CHAOS THEORY APPLIED TO INPUT SPACE REPRESENTATION OF AUTONOMOUS NEURAL NETWORK-BASED SHORT-TERM LOAD FORECASTING MODELS

Vitor Hugo Ferreira[*]
vitor@vm.uff.br

Alexandre Pinto Alves da Silva[†]
alex@coep.ufrj.br

[*]Electrical Engineering Department,
Fluminense Federal University (UFF)
Rua Passo da Pátria, 156, Sala 509, Bloco D
CEP 24210-240 – Niterói RJ

[†]Electrical Engineering Program, PEE-COPPE, Federal
University of Rio de Janeiro (UFRJ)
P.O. 68504
CEP 21945-972 – Rio de Janeiro RJ

RESUMO

Teoria do Caos Aplicada à Definição do Conjunto de Entradas de Modelos Neurais Autônomos para Previsão de Carga em Curto Prazo
Após 1991, a literatura sobre previsão de carga passou a ser dominada por propostas baseadas em modelos neurais. Entretanto, um empecilho na aplicação destes modelos reside na possibilidade do ajuste excessivo dos dados, i.e, overfitting. O excesso de não-linearidade disponibilizado pelos modelos neurais de previsão de carga, que depende da representação do espaço de entrada, vem sendo ajustado de maneira heurística. Modelos autônomos incluindo técnicas automáticas e acopladas para seleção de entradas e controle de complexidade dos modelos foram propostos recentemente para previsão de carga em curto prazo. Entretanto, estas técnicas necessitam da especificação do conjunto inicial de entradas que será processado pelo modelo visando determinar aquelas mais relevantes. Este trabalho explora a teoria do caos como ferramenta de análise não-linear de séries temporais na definição automática do conjunto de atrasos de uma dada série de carga a serem utilizados como entradas de modelos neurais autônomos. Neste trabalho, inferência Bayesiana aplicada a perceptrons de múltiplas camadas e máquinas de vetores relevantes são utilizadas no desenvolvimento de modelos neurais autônomos.

PALAVRAS-CHAVE: Previsão de carga, Redes Neurais Artificiais, Seleção de Entrada, Teoria do Caos, Sincronização caótica, Inferência Bayesiana, Perceptron de Multi-camadas , Máquinas de Vetores Relevantes.

ABSTRACT

After 1991, the literature on load forecasting has been dominated by neural network based proposals. However, one major risk in using neural models is the possibility of excessive training, i.e., data overfitting. The extent of nonlinearity provided by neural network based load forecasters, which depends on the input space representation, has been adjusted using heuristic procedures. The empirical nature of these procedures makes their application cumbersome and time consuming. Autonomous modeling including automatic input selection and model complexity control has been proposed recently for short-term load forecasting. However, these techniques require the specification of an initial input

Artigo submetido em 03/03/2011 (Id.: 01285)
Revisado em 06/05/2011, 14/08/2011
Aceito sob recomendação do Editor Associado Prof. Carlos Roberto Minussi

set that will be processed by the model in order to select the most relevant variables. This paper explores chaos theory as a tool from non-linear time series analysis to automatic select the lags of the load series data that will be used by the neural models. In this paper, Bayesian inference applied to multi-layered perceptrons and relevance vector machines are used in the development of autonomous neural models.

KEYWORDS: Load Forecasting, Artificial Neural Networks, Input Selection, Chaos Theory, Chaotic Synchronization, Bayesian Inference, Multi-layered Perceptron, Relevance Vector Machines.

1 INTRODUCTION

The decision making process in power systems, including economic dispatch, hydrothermal coordination, automatic generation control, energy trading and so on, requires the knowledge of the future behavior of the load dynamics. Along the last two decades, many load forecasting models have been proposed, with the neural network based models receiving great attention. This is because they have been showing superior prediction performance, specially for short-term applications (Hippert, et. al., 2001). In fact, the neural network based models have been presenting outstanding results for multivariate problems envolving databases with huge cardinality, as the short-term load forecasting problem (Ferreira and Alves da Silva, 2007), (Ferreira and Alves da Silva, 2009) and (Ferreira and Alves da Silva, 2010). Even been more robust than traditional models, critical questions like input space representation and complexity control of neural network have not received the necessary attention.

The input selection stage is the one of the most important tasks in the development of load forecasting models. Feature extraction via non-linear techniques like wavelets uses only information about the time-series to be predicted without direct concern with the forecasting accuracy. In this sense, an input selection methodology directly related with the neural network model is required. The methods that use the model itself in the input selection step are called wrapper methods, and the ones that consider only the dynamics and statistics of the time-series are called filter methods (Guyon and Elisseeff, 2003). For forecasting purposes, the wrapper methods are more indicated since these techniques aim to select the inputs that are most sutiable to the model in terms of forecasting performance.

The complexity control of neural models has the objective of adjusting the non-linear extent of the neural network to the regularity exhibited by the data. This step is necessary to avoid the harmful modeling of the noisy component of the data, named overfitting. This can compromise the generalization capacity of the neural model, i.e., good predictions for unseen data.

Autonomous neural forecasting models, including automatic input selection, complexity control and structure selection, are necessary to reduce the necessity of intervention from experts. These automatic procedures allow the extension of the forecasting to the bus load level. Autonomous Neural Network Load Forecasting models have been proposed in the literature (Ferreira and Alves da Silva, 2007) using Bayesian Inference Applied to Multi-Layered Perceptrons (BIAMLPs) and Support Vector Machines (SVMs) training and specification. These procedures include automatic and coupled procedures for input selection, complexity control and model specification. However, these procedures still require the definition of an initial set of inputs.

In order to improve the autonomous capability of the models proposed in (Ferreira and Alves da Silva, 2007), techniques for automatic definition of the initial set of inputs from the available time-series are necessary. This paper investigates the application of Chaos Theory as a tool for automatic definition of the initial set of inputs to be used with the autonomous neural models proposed in (Ferreira and Alves da Silva, 2007). The BIAMLPs are used in this paper and they are compared with Relevance Vector Machines (RVMs). Being a sparse kernel model, a RVM can be seen as a SVM derived from the application of Bayesian Inference. The forecasting performance of the models are compared using three public load and temperature databases. The main contributions of the paper can be summarized as follows:

a) proposal of an automatic method for selecting inputs of neural network load forecasting models, based on time-series and calendar information, only; and

b) evaluation of the applicability of RVMs to the load forecasting problem.

This paper is organized as follows. In Section 2, Chaos Theory is presented in the context of Input Space Reconstruction. BIAMLPs are described in Section 3. Section 4 is devoted to the description of the RVMs. The database description and results are shown in Section 5. The discussion, main conclusions and future work are presented in Section 6.

2 CHAOS THEORY

The Chaos Theory development is motivated by the study of dynamical systems sensitive to initial conditions. After the transient effects, a dynamical system $F\left(\underline{X}\right) : \mathbb{R}^D \rightarrow \mathbb{R}^D$ evolving in a state space $\underline{X} \in \mathbb{R}^D$ can be defined by the following expression:

$$\underline{X}(k+1) = F\left[\underline{X}(k)\right] \tag{1}$$

From the actual state $\underline{X}(k)$, all of the adjacent states of the deterministic system described by equation (1) can be obtained. The sensitivity to the initial conditions makes the trajectory of the system dependent on the knowledge of the function $F(\underline{X})$ and the value of the initial state. The set of initial conditions that drives asymptotically the system to a given region of the space is called basin of attraction, and the region where the system is driven is named attractor (Kantz and Schreiber, 1997).

2.1 TAKENS THEOREM

The above definitions are valid in the multidimensional space where the system $F(\underline{X})$ is confined. However, in practice, only scalar measures $x(k)$, $k = 1, 2, ..., N$, are avaliable through a measurement function $s(\underline{X}) : \mathbb{R}^D \to \mathbb{R}$, i.e.,

$$x(k) = s\left[\underline{X}(k)\right] + \eta(k) \tag{2}$$

where $\eta(k)$ represents the measurement noise.

The measurement function $s(\underline{X})$ comprises the multivariate information contained in $\underline{X}(k)$ in a scalar measure $x(k)$, projecting non-observable variables of the system in a real scale. Since $s(\underline{X})$ is unknown, in the presence of measurement noise $\eta(k)$, the perfect reconstruction of $\underline{X}(k)$ from a set of measures $x(k)$ is impossible. However, the perfect estimation of the original space is unnecessary, being sufficient the definition of a new representation space with a equivalent attractor (Takens, 1981). Called embedded space, this space can be obtained from the equation:

$$\underline{x}(k) = \left[x(k), x(k-\tau), \ldots, x(k-(d-1)\tau)\right]^t \tag{3}$$

where τ and d are parameters named delay and embedding dimension, respectively.

