaller models, i.e. with a lower dimension parameter space on which computationally intense optimization methods such as EA can be applied. Just as individual models can be local models, they can also be optimized for various criteria, leading to a simple implementation of multicriteria optimization. The local model committees are structured into subcommittees to simplify the interpretation and the meaning of the overall model.

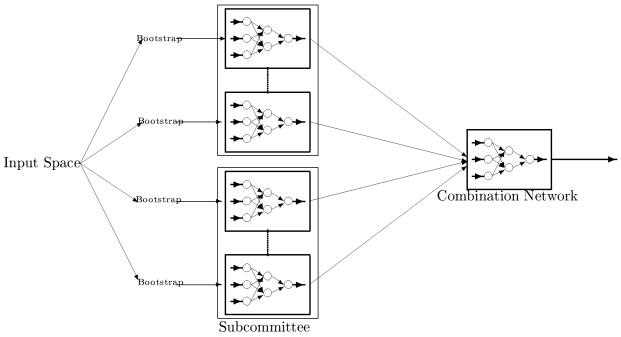

Fig. 3. Representation of the Local Model Structure.

Finally, when multi-step ahead prediction is necessary, the combined estimate error will be reduced and uncorrelated to the individual estimate error which will improve the performance of the multi-step ahead prediction. [3] presents the main theoretical results and the way in which the ensemble techniques are implemented using the statistical techniques mentioned in the previous sections.

IX. Illustrative Examples

In order to illustrate the black-box modelling methodology, we will examine its application to a chaotic time series. The chosen time series was used in a prediction competition [21] so that comparisons with the results from recent techniques are available. The data are related to a 5-scrolls attractor generated from a generalized Chua circuit. The time series shown in Figure 4 consists of a set of 2000 points and the next 200 points have to be predicted. The performance criterion used is the MSE.

The analysis of the time series [22] shows that it is noise free and chaotic, with three unstable equilibria at approximately -0.25, 0 and 0.25 and is also symmetrical (about the horizontal axis). The symmetry was used to generate another 2000 points. To improve the accuracy of the models, the time series was upsampled (interpolated) by a factor of two.

The model structures are NOE with a 200 sample regression order. The order was obtained from correlation and frequency analysis as linear models did not provided reliable estimators. The preprocessing of the 200 regressors consisted of applying PCA to obtain an uncorrelated input space. The analysis suggests retaining only the first 17 components accounting for 99.9% variance. The identification of dominant regions in the output space was carried out on the first component using

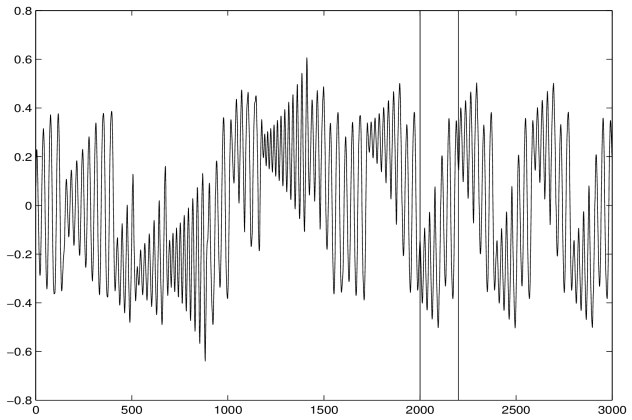

Fig. 4. Competition data (before vertical line) and data to be predicted (between the vertical lines).

fuzzy clustering methods (FCM) and confirmed the relevance of the three equilibria mentioned in [22].

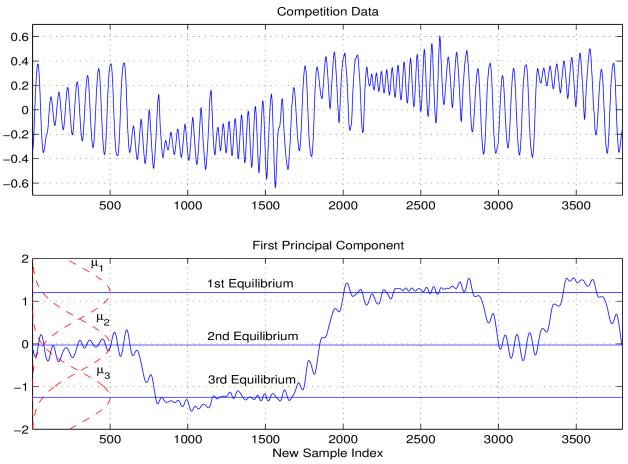

Fig. 5. Resampled Time Series, 1st Principal Component and Membership Functions.

Figure 5 shows the resampled time series and the 1st principal component. The corresponding Gaussian membership functions μ_1,μ_2 and μ_3 are used to draw uniformly distributed statistical bootstraps containing samples in each particular equilibrium. Again, the performance of linear models was insufficient to be used in the determination of the optimal statistical bootstrap characteristics so that alternate methods such as statistical and frequency analysis (wavelet analysis) had to be used instead. The *local bootstraps* were then used to train and evaluate the performance of sub-committee local neural network models. At each step of the modelling, the maximum reliable horizon of prediction was determined and used to estimate the performance of the next stage models.