Takens' Theorem (Takens, 1981) defines the conditions for which the attractor in the embedded space $\underline{x} \in \mathbb{R}^d$, given by equation (3), is equivalent to the attractor in the original space $\underline{X} \in \mathbb{R}^D$. In case of unlimited data, noise free and assuming the existence of a mapping $\mathrm{Z}(\underline{x}) : \mathbb{R}^d \to \mathbb{R}^D$ and the corresponding inverse mapping $\mathrm{Z}^{-1}(\underline{X}) : \mathbb{R}^D \to \mathbb{R}^d$, both smoth, continuous, bi-unique and continuously differentiable, $\underline{x} \in \mathbb{R}^d$ will be a immersion of $\underline{X} \in \mathbb{R}^D$ if $d > 2D$ for τ arbitrarily chosen. While the Takens' Theorem devotes attention only to the embedding dimension d, in practical applications the choice of the embedding delay τ is also vital for the definition of the embedded space (Abarbanel et.al., 1993).

There are many criteria proposed in the literature for the definition of τ, including techniques based on geometrical and statistical foundations, with the statistical ones been more used and suitable for time-series applications (Kantz and Schreiber, 1997). Among the statistical criteria, the analysis of the autocorrelation function of $x(k)$, $r_{xx}(k)$, is the simplest technique. In order to pursue a trade-off between attractor compression and reconstruction based on almost uncorrelated directions, the first minimum of the absolute value of $r_{xx}(k)$, $|r_{xx}(k)|$, can be used as an estimate for τ. Although simple, generally the definition of τ based on the analysis of $r_{xx}(k)$ does not avoid the attractor collapse, since non-linear interdependences can fold the attractor along trajectories of this nature.

Information Theory provides indices for the evaluation of general relationships (linear or non-linear) among random variables. The mutual information, $I_x(r)$, measures the degree of information that $x(k-r)$ gives about $x(k)$, i.e., the reduction of uncertainty about $x(k)$ due to the knowledge of $x(k-r)$. Using variable discretization to estimate the required probabilities, $I_x(r)$ is given by:

$$I_x(r) = H_x(0) + H_x(r) - H_{xx}(r) \tag{4}$$

$$\begin{aligned}
&H_x(r) = -\sum_{i=1}^{p} P\left[x(k-r) \in \nu_i\right] \times \log P\left[x(k-r) \in \nu_i\right] \\
&H_{xx}(r) = -\sum_{i=1}^{p}\sum_{j=1}^{p} P\left[x(k) \in \nu_i, x(k-r) \in \nu_j\right] \times \\
&\log P\left[x(k) \in \nu_i, x(k-r) \in \nu_j\right]
\end{aligned} \tag{5}$$

where p represents the number of intervals in the discretization; $P\left[x(k-r) \in \nu_i\right]$ is the marginal probability of $x(k-r)$ in the ν_i interval; $P\left[x(k) \in \nu_i, x(k-r) \in \nu_j\right]$ the joint probability of the discretized $x(k)$ and $x(k-r)$. Similarly to the analysis of $r_{xx}(k)$, the first minimum of $I_x(r)$ can be used as an estimate of τ (Fraser and Swinney, 1986).

The literature about the estimation of the embedding dimension d shows several techniques based on the calculation of invariant features of the attractor (Kantz and Schreiber, 1997) and (Abarbanel et.al., 1993). Besides being computationaly intensive, these techniques are very subjective, requiring constant intervention of experts during the modeling stage. One of the most popular techniques for estimation of embedding dimension d is based on the identification of spurious trajectories, being known as false nearest neighbors method

(Kennel et. al., 1992). This denomination is based on the way the spurious intersections of the attractor are identified; through the observation of changes in the neighborhood of a given point due to the increase of the dimension. Neighboring points due to the system dynamics remain in this condition (neighbors) when d increases. Points that leave the neighborhood due to the dimension increase are called false nearest neighbors. These points were located in the neighborhood of the testing point because of the incomplete reconstruction of the attractor.

In order to increase the automation level of the false nearest neighbors method, Cao (Cao, 1997) proposes a practical method for estimation of d. Let $\Delta\left(i,j,d\right)$ be the distance between points $\underline{x}\left(i\right)$ and $\underline{x}\left(j\right)$, both reconstructed in the dimension d, given by:

$$\Delta\left(i,j,d\right)=\max_{l=1,\ldots,d}\left|x_l\left(i\right)-x_l\left(j\right)\right| \tag{6}$$

In equation (6), $x_l\left(i\right)$ represents the l-th element of the vector $\underline{x}_l\left(i\right)$ at instant i and $\Delta\left(i,j,d\right)$ the infinite norm of the difference between $\underline{x}\left(i\right)$ and $\underline{x}\left(j\right)$. The nearest neighbor of $\underline{x}\left(i\right)$ is the point for which $\Delta\left(i,j,d\right)$ is minimum, i.e.,

$$n\left(i,d\right)=\arg\left[\min_{j=(d-1)\tau+1,\ldots,N}\Delta\left(i,j,d\right)\right] \tag{7}$$

where $n\left(i,d\right)$ is the index of the vector $\underline{x}\left[n\left(i,d\right)\right]$ closest to $\underline{x}\left(i\right)$ in the space of dimension d, according to the $\Delta\left(i,j,d\right)$ metric.

Additionally, let $a\left(i,d\right)$ be the relation between nearest neighbors in consecutive dimensions d and $\left(d+1\right)$ given by:

$$a\left(i,d\right)=\frac{\Delta\left[i,n\left(i,d\right),d+1\right]}{\Delta\left[i,n\left(i,d\right),d\right]} \tag{8}$$

In equation (8), if $\Delta\left[i,n\left(i,d\right),d\right]$ is zero, $n\left(i,d\right)$ is replaced by the index of the next (adjacent) nearest neighbor. The mean value of $a\left(i,d\right)$ is used to define the $J\left(d\right)$ statistic:

$$J\left(d\right)=\frac{1}{N-\left(d-1\right)\tau}\sum_{i=(d-1)\tau+1}^{N}a\left(i,d\right) \tag{9}$$

The relative variation $\delta\left(d\right)$ of this statistic due to the increase on the embedding dimension d is given by:

$$\delta\left(d\right)=\frac{J\left(d+1\right)}{J\left(d\right)} \tag{10}$$

According (Cao, 1997), for time-series originated from an attractor, the variation $\delta\left(d\right)$ stabilizes when the embedding dimension d is greater than a value d_0. In other words, in dimensions above d_0 the number and location of false nearest neighbors do not change, so that $J\left(d\right)$ stops changing. Thus, the embedding dimension is given by $d=d_0+1$.

For automatic detection of the stabilization dimension d_0, let $d_{\max}$ be the maximum embedding dimension for which the statistic $\delta\left(d\right)$ is calculated, supposing that the stabilization of $\delta\left(d\right)$ occurs for $d_0<d_{\max}$. Given the pairs $\left[d,\delta\left(d\right)\right]$, $d=1,2,\ldots,d_{\max}$, a linear regression model of the evolution of $\delta\left(d\right)$ along d is estimated, i.e.,

$$\delta\left(d\right)=\kappa+\nu d+\zeta \tag{11}$$

A hypothesis test about the linear model given by is performed, at α significance level and null hypothesis defined as ν equal to zero, i.e., the angular coeficient of the model being null (Griffiths et. al., 1993). If the null hypothesis is rejected, the first pair $\left[d,\delta\left(d\right)\right]$ is removed and a new linear regression model like is estimated considering only the points $d=2,3,\ldots,d_{\max}$. This procedure is repeated until the null hypothesis can not be rejected, i.e., the hypothesis of constant $\delta\left(d\right)$ can not be discarded. Then, the stabilization point of $\delta\left(d\right)$ statistic is found, with the embedding dimension given by the first dimension used in the estimation of the linear regression model for which the null hypothesis is not rejected.

The heuristic defined above depends on the definition of two parameters, $d_{\max}$ and α. The choice of significance level α, although heuristic, is more intituive than the choice of the parameters that must be specified in other embedding dimension estimation approaches. The definition of $d_{\max}$ is directly related to computational effort. In this work, $d_{\max}=30$ and $\alpha=0.01$.

2.2 CHAOTIC SYNCHRONIZATION

Let's assume two discrete chaotic systems, an autonomous driving system $\underline{X}\in\mathbb{R}^D$ and the response system $\underline{Y}\in\mathbb{R}^R$ with dynamics given by the equations:

$$\begin{array}{l}\underline{X}\left(k+1\right)=F\left[\underline{X}\left(k\right)\right]\\ \underline{Y}\left(k+1\right)=U\left[\underline{Y}\left(k\right),\underline{X}\left(k\right)\right]\end{array} \tag{12}$$

In , $F\left(\underline{X}\right):\mathbb{R}^D\rightarrow\mathbb{R}^D$ and $U\left(\underline{Y},\underline{X}\right):\mathbb{R}^R\times\mathbb{R}^D\rightarrow\mathbb{R}^R$ represent the dynamics of the driving and response systems, respectively. These systems will be in generalized synchronism if their trajectories along their state spaces are related, i.e., a function $\varphi\left(\underline{X}\right):\mathbb{R}^D\rightarrow\mathbb{R}^R$ can be defined such that:

$$\underline{Y}\left(k\right)=\varphi\left[\underline{X}\left(k\right)\right] \tag{13}$$

Since the equations that define the functions $F\left(\underline{X}\right)$, $U\left(\underline{Y},\underline{X}\right)$ and $\varphi\left(\underline{X}\right)$ are unknown, methods for detection of synchronism based on data collected from these systems are required.

Rulkov and co-workers (Rulkov, et. al., 1995) propose a method based on the idea of false nearest neighbors for synchronism detection. Called mutual false nearest neighbors, the method assumes that the function $\varphi\left(\underline{X}\right)$ exists and it is smooth and differentiable. In this case, neighbor points in $\underline{X}$ space will be associated with neighbor points in the response system $\underline{Y}$.

Let $\underline{X}\left[n\left(i,D\right)\right]$ be the nearest neighbor of $\underline{X}\left(i\right)$. Assuming that $\varphi\left(\underline{X}\right)$ exists and that the distance between nearest neighbors in each state space is small, the aproximated relation between neighbors can be derived (Rulkov, et. al., 1995) as follows:

$$\underline{Y}\left(i\right)-\underline{Y}\left[n\left(i,D\right)\right]\approx D\left[\underline{X}\left(i\right)\right]\left\{\underline{X}\left(i\right)-\underline{X}\left[n\left(i,D\right)\right]\right\} \tag{14}$$

In equation (14), $D\left(\underline{X}\right):\mathbb{R}^D\rightarrow\mathbb{R}^R\times\mathbb{R}^D$ is the Jacobian matrix of $\varphi\left(\underline{X}\right)$. Similarly, observing the nearest neighbor of $\underline{Y}\left(i\right)$ in the state space of the response system denoted by $\underline{Y}\left[n\left(i,R\right)\right]$,

$$\underline{Y}\left(i\right)-\underline{Y}\left[n\left(i,R\right)\right]\approx D\left[\underline{X}\left(i\right)\right]\left\{\underline{X}\left(i\right)-\underline{X}\left[n\left(i,R\right)\right]\right\} \tag{15}$$

The ratio between the Euclidean norms of equations (14) and (15) is given by:

$$M\left[\underline{X}\left(i\right),\underline{Y}\left(i\right)\right]=\frac{\frac{\left\|\underline{Y}(i)-\underline{Y}[n(i,D)]\right\|}{\left\|\underline{X}(i)-\underline{X}[n(i,D)]\right\|}}{\frac{\left\|\underline{Y}(i)-\underline{Y}[n(i,R)]\right\|}{\left\|\underline{X}(i)-\underline{X}[n(i,R)]\right\|}} \tag{16}$$

If the mapping $\varphi\left(\underline{X}\right)$ exists, then the index $M\left[\underline{X}\left(i\right),\underline{Y}\left(i\right)\right]$ will be close to one for all i.

Since the original state spaces $\underline{X}$ and $\underline{Y}$ are unknown, let $\underline{y}\left(k\right)\in\mathbb{R}^r$ be the reconstructed space of the response system $\underline{Y}$ and $\underline{x}\left(k\right)\in\mathbb{R}^D$ the reconstruction of the driving system $\underline{X}$, both obtained from Takens' Theorem given by equation (3). Let $\underline{x}'\left(k\right)\in\mathbb{R}^r$ be an auxiliar reconstruction of the driving system $\underline{X}$ with embedding dimension equal to the one obtained for the response system $\underline{Y}$. The nearest neighbors in the sense of the infinite norm given by equation (6) for each point in each embedded space are calculated, with $\underline{y}\left[n\left(k,r\right)\right]$ being the nearest neighbor of $\underline{y}\left(k\right)$, $\underline{x}\left[n\left(k,d\right)\right]$ the nearest neighbor of $\underline{x}\left(k\right)$, and $\underline{x}'\left[n\left(\bar{k},d'\right)\right]$ the nearest neighbor of $\underline{x}'\left(k\right)$. Then, the index $m\left[\underline{x}\left(t\right),\underline{y}\left(t\right)\right]$ known as mutual false nearest neighbors can be defined by the following equation (Rulkov, et. al., 1995):

$$m\left[\underline{x}\left(k\right),\underline{y}\left(k\right),d,r\right]=\frac{\frac{\left\|\underline{x}'(k)-\underline{x}'\left[n\left(k,d'\right)\right]\right\|}{\left\|\underline{x}'(k)-\underline{x}'[n(k,d)]\right\|}}{\frac{\left\|\underline{y}(k)-\underline{y}[n(k,r)]\right\|}{\left\|\underline{y}(k)-\underline{y}[n(k,d)]\right\|}} \tag{17}$$

Similar to the index $M\left[\underline{X}\left(k\right),\underline{Y}\left(k\right)\right]$ the value of $m\left[\underline{x}\left(k\right),\underline{y}\left(k\right),d,r\right]$ is expected to be close to 1 for all k. However, since the embedded spaces $\underline{y}\left(k\right)$, $\underline{x}\left(k\right)$ and $\underline{x}'\left(k\right)$ are constructed from noisy data, the mean value of $m\left[\underline{x}\left(k\right),\underline{y}\left(k\right),d,r\right]$ calculated from all avaliable data, is used for synchronism detection. In this case, if the mapping $\varphi\left(\underline{X}\right)$ exists, the mean value of $m\left[\underline{x}\left(k\right),\underline{y}\left(k\right),d,r\right]$ is expected to be close to 1. Otherwise, the mean value will be greater than 1.

2.3 CHAOS INPUT SELECTION ALGORITHM

The application of Takens' Theorem and chaotic synchronization for input selection for neural network forecasting models can be summarized as follows:

1. Given a time-series database, define the one to be predicted, $y\left(k\right)\in\mathbb{R}$, $k=1,2,...,N$, and the exogenous time-series, $x_i\left(k\right)\in\mathbb{R}$, $k=1,2,...,N$, $i=1,2,...,S$, where N is the number of points and S the number of avaliable exogenous time-series;
2. Define the maximum dimension parameter $d_{\max}$ and the confidence level α. In this work, $d_{\max}=30$ and $\alpha=0.01$;
3. Estimate the embedding parameters τ_y and d_y using the methods described in section , obtaining the reconstructed space $\underline{y}\left(k\right)\in\mathbb{R}^{d_y}$ via Takens' Theorem, given by equation (3), with $k=\left(d_y-1\right)\tau_y+1,\left(d_y-1\right)\tau_y+2,...,N$;
4. For each exogenous time-series $x_i\left(k\right)\in\mathbb{R}$, do:
   (a) Estimate the embedding parameters τ_{x_i} and d_{x_i} for the reconstructed space $\underline{x}_i\left(k\right)\in\mathbb{R}^{d_{x_i}}$ given by equation (3) with $k=\left(d_{x_i}-1\right)\tau_{x_i}+1,\left(d_{x_i}-1\right)\tau_{x_i}+2,...,N$;
   (b) Detect the existence of synchronism between $\underline{y}\left(k\right)\in\mathbb{R}^{d_y}$ and $\underline{x}_i\left(k\right)\in\mathbb{R}^{d_{x_i}}$ by calculating the mean of $m\left[\underline{x}\left(k\right),\underline{y}\left(k\right),d,r\right]$, i.e., $\overline{m}\left[\underline{x}\left(k\right),\underline{y}\left(k\right)\right]$, required by the mutual false nearest neigbhors method (section)
   (c) If the synchronism does not exist, i.e., $\overline{m}\gg 1$, discard the reconstruction $\underline{x}_i\left(k\right)\in\mathbb{R}^{d_{x_i}}$. Otherwise, include $\underline{x}_i\left(k\right)\in\mathbb{R}^{d_{x_i}}$ in the input set.

5. If another information is avaliable, i.e., qualitative information, binary variables, etc., insert them in the input representation.

Once defined the initial input space representation, a neural network can be applied to model the equation (12), i.e., the function that maps $\underline{Y}\,(k)$ and $\underline{X}\,(k)$ on $Y\,(k+1)$, the first element of vector $\underline{Y}\,(k+1)$.