As a result of this modelling procedure, the global structure is defined by a preprocessing unit followed by three subcommittees of six neural network models and three committee combiners providing the prediction for the local models. The local model predictions are then combined by the *global combiner* on the evidence of the first principal component

membership level. The structure of the global model is shown on Figure 6.

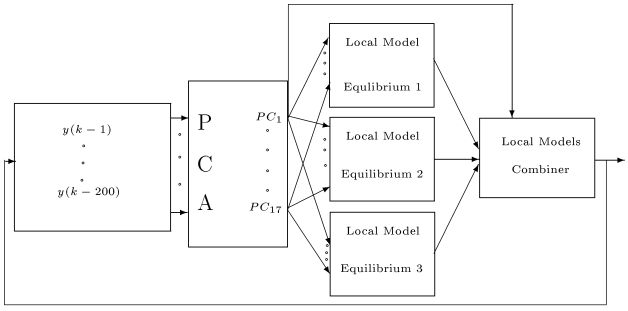

Fig. 6. Representation of the Model Structure.

The optimal statistical bootstrap characteristics are defined locally using the multi-step ahead prediction performance of linear models. For each region, a committee of 50 neural networks (MLPs) is trained with bootstraps containing 300 samples, the data set being made of the original time series and an additional symmetrical time series. The bias-variance trade-off is obtained by implementing an early-stop backpropagation algorithm supervised by statistical bootstrap generalization estimates. The networks are then pruned by statistical neuron and weight elimination methods. Only the best six local solutions, i.e. with the best multi-step ahead prediction generalization properties are retained. In a final refinement, the models are optimized using EA methods and multi-step ahead prediction criteria, the horizon of prediction increasing as performance allows. The final model is then composed of 18 sub-committee neural networks, three committee combiners and one global combiner.

When the identification is complete, the 200 step ahead prediction of the time series is performed. Figure 7 shows the comparison between the predictions and the actual time series while Table I shows the MSE on the 200 predicted points and comparisons with the first three best submissions to the competition.

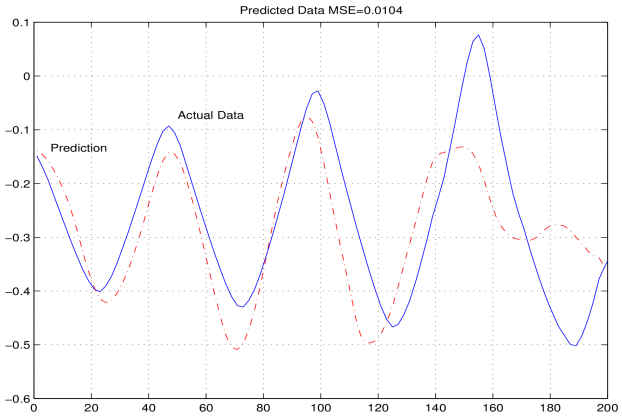

Fig. 7. Predicted Data and Actual Data.

Method	MSE (H=200)
Mc Names	0.0018
Bersini	0.0475
Bakker	0.0645
Our Method	xxxx

TABLE I
COMPARISON OF THE PREDICTION WITH THE BEST SUBMISSIONS TO THE COMPETITION

The method has also been successfully applied to other problems. Here is the comparison for the Sunspot time series:

Model	Training 1700-1920	Test 1921-1955	Validation 1956-1979
Tong and Lim	0.097	0.097	0.28
Weigend et al.	0.082	0.086	0.35
Svarer Linear	0.132	0.13	0.37
Svarer Pruned	$0.09^{0.001}$	$0.082^{0.007}$	$0.35^{0.05}$
Network Committee	$0.096^{0.002}$	$0.104^{0.032}$	$0.169^{0.0019}$
Hybrid Neural Network	0.09	0.085	0.157

TABLE II
COMPARISONS OF THE NORMALIZED ERROR FOR SINGLE-STEP AHEAD FORECASTING.

X. CONCLUSIONS

The aim of that paper was to introduce a modelling methodology in a general black-box modelling context. The success of the method will depend on the analysis of the problem and the choice of the appropriate preprocessing which lean towards the concept of grey-box modelling. We presented a hybrid neural network model structure motivated by results from the field of Bayesian modelling. Such an approach offers solutions to the major neural network modelling problems such as topology and parameter determination. Although the method can potentially be applied to any black-box modelling problem, the emphasis was set against multi-step ahead prediction requirements. In Figure 8, a brief flow chart describing the main steps of the method is given.