3 BAYESIAN INFERENCE APPLIED TO MLPS TRAINING AND SPECIFICATION

Let $\underline{x} \in \mathbb{R}^n$ be the vector containing the input signals and $\underline{w} \in \mathbb{R}^M$ the vector with all weights and biases of the MLP with one hidden layer and only one output, being $M = mn + 2m + 1$ with m equal to the number of neurons in the hidden layer. The biases of the sigmoidal functions in the hidden layer are represented by b_k, with b being the bias of the single linear neuron of the ouput layer. The output of this MLP is given by:

$$f(,) = \sum_{k=1}^{m} \left[w_k \varphi \left(a_k \sum_{i=1}^{n} (w_{ik} x_i) + b_k \right) \right] + b \qquad (18)$$

Given a dataset $U = \left\{ \underline{\underline{X}}, \underline{Y} \right\}$ with N input-output pairs, $\underline{\underline{X}} \in \mathbb{R}^N \times \mathbb{R}^n$, $\underline{Y} \in \mathbb{R}^N$, $\underline{Y} = [d_1, d_2, ..., d_N]^t$, $d_j \in \mathbb{R}$ being the desired output, the objective of training a MLP from the Bayesian perspective is the estimation of the vector $\underline{w}$ that maximizes the posterior probability given by:

$$p\left(\underline{w} \middle| \underline{\underline{X}}, \underline{Y} \right) = \frac{p\left(\underline{\underline{X}}, \underline{Y} \middle| \underline{w} \right) p\left(\underline{w} \right)}{p\left(\underline{\underline{X}}, \underline{Y} \right)} \qquad (19)$$

From the definition of joint probability,

$$\begin{aligned} &p\,(A, B) = p\,(A|B)\,p\,(B) \\ &p\left(\underline{\underline{X}}, \underline{Y} \right) = p\left(\underline{Y}, \underline{\underline{X}} \right) = p\left(\underline{Y} | \underline{\underline{X}} \right) p\left(\underline{\underline{X}} \right) \end{aligned} \qquad (20)$$

and

$$p\left(\underline{Y}, \underline{\underline{X}} \middle| \underline{w} \right) = p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right) p\left(\underline{\underline{X}} \middle| \underline{w} \right) = p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right) p\left(\underline{\underline{X}} \right) \qquad (21)$$

since the input patterns $\underline{\underline{X}}$ are independent of the value of $\underline{w}$. Putting these results in equation (19):

$$p\left(\underline{w} \middle| \underline{\underline{X}}, \underline{Y} \right) = \frac{p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right) p\left(\underline{w} \right)}{p\left(\underline{Y} | \underline{\underline{X}} \right)} \qquad (22)$$

In equation (22), $p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right)$ is the likelihood of $\underline{Y}$, $p\left(\underline{w} \right)$ the prior probability of $\underline{w}$ and $p\left(\underline{Y} | \underline{\underline{X}} \right) = \int p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right) p\left(\underline{w} \right) d\underline{w}$ a normalization factor.

The prior probability $p\left(\underline{w} \right)$ represents the prior knowledge of the behavior of $\underline{w}$. Prior insights about specific values for $\underline{w}$ for general problems are unknown, but models with small weights can reproduce smooth mappings (Bishop, 1995). The likelihood $p\left(\underline{Y} \middle| \underline{\underline{X}}, \underline{w} \right)$ represents the knowledge about the distribution of the noise in the desired output. Assuming that $\underline{w}$ follows a Gaussian distribution with null mean vector and diagonal covariance matrix equal to $\alpha^{-1}\underline{\underline{I}}$, where $\underline{\underline{I}}$ is the identity matrix of dimension $M \times M$, and that the desired output is corrupted by aditive gaussian white noise with variance β^{-1}, i.e., $d_j = f(\underline{x}_j, \underline{w}) + \zeta_j$, the application of equation (22) results:

$$p\left(\underline{w} \middle| \underline{\underline{X}}, \underline{Y} \right) = \frac{e^{[-S(\underline{w})]}}{\int e^{-S(\underline{w})} d\underline{w}} \qquad (23)$$

where

$$S(\underline{w}) = \frac{\beta}{2} \sum_{j=1}^{N} \left[d_j - f\left(\underline{x}_j, \underline{w} \right) \right]^2 + \frac{\alpha}{2} \sum_{l=1}^{M} w_l^2 \qquad (24)$$

Therefore, maximize the posterior probability $p\left(\underline{w} \middle| \underline{\underline{X}}, \underline{Y} \right)$ is equivalent to minimize $S(\underline{w})$.

For multivariate problems, the use of a single prior for all weights and biases is not recommended (Ferreira and Alves da Silva, 2007). It is not expected that weights that connect different kinds of inputs have the same distribution in weight space. In (Ferreira and Alves da Silva, 2007), the weights that connect each input to the neurons in hidden layer are grouped, with each group having its own prior distribution $p\left(\underline{w}_i \right)$. All priors are Gaussian with null mean vector and respective diagonal covariance matrix $\alpha_i{}^{-1}\underline{\underline{I}}_i$, with $\underline{\underline{I}}_i$ being the identity matrix of dimension $M_i \times M_i$ where M_i represents the number of weights or bias included in the i-th group. The same idea is applied to the groups of weights associated with the biases (one α_i for the connections with the hidden neurons and another for the output neuron connection). One last α_i is associated with all connection weights between the hidden and output layers. Therefore, for n dimensional input vectors $\underline{x}$, the total number of α_is is n+3. In this case, $S(\underline{w})$ is given by:

$$S(\underline{w}) = \frac{\beta}{2} \sum_{j=1}^{N} \left[d_j - f\left(\underline{x}_j, \underline{w} \right) \right]^2 + \frac{1}{2} \sum_{i=1}^{n+3} \alpha_i \sum_{l=1}^{M} w_{il}^2 \qquad (25)$$

Details about the iterative algorithm for minimization of $S(\underline{w})$ and estimation of parameters and hyperparameters α_i's and β can be find in (Mackay, 1992) and (Bishop, 1995).

The magnitude of α_i's related to the inputs connections can be used for ranking the relevance of each input signal in the calculation of the output. This characteristic makes this specification of the priors be known as Automatic Relevance Determination (ARD) (Mackay, 1992). Besides ranking capacity, irrelevance levels must be specified to determine the irrelevant inputs that should be discarded by the model. Since this irrelevant threshold is problem dependent, (Ferreira and Alves da Silva, 2007) proposed an empiric method for automatic determination of the referred threshold. Artificial probe signals, unrelated with the desired output and generated from uniform distributions, are included in the original input space. After training the MLP with this augmented input space, the α_i related to the probe signals is used as irrelevance threshold. The relevant inputs are selected and the final model is trained. For continuous variables, the probe signal are generated from a uniform distribution defined in the same scale as the original normalized inputs. For dummy variables, a discrete uniform distribution is used. Since this technique uses the model along the input selection step, it can be included in the group of wrapper methods (Guyon and Elisseeff, 2003). More details can be find in (Ferreira and Alves da Silva, 2007).

The Bayesian inference can also be applied to the selection of the most probable MLP structure to represent a given mapping among a set of hypothesis $\mathrm{H} = \{H_1, H_2, ..., H_K\}$. The set of relevant inputs for each hypothesis was previously defined by ARD with probe signals, and the difference between hypothesis is the number of neurons in the hidden layer. Assuming that all hypothesis are equiprobable and using a Gaussian aproximation around the parameters and hyperparameters previously estimated, the logarithm of the evidence for the models, $\ln p\left(\underline{Y}\middle| H_h\right)$, can be obtained by (Bishop, 1995):

$$\ln p\left(\underline{Y}\middle| H_h\right) = -S\left(\underline{w}\right) - \tfrac{1}{2}\ln\left|\underline{\underline{A}}\left(\underline{w}\right)\right| + \tfrac{1}{2}\sum_{i=1}^{n+3} M_i\alpha_i \\ + \ln(\beta^{\frac{N}{2}} m^2 m!) + \tfrac{1}{2}\sum_{i=1}^{n+3}\ln\left(\tfrac{2}{\gamma_i}\right) + \tfrac{1}{2}\ln\left(\tfrac{2}{N-\gamma}\right) \tag{26}$$

4 RELEVANCE VECTOR MACHINES

Relevance Vector Machines (RVMs) (Tipping, 2001) are kernel-based sparse probabilistic models. Sparse in the sense that only some vectors of the training set contribute for the estimation of the regression surface. These points are called relevant vectors.

Given a dataset $U = \left\{\underline{\underline{X}}, \underline{Y}\right\}$ including the input-output pairs, let's assume the traditional probabilistic formulation considering an additive noise $\zeta_k \in$ present in the desired output, i.e., $d_k = F\left(\underline{x}_k\right) + \zeta_k$. In order to model $F\left(\underline{x}\right) : \mathbb{R}^n \to \mathbb{R}$, let $f\left(\underline{x}, \underline{w}\right) : \mathbb{R}^n \to \mathbb{R}$ be a function formed by the linear combination of functions $\Phi\left(\underline{x}, \underline{z}\right) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ centered at each point of the dataset D:

$$f\left(\underline{x}, \underline{W}\right) = \sum_{i=1}^{N} w_i \Phi\left(\underline{x}, \underline{x}_i\right) + b = \left[\underline{\Phi}\left(\underline{x}\right)\right]^t \underline{W} \tag{27}$$

In equation (27), $\underline{w} \in \mathbb{R}^N$, $b \in \mathbb{R}$, $\underline{W} \in \mathbb{R}^{N+1}$, $\underline{W} = \begin{bmatrix} \underline{w}^t & b \end{bmatrix}^t$, with $\underline{\Phi}\left(\underline{x}\right) : \mathbb{R}^n \to \mathbb{R}^{N+1}$, a matrix including the functions $\Phi\left(\underline{x}, \underline{x}_i\right) = \Phi_i\left(\underline{x}\right)$ and a constant term equal to one representing the bias.

Using Bayes' rule, the posterior probability $p\left(\underline{W}\middle| \underline{\underline{X}}, \underline{Y}\right)$ is given by:

$$p\left(\underline{W}\middle| \underline{\underline{X}}, \underline{Y}\right) = \frac{p\left(\underline{Y}\middle| \underline{\underline{X}}, \underline{W}\right) p\left(\underline{W}\right)}{p\left(\underline{Y}\middle|\underline{\underline{X}}\right)} \tag{28}$$

As in equation (19), $p\left(\underline{Y}\middle|\underline{\underline{X}}\right) = \int p\left(\underline{Y}\middle| \underline{\underline{X}}, \underline{w}\right) p\left(\underline{w}\right) d\underline{w}$ is a normalization factor, $p\left(\underline{W}\right)$ the prior probability of $\underline{W}$ and $p\left(\underline{Y}\middle| \underline{\underline{X}}, \underline{W}\right)$ is the likelihood function related to the distribuition of the additive noise ζ_k presented in the desired output.

Assuming that the samples of ζ_k are generated independently from the same Gaussian distribution with zero mean and variance $\sigma^2 \in \mathbb{R}$, the likelihood function $p\left(\underline{Y}\middle| \underline{\underline{X}}, \underline{W}\right)$ is given by:

$$p\left(\underline{Y}\middle| \underline{\underline{X}}, \underline{W}, \sigma^2\right) = \frac{1}{\left(2\pi\sigma^2\right)^{\frac{N}{2}}} \exp\left(-\frac{\left\|\underline{Y} - \underline{\underline{\Phi}}\underline{W}\right\|^2}{2\sigma^2}\right) \tag{29}$$

where $\underline{\underline{\Phi}} \in \mathbb{R}^N \times \mathbb{R}^{N+1}$ is the modeling matrix including all the functions $\Phi_i\left(\underline{x}\right)$ evaluated at each point of the training set, i.e., the *ij*-th element is $\underline{\underline{\Phi}}_{ij} = \Phi_j\left(\underline{x}_i\right)$ and $\underline{\underline{\Phi}}_{i(N+1)} = 1$.

The prior probability $p\left(\underline{W}\right)$ can be defined as a product of Gaussian distributions given by:

$$p\left(\underline{W}\middle| \underline{\alpha}\right) = \prod_{i=1}^{N+1} \frac{1}{\sqrt{2\pi{\alpha_i}^{-1}}} \exp\left(-\frac{1}{2{\alpha_i}^{-1}} W_i^2\right) \tag{30}$$

In equation (30) distinct Gaussian distributions are considered, all of them with zero mean but different variances. These hyperparameters are responsible for magnitude control of each parameter W_i. As in ARD, weights with large α_i will tend to be highly centered around the null vector. The estimation of α_i and identification of weights with sufficient large α_i can be used to select the functions $\Phi_i\left(\underline{x}\right)$ that will be included in the final model. This feature enables RVMs to present a sparse representation such as in other kernel methods like Support Vector Machines (SVMs).

The definition of hyperparameters σ^2 and $\underline{\alpha}$ requires the specification of prior probabilities for them. Non-informative gamma distributions are used, reflecting the prior absence of knowledge about hyperparameters' distributions (Tipping, 2001).

Using the prior and the likelihood distribuitions defined by equations (29) and (30), respectively, in equation (28), and making a convolution of Gaussians to calculate the normalization factor $p\left(\underline{Y}|\underline{\underline{X}}\right) = \int p\left(\underline{Y}|\,\underline{\underline{X}},\underline{w}\right)p\left(\underline{w}\right)d\underline{w}$, the posterior probability $p\left(\underline{W}|\,\underline{\underline{X}},\underline{Y},\underline{\alpha},\sigma^2\right)$ can be written as:

$$p\left(\underline{W}|\,\underline{\underline{X}},\underline{Y},\underline{\alpha},\sigma^2\right) = \frac{\exp\left[-\frac{1}{2}\left(\underline{W}-\underline{\mu}\right)^t\underline{\underline{\Sigma}}^{-1}\left(\underline{W}-\underline{\mu}\right)\right]}{(2\pi)^{\frac{N+1}{2}}\left|\underline{\underline{\Sigma}}\right|^{\frac{1}{2}}} \tag{31}$$

where $\underline{\underline{\Sigma}} \in \mathbb{R}^{N+1} \times \mathbb{R}^{N+1}$ and $\underline{\mu} \in \mathbb{R}^{N+1}$ are given by

$$\begin{aligned}\underline{\underline{\Sigma}} &= \left(\sigma^2\underline{\underline{\Phi}}^t\underline{\underline{\Phi}}+\underline{\underline{A}}\right)^{-1}\\ \underline{\mu} &= \sigma^{-2}\underline{\underline{\Sigma\Phi}}^t\underline{Y}\end{aligned} \tag{32}$$

with $\underline{\underline{A}} \in \mathbb{R}^{N+1}$ being a diagonal matrix with the *ii*-th element $a_{ii} = \alpha_i$. The expected value of the desired output $\widehat{d}_{N+1}$ and the estimate of the corresponding variance $\widehat{\sigma}^2$ associated with a testing point $\underline{x}_{N+1}$ are obtained through the expressions:

$$\begin{aligned}\widehat{d}_{N+1} &= f\left(\underline{x}_{N+1},\underline{\mu}^{MP}\right) = \left[\underline{\Phi}\left(\underline{x}_{N+1}\right)\right]^t\underline{\mu}^{MP}\\ \widehat{\sigma}^2 &= \left(\sigma^{MP}\right)^2+\left[\underline{\Phi}\left(\underline{x}_{N+1}\right)\right]^t\underline{\underline{\Sigma}}^{MP}\underline{\Phi}\left(\underline{x}_{N+1}\right)\end{aligned} \tag{33}$$

In equation (33), $\underline{\mu}^{MP}$ and $\underline{\underline{\Sigma}}^{MP}$ are calculated by equations (32) using estimated $\underline{\alpha}^{MP}$ and σ^{MP}. An iterative method for calculating hyperparameters $\underline{\alpha}^{MP}$ and σ^{MP}, based on evidence maximization, analogous to Mackays's evidence maximization for MLPs, can be found in (Tipping, 2001).

Unlike other sparse kernel-based models whose basis functions must agree with Mercer's Theorem conditions (Vapnik, 1998), (Schölkopf and Smola, 2002), the function $\Phi\left(\underline{x},\underline{z}\right)$ used in RVMs does not need to meet the Mercers' conditions. In this work a Gaussian function is used:

$$\Phi\left(\underline{x},\underline{z}\right) = e^{-\sum\limits_{k=1}^{n}\eta_k\left(x_k-z_k\right)^2} \tag{34}$$

In equation (34), $\eta_k \in \mathbb{R}^+$ denotes another set of hyperparameters that are iteratively estimated by evidence maximization (Tipping, 2001). This choice for $\Phi\left(\underline{x},\underline{z}\right)$ allows the creation of a input selection method analogous to ARD (presented in section). After the estimation of η_k's, inputs with smallest η_k contribute less for output calculation. In other words, the magnitude of η_k can be used for ranking the input variables. Similarly to the input selection method used for MLPs and presented in section , artificial probe signals are included in the original input space to define irrelevant thresholds for the inputs. After training with augmented input space including the probe signals, the relevant inputs are selected for re-training the model and making predictions.

5 AUTONOMOUS MODELING

Chaos Theory, BIAMLP and RVM are combined in this paper in order to develop an analytic, coupled and unified framework for autonomous forecasting using neural models. This framework includes input space representation selection, structure definition and complexity control of the forecasting model, all of them disregarding the necessity of a validation set. The use of validation sets brings some practical and theoretical problems as described in (Amari et. al., 1996) and (Cataltepe et. al., 1999). A pratical disadvantage of cross-validation, specially in time series applications, is related to the definition of the validation set, since serial correlations or recent information can be neglected in the training phase. Using all the available data for model development, the framework proposed here can be summarized as follows:

1. Apply the Chaos Input Selection algorithm (section 2.3) to define the initial input representation space;
2. Use BIAMLP or RVM to model the mapping between input-output pairs;
3. Discard the irrelevant inputs using the wrapper methods described in sections 3 (BIAMLP) and 4 (RVMs);
4. Make predictions by recursion for all the forecasting horizon.