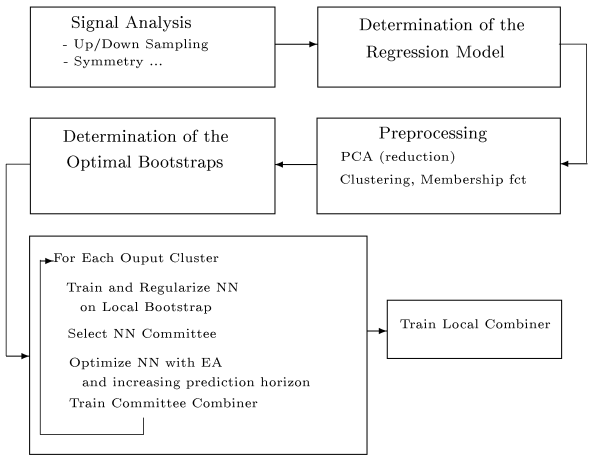

Fig. 8. Flowchart of the modelling Method.

REFERENCES

[1] J. Sjöberg, Q. Zhang, L. Ljung, A. Benveniste, B. Delyon, P. Glorennec, H. Hjalmarson, A. Juditsky, Non-Linear Black-box modelling in System Identification : A Unified Overview, *Automatica*, Vol. 31, No 12, pp.1691-1724, 1995.

[2] L. Ljung, *System Identification : Theory for the User*, *Prentice-Hall*, 1987.

[3] G-M. Cloarec, Statistical Methods for Neural Networks Predictions Models, *Research Report EE/JVR/97/2*, Control Systems Group, School of Electronic Engineering, Dublin City University, Ireland, 1997.

[4] M. Vidysagar, Statistical Learning in Control and Matrix Theory, *Nonlinear modelling : Advanced Black-Box Techniques*, Chap. 7, pp177-207, Kluwer Academic Publishers, 1998.

[5] F. Snijder, R.Babuška, M. Verhaegen, Finding the Bias-Variance Tradeoff During Neural Network Training and Its Implication On Structure Selection, *Proceedings of the International Joint conference on Neural Networks (IJCNN'98), pp1613-1618, 1998.*

[6] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, *Chapman & Hall*, 1993.

[7] A.C. Rencher, Principal Component Analysis, Chap. 12, *Methods of Multivariate Analysis*, Wiley Series in Probability and Mathematical Statistics, 1995.

[8] C. Goutte, J. Larsen, Optimal Cross-Validation Split Ratio : Experimental Investigation,
Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN'98), Perspectives in Neural Computing, Berlin: Springer Verlag, pp. 681-687, 1998.

[9] W.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, 1995.

[10] D.J.C. Mackay, Probable Networks and Plausible Predictions : A Review of Practical Bayesian Methods for Supervised Neural Networks, *Commissioned review article in Network (IOPP),1995.*

[11] J. Moody, Prediction Risk and Architecture Selection for Neural Networks, *Statistics Applied to Neural Networks and Pattern Recognition Application*, NATO AS1 series F, Springer-Verlag, 1994.

[12] S. Lawrence, C.L. Giles, A.C. Tsoi, What Size Neural Network Gives Optimal Generalization ? Convergence Properties of Backpropagation., *Technical Report UMIACS-TR-96-22 and CS-TR-3617*, Institute for Advanced Computer Studies, University of Maryland, MD, 1996.

[13] L.K. Hansen, J, Larson, F. Torben, Early Stop Criterion from the Bootstrap Ensemble, *Proceedings of ICASSP'97*, Vol. 4, pp3205-3208, Munich, Germany, 1997.

[14] J.C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithm, Plenum Press, 1981.

[15] R. Babuška, Fuzzy modelling and Identification, *Ph.D. Dissertation*, Delft University, The Netherlands, 1996.

[16] R.R. Yager, D.P. Filev, Approximate Clustering Via the Mountain Method, *IEEE Tran. Systems, Man and Cybernetics*, Vol. 24, pp.1279-1289, 1994.

[17] T. Takagi, M. Sugeno, Fuzzy Identification of Systems and its Application to modelling, *IEEE transactions on Systems, Man and Cybernetics*, Vol. SMC-15, No 1, pp 116-132, Jan. 1985.

[18] Y. Shang, B.W. Wah, Global Optimization for Neural Network Training, *IEEE Computer*, Vol. 29, pp. 45-54, March 1996.

[19] K. Balakrishnan, V. Honavar, Evolutionary Design of Neural Architectures, A Preliminary Taxinomy and Guide to Literature, *Research Report CS TR #95-01*, Artificial Intelligence Research Group, Department of Computer Science, Iowa State University,1995.

[20] M.P. Perrone, L.N. Cooper, When Networks Disagree : Ensemble Methods for Hybrid Neural Networks, *Artificial Neural Networks for Speech and Vision*, Chap. 10, pp. 126-142, Chapman & Hall, 1993.

[21] J.A.K. Suykens, J. Vanderwalle, The K.U. Leuven Time Series Prediction Competition, *Nonlinear modelling : Advanced Black-Box Techniques*, Chap. 9, pp241-253, Kluwer Academic Publishers, 1998.

[22] J. Mc Names, A Nearest Trajectory Strategy for Time Series Predictions, *Proceedings of the International Workshop on Advanced Black-Box Techniques for Nonlinear modelling*, pp 112-128, Leuven, Belgium, 1998.