The autonomous modeling proposed can be summarized by the flowchart in Figure 1.

6 RESULTS

The models presented in previous section are evaluated through the application of them to three public databases. The first database presents hourly load L(k), temperature T(k) and temperature squared T2(k) for the period of January 1, 1985 to March 31, 1991. This database was used in a forecasting competition (Ramanathan et. al., 1997) and can be found in the web at the adress http://www.ee.washington.edu/class/555/el-sharkawi/datafiles/forecasting.zip. For this database, hourly load must be predicted from 16 to 49 steps ahead for weekdays and from 16 to 80 steps ahead for weekend. The forecasts are made daily by 9 a.m., with the testing period starting

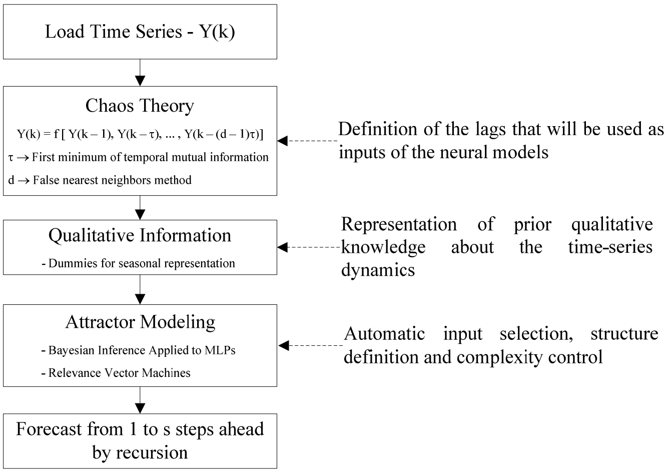
Load Time Series - Y(k)
Chaos Theory
Y(k) = f [Y(k – 1), Y(k – τ), ... , Y(k – (d – 1)τ)]
τ → First minimum of temporal mutual information
d → False nearest neighbors method
Qualitative Information
- Dummies for seasonal representation
Attractor Modeling
- Bayesian Inference Applied to MLPs
- Relevance Vector Machines
Forecast from 1 to s steps ahead by recursion
Definition of the lags that will be used as inputs of the neural models
Representation of prior qualitative knowledge about the time-series dynamics
Automatic input selection, structure definition and complexity control

Figure 1: Proposed autonomous neural modeling

in November 1, 1990 and finishing in March 31, 1991. For definition of the lags by Chaos Theory, the data from January 1, 1985 to October 31, 1990 (database avaliable at the begining of the forecasting period) are used. After the definition of the lags that will be used as inputs to the model, the input-output pairs used for training the models corresponds to the data from the current month, the two previous months and the corresponding pairs from the same period in the last year. This subset of the training data is used for training in order to reduce the computational effort for training. Some statistics for this database are shown in Table 1 to Table 3, where mean, standard deviation and the relation between then are presented, respectively. This statistics are calculated after detrending of the load time-series using a time linear regression model.

Table 1: Mean value (μ) of load [MW] for each hour of each day of the week for Case 1

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	1254.40	1194.90	1242.38	1255.43	1262.99	1259.69	1280.40
2	1172.75	1136.45	1177.09	1190.06	1195.96	1190.38	1197.90
3	1138.39	1124.41	1156.26	1169.07	1173.96	1166.53	1162.60
4	1130.82	1138.98	1165.46	1178.35	1181.34	1172.65	1155.62
5	1148.91	1213.65	1238.47	1251.98	1252.47	1241.58	1182.57
6	1205.35	1460.67	1497.31	1511.86	1504.08	1483.24	1261.84
7	1313.81	1892.48	1953.92	1970.10	1950.84	1916.07	1410.75
8	1531.31	2159.90	2217.47	2234.28	2214.76	2189.88	1660.89
9	1800.35	2192.98	2212.18	2225.31	2216.88	2214.18	1937.54
10	1952.27	2158.99	2140.05	2151.04	2152.10	2162.16	2079.72
11	1976.98	2118.41	2080.18	2088.92	2091.45	2102.67	2081.33
12	1920.77	2042.51	2001.25	2007.63	2010.15	2016.17	1997.16
13	1851.71	1971.86	1939.29	1942.58	1943.42	1943.43	1889.30
14	1769.19	1912.90	1888.46	1891.47	1889.06	1888.22	1782.91
15	1690.36	1862.64	1844.51	1847.70	1842.91	1842.25	1707.22
16	1664.80	1861.22	1850.60	1851.33	1845.27	1839.48	1683.62
17	1730.42	1935.49	1933.11	1930.66	1918.38	1898.34	1747.17
18	1842.83	2074.13	2078.28	2070.02	2051.19	2006.85	1854.13
19	1888.74	2125.05	2129.52	2120.68	2102.06	2038.33	1887.80
20	1891.54	2098.97	2100.12	2092.60	2077.26	1986.40	1851.72
21	1870.63	2044.92	2048.90	2045.27	2026.04	1920.64	1798.84
22	1787.53	1912.22	1921.44	1928.89	1915.55	1818.97	1715.01
23	1578.95	1674.75	1691.71	1699.38	1692.82	1660.78	1582.42
24	1353.88	1418.81	1431.23	1439.66	1438.93	1459.70	1408.17

Table 2: Standard deviation (σ) of load [MW] for each hour of each day of the week for Case 1

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	305.39	275.4	270.65	269.67	286.32	286.93	301.43
2	307.17	281.61	276.8	277.51	293.93	295.24	307.69
3	314.17	290.86	286.37	290.24	305.08	307.41	317.07
4	324.38	304.65	299.98	306.43	319.54	322.61	328.93
5	338.45	328.98	324.48	333.02	343.11	346.66	343.9
6	365.24	393.24	387.68	397.18	402.44	402.56	372.86
7	402.43	505.96	489.21	499.72	499.13	494.39	413.99
8	449.22	549.8	526.23	538.94	537.88	531.65	457.08
9	486.47	498.32	476.88	491.28	495.52	489.85	488.76
10	483.16	435.79	412.54	430.33	440.5	433.36	479.88
11	454.77	391.01	368.16	387.14	398.16	387.5	447.8
12	426.72	358.26	335.97	356.11	367.55	354.02	412.04
13	410.68	336.63	316.45	335.3	341.57	329.37	381.23
14	397.3	319.71	299.8	317.26	318.96	311.62	357.91
15	385.47	312.7	292.6	312.67	308.14	306.21	347.51
16	387.96	329.41	311.96	329.34	325	324.68	359.28
17	434.3	390.07	378.21	392.74	385.46	384.67	416.4
18	491.93	470.73	466.62	478.29	468.13	461.22	482.08
19	492.26	488.21	485.85	497.1	490.37	476.66	484.54
20	459.13	460.68	460.66	473.98	468.11	450.43	449.84
21	404.15	398.89	400.57	418.37	414.21	406.38	402.81
22	334.54	326.42	328.31	346.83	344.84	354.07	350.5
23	296.4	288.95	288.26	305.52	305.35	324.95	326.63
24	277.71	269.8	270.93	287.36	288.28	306.65	311.43

Table 3: Relation between mean and standard deviation (μ/σ) for load for each hour of each day of the week for Case 1

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	4.11	4.34	4.59	4.66	4.41	4.39	4.25
2	3.82	4.04	4.25	4.29	4.07	4.03	3.89
3	3.62	3.87	4.04	4.03	3.85	3.79	3.67
4	3.49	3.74	3.89	3.85	3.70	3.63	3.51
5	3.39	3.69	3.82	3.76	3.65	3.58	3.44
6	3.30	3.71	3.86	3.81	3.74	3.68	3.38
7	3.26	3.74	3.99	3.94	3.91	3.88	3.41
8	3.41	3.93	4.21	4.15	4.12	4.12	3.63
9	3.70	4.40	4.64	4.53	4.47	4.52	3.96
10	4.04	4.95	5.19	5.00	4.89	4.99	4.33
11	4.35	5.42	5.65	5.40	5.25	5.43	4.65
12	4.50	5.70	5.96	5.64	5.47	5.70	4.85
13	4.51	5.86	6.13	5.79	5.69	5.90	4.96
14	4.45	5.98	6.30	5.96	5.92	6.06	4.98
15	4.39	5.96	6.30	5.91	5.98	6.02	4.91
16	4.29	5.65	5.93	5.62	5.68	5.67	4.69
17	3.98	4.96	5.11	4.92	4.98	4.94	4.20
18	3.75	4.41	4.45	4.33	4.38	4.35	3.85
19	3.84	4.35	4.38	4.27	4.29	4.28	3.90
20	4.12	4.56	4.56	4.41	4.44	4.41	4.12
21	4.63	5.13	5.12	4.89	4.89	4.73	4.47
22	5.34	5.86	5.85	5.56	5.55	5.14	4.89
23	5.33	5.80	5.87	5.56	5.54	5.11	4.84
24	4.88	5.26	5.28	5.01	4.99	4.76	4.52

The second database includes daily load $L(k)$ and daily maximum temperature $T(k)$ from the period of January 1, 1997 to January 31, 1999. As the first database, this one was also used in a forecasting competition (Chen et. al., 2004), where the objective was the daily prediction of the load from January 1, 1999 to January 31, 1999, i.e., forecasts for 1 to 31 steps ahead. The data from January 1, 1997 to December 31, 1998 was used for input space definition and training of the models. This database can be found at the website http://neuron.tuke.sk/competition. As for case 1, some statistics for this database are presented in Table 4. These statistics are estimated after detrending the load time-series using a time linear regression model.

The third database shows half-hourly load $L(k)$ and temperature $T(k)$ for the period of December 4, 2001 to December 31, 2003. The hourly databases are obtained by the mean value between two registers in the hour (Mandal et. al., 2005). The objective is to forecast hourly load from one to six hours ahead along the period from September 1, 2003 to September 7, 2003. The data from December 4, 2001 to August 31, 2003 are used for initial input space definition. The same subset selected for training the models in Case 1, i.e., data from the current month, the two previous months and respective pairs from the same period in the last year, are used for development of the models in this case. This database is related to Victoria State and can be found in web at the address at http://www.aemo.com.au/data/aggPD_2000to2005.html. As for case 1, Table 5 to Table 7 presented some statistics for this load time-series, all of them calculated after detrending the load time-series using a time linear regression model.

Table 4: . Mean (μ) [MW], standard deviation (σ) [MW] and relation between them (μ/σ) of load for each day of the week for Case 2

	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
Mean (μ)	612.49	682.96	688.31	690.95	684.62	676.84	650.26
Standard Deviation (σ)	84.45	92.35	92.12	89.77	91.37	92.06	88.22
Relation (μ/σ)	7.25	7.40	7.47	7.70	7.49	7.35	7.37

Table 5: Mean value (μ) of load [MW] for each day of the week for Case 2

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	4911.9	4808.13	5079.67	5117.9	5119.24	5118.91	5119.94
2	4875.96	4801.66	5044.94	5078.14	5085.4	5095.53	5060.07
3	4782.06	4763.93	4960.62	4984.03	4995.34	5004.68	4947.98
4	4401.63	4431.1	4589.7	4612.97	4618.5	4628.13	4551.57
5	4152.15	4262.61	4402.02	4429.85	4431.78	4436.53	4317.19
6	4037	4354.44	4496.01	4517.32	4515.5	4517	4262.43
7	4066.58	4826.33	4984.72	4999.78	4985.39	4972.71	4399.26
8	4158.34	5375.88	5554.25	5562.46	5533.91	5509.73	4611.46
9	4325.1	5647.51	5817.56	5829.91	5800.44	5777.66	4839.33
10	4567.42	5772.07	5916.5	5934.1	5882.83	5867.16	5076.24
11	4702.58	5810.21	5941.58	5953.49	5884.2	5873.78	5121.54
12	4746.29	5822.27	5944.28	5955.7	5875.2	5849	5080.81
13	4757.63	5830.65	5944	5942.34	5865.5	5815.27	5019.88
14	4750.63	5853.5	5972.92	5948.61	5874.02	5804.81	4971.58
15	4725.52	5865.96	5978.84	5951.32	5880.68	5784.28	4917.71
16	4715.45	5835.81	5956.8	5908.81	5833.62	5716.18	4865.32
17	4774.23	5844.06	5956.62	5898.37	5831.1	5678.23	4870.75
18	4942.52	5883.58	5968.46	5913.43	5857.27	5695.59	5011.29
19	5186.34	5897.21	5962.75	5912.3	5894.92	5734.42	5225.79
20	5216.59	5749.24	5827.19	5780.46	5809.84	5644.09	5211.11
21	5175.69	5613.51	5689.01	5651.33	5693.73	5516.1	5110.84
22	5008.82	5374.09	5444.2	5425.09	5449.72	5285.24	4956.48
23	4747.14	5039.01	5109.31	5094.26	5112.47	5018.1	4783.94
24	4780.49	5044.81	5105.24	5100.88	5117.59	5110.11	4875.47

The hourly and daily load data used in this paper present seasonal patterns widely known in load forecasting area namely: hour, week and yearly seasonal pattern (Ferreira and Alves da Silva, 2007), (Hippert et. al., 2001). The yearly pattern is related to the seasons and is modelled by the temperature information. The other patterns are modelled as qualitative information, being represented as binary variables indicating the hour of the day (24 dummies) and day of the week (7 dummies) to be forecasted. The daily database (Case 2) uses only the dummies for day of the week.

Table 6: Standard deviation (σ) of load [MW] for each hour of each day of the week for Case 1

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	298.21	265.38	285.12	304.75	296.86	332.77	353.88
2	321.37	280.46	305.33	317.8	315.69	346.73	352.28
3	328.27	294.6	333.05	334.1	336.06	357.67	356.58
4	327.06	298.08	329.37	334.21	331.39	350.88	359.41
5	288.19	263.72	287.78	300.56	293.63	308.97	300.42
6	246.39	272.2	295.97	320.02	307.75	310.62	261.5
7	221.44	413.35	419.64	458.5	449.85	441.84	260.79
8	184.96	532.26	462.34	525.07	528.09	547.31	262.98
9	184.18	582.7	471.58	554.7	555.64	602.49	267.04
10	220.2	568.89	465.86	543.64	533.1	583.57	296.25
11	235.45	540.92	460.99	535.34	507.53	569.42	290.66
12	232.38	534.12	477.42	534.79	492.88	559.1	266.11
13	242.73	537	514.83	549.57	501.61	557.75	254.9
14	265.97	570.51	576.92	580.26	533.8	571.24	266.55
15	290.38	604.34	619.3	586.49	551.27	590.2	282.68
16	298.58	628.04	654	589.7	550.41	580.07	284.89
17	310.48	630.72	668.58	585.52	544.94	560.22	290.25
18	345.18	595.76	636.5	552.65	546.37	540.66	336.33
19	498.39	682.53	703.78	650.98	658.23	623.66	506.14
20	493.38	642.28	680.51	636.51	640.91	596.59	499.64
21	397.03	531.36	576.81	522.05	529.82	489.03	374.43
22	354.38	473.13	497.89	459.82	472.77	436.59	321.06
23	320.84	415.75	441.88	417.22	442.36	408.4	310.63
24	273.5	328.32	349.47	339.6	369.95	383.15	291.34

Table 7: Relation between mean (μ) and standard deviation (σ) for load for each hour of each day of the week for Case 1

Hour	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
1	16.47	18.12	17.82	16.79	17.24	15.38	14.47
2	15.17	17.12	16.52	15.98	16.11	14.7	14.36
3	14.57	16.17	14.89	14.92	14.86	13.99	13.88
4	13.46	14.87	13.93	13.8	13.94	13.19	12.66
5	14.41	16.16	15.3	14.74	15.09	14.36	14.37
6	16.38	16	15.19	14.12	14.67	14.54	16.3
7	18.36	11.68	11.88	10.9	11.08	11.25	16.87
8	22.48	10.1	12.01	10.59	10.48	10.07	17.54
9	23.48	9.69	12.34	10.51	10.44	9.59	18.12
10	20.74	10.15	12.7	10.92	11.04	10.05	17.13
11	19.97	10.74	12.89	11.12	11.59	10.32	17.62
12	20.42	10.9	12.45	11.14	11.92	10.46	19.09
13	19.6	10.86	11.55	10.81	11.69	10.43	19.69
14	17.86	10.26	10.35	10.25	11	10.16	18.65
15	16.27	9.71	9.65	10.15	10.67	9.8	17.4
16	15.79	9.29	9.11	10.02	10.6	9.85	17.08
17	15.38	9.27	8.91	10.07	10.7	10.14	16.78
18	14.32	9.88	9.38	10.7	10.72	10.53	14.9
19	10.41	8.64	8.47	9.08	8.96	9.19	10.32
20	10.57	8.95	8.56	9.08	9.06	9.46	10.43
21	13.04	10.56	9.86	10.83	10.75	11.28	13.65
22	14.13	11.36	10.93	11.8	11.53	12.11	15.44
23	14.8	12.12	11.56	12.21	11.56	12.29	15.4
24	17.48	15.37	14.61	15.02	13.83	13.34	16.73

Table 8 shows the estimated embedding parameters τ and d via the first minimum of the mutual information $I_x(r)$ and false nearest neighbors method, respectively, and the mean value of the mutual false nearest neighbors statistic $\overline{m}\left[\underline{x}(k), \underline{y}(k)\right]$. As expected, the value of $\overline{m}\left[\underline{x}(k), \underline{y}(k)\right]$ confirms that there is synchronism between the reconstructed load and temperature for all cases. The results presented in Table 8 for case 1 confirms that the calculated embedded parameters are invariant characteristics of the attractor, since the values calculated for $T(k)$ and $T^2(k)$ are very close.

Table 9 shows the mean absolute percentage error (MAPE) for the three databases studied in this paper. For case 3, the results are discretized by step ahead to compare with the benchmark of the literature (Mandal, et. al., 2005). The last line of this Table presents the best results found in the literature for these databases. The analysis of Table 9 shows that the autonomous models proposed in this paper, specially the BIAMLP, are competitive against the best results in the literature. It is noteworthy that the benchmark results presented in Table 2 was obtained by highly specialized and dedicated group of modeling experts, while the autonomous models proposed here are mainly automatic, requiring little manual intervention. Further, among the usual parameters that must be specified in neural network training, the analyst must specifies only the maximum dimension parameter $d_{\max}$, the confidence level α (Chaos Input Selection Algorithm) and the maximum number of neurons in the hidden layer to be tested in BIAMLP. Compared with the effort to define heuristically the input space representation, together with the necessity of criteria to neural network structure definition, the autonomous modeling proposed here shows a considerable level of automation.

Table 8: Embedding parameters ((τ, d) and mutual false nearest neighbors statistic $\overline{m}\left[\cdot,\cdot\right]$

	Case 1		Case 2		Case 3	
	D	τ	d	τ	d	τ
$L(k)$	10	6	12	4	12	13
$T(k)$	18	13	14	15	19	13
$T^2(k)$	17	13	-	-	-	-
$\overline{m}\left[\underline{l}\left(k\right),\underline{t}\left(k\right)\right]$	2.96		1.72		1.95	
$\overline{m}\left[\underline{l}\left(k\right),\underline{t^2}\left(k\right)\right]$	2.97		-		-	

Table 9: MAPE for forecasting period

	Case 1	Case 2	Case 3 (steps ahead)					
			1	2	3	4	5	6
BIAMLP	4.83	3.25	0.64	1.02	1.55	1.69	1.87	1.88
RVM	8.64	3.00	1.09	1.80	2.10	2.29	2.72	2.94
Benchmark	4.73	1.98	0.56	0.83	1.00	1.15	1.20	1.30

Despite the distinct statistical features of the databases under consideration, as presented in Tables 1 to 7, which show significant differences both in level (mean) and variability (standard deviation), the performance of the proposed autonomous models, particularly from BIAMLP, was always robust. Even for Case 2, for which BIAMLP presents the worst result when compared with the benchmark, BIAMLP would be ranked in the top five competitors (Chen et. al., 2004). For Case 1 data, BIAMLP's results are statistically equivalent to the ones obtained by the winner of the competition. Statistical equivalence between the results from BIAMLP and the corresponding benchmark is also confirmed for Case 3. These findings highlight the robustness of the BIAMLP's performance with respect to different time-series.

The importance of qualitative information is demonstrated in Table 10 with the results obtained by BIAMLP when the dummy variables are discarded from the initial input set. In this case, the inputs are all selected from the time-series data, discarding the prior knowledge about the dynamics of the data. The consistent increase on MAPE for the forecasting period confirms the importance of qualitative information, and shows that Chaos Theory itself does not deal with seasonality modeling. Since the calendar information are available for time-series data, the use of this qualitative information does not reduce the level of automation of the proposed models. In fact, qualitative information (general electric load seasonal characteristics) is included based on the premise that calendar information (time, day of the week, and corresponding month) is available. Therefore, for electric load time series, such qualitative information is automatically inserted as dummy variables ("1 of n" binary coding). Besides, as the developed neural networks can disregard irrelevant information, such inputs are automatically excluded from the forecasting model when a general seasonal characteristic is not present in a particular dataset.

Table 10: MAPE for forecasting period using Chaos Input Selection Algorithm without qualitative information

	Case 1	Case 2	Case 3 (steps ahead)					
			1	2	3	4	5	6
BIAMLP	11.62	4.37	1.20	1.70	1.88	2.41	2.28	2.66
Benchmark	4.73	1.98	0.56	0.83	1.00	1.15	1.20	1.30

7 CONCLUSION

This work investigates the application of Chaos Theory as a input space representation tool in the development of autonomous neural network load forecasting models. Autonomy should be understood here as a set of automatic and coupled procedures for input space definition and selection, structure specification and complexity control (regularization). In this work, two neural network-based models are used: the Bayesian Inference Applied to MLPs training and specification (BIAMLPs) and Relevance Vector Machines (RVMs). The obtained results, comparable with the benchmarks available in the literature, specially for the BIAMLPs, show the potential of the proposal. The automation level of the techniques proposed in (Ferreira and Alves da Silva, 2007) has been increased, enabling the application of the new models to problems envolving multiple time series, as for example, bus load forecasting. Bus load forecasting is needed for feeding important power system control center functions, such as state estimation, generation scheduling, and security assessment.

In terms of computational effort, BIAMLP requires about 10 minutes to estimate the model and to provide the one to eighty hours ahead load forecasts for Case 1 (with Matlab® in a PC Intel® Core$^{\text{TM}}$ 2 Duo 2,66 GHz, 3323 MB RAM Memory, running Windows Vista 32 Bits). Therefore, BIAMLP's computational effort is suitable for short-term load forecasting.

The result for RVMs, although competitive, can be improved by selecting more appropriate basis functions $\Phi_i(\underline{x})$. One interesting theoretical feature of RVMs is the possibility of using different basis functions, such as periodical functions, in order to model seasonal patterns without the use of dummy variables. The development of BIAMLPs considering non-Gaussian noise in the output is another interesting research area, by means of Monte Carlo methods for BIAMLPs definition (Neal, 1996). Beyond those issues, the local modeling of the attractor against the global one used here can still improve the results. In order to automate the identification of the regions to be independently modeled, automatic clustering methods are required.

ACKNOWLEDGEMENTS

Special thanks to the support of National Council for Scientific and Technological Development (CNPq) that funds the author Alexandre Pinto Alves da Silva and to FAPERJ – Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro – that funds author Vitor Hugo Ferreira under Grant INST E-26/110.158/2010.

REFERENCES

Abarbanel, H.D.I., Brown, R., Sidorowich, J.J., Tsimring, L.S. (1993) The Analysis of Observed Chaotic Data in Physical Systems, *Reviews of Modern Physics*, v.65, n.4, pp. 1331-1392.

Amari, S., Murata, N., Müller, K.R., Finke, M., Yang, H. (1996). "Statistical Theory of Overtraining – Is Cross-Validation Asymptotically Effective?", *Advances in Neural Information Processing Systems 8*, MIT Press, pp. 176-182.

Bishop, C.M. (1995). Neural Networks for Pattern Recognition, Oxford University Press.

Cao, L. (1997). Practical Method for Determining the Minimum Embedding Dimension of a Scalar Time Series, *Physica D*, v.110, n.1-2, pp. 43-50.

Cataltepe, Z., Abu-Mostafa, Y.S., Magdon-Ismail, M. (1999). No Free Lunch for Early Stopping", *Neural Computation*, v.11, n.4, pp. 995-1009, May 1999.

Chen, B.-J., Chang, M.-W., Lin, C.-J. (2004). Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001, *IEEE Trans. on Power Systems*, **19**(4), pp. 1821-1830.

Ferreira, V.H., Alves da Silva, A.P. (2007). Toward Estimating Autonomous Neural Network-based Electric Load Forecasters, *IEEE Transactions on Power Systems*, **22** (4), n.4, pp. 1554-1562.

Ferreira, V.H., Alves da Silva, A.P. (2009). Automatic Kernel Based Models for Short Term Load Forecasting, *Proceedings of the 15th International Conference on Intelligent System Application to Power Systems*, Curitiba, Paraná, Brazil.

Ferreira, V.H., Alves da Silva, A.P. (2010). Teoria do Caos Aplicada à Definição do Conjunto de Entradas de Modelos Neurais Autônomos para Previsão de Carga em Curto Prazo. *Anais do XVIII Congresso Brasileiro de Automática (XVIII CBA)*, Bonito-MS, pp.4439-4444.

Fraser, A.M., Swinney, H.L. (1986). Independent Coordinates for Strange Attractors from Mutual Information, *Physical Review A*, v.33, n.2, pp. 1134-1140.

Griffiths, W.E., Hill, R.C., Judge, G.G. (1993). Learning and Practicing Econometrics, John Wiley & Sons.

Guyon, I., Elisseeff, A. (2003). An Introduction to Variable and Feature Selection, *Journal of Machine Learning Research*, n.3, pp. 1157-1182.

Hippert, H.S., Souza, R.C., and Pedreira, C.E. (2001). Neural Networks for Load Forecasting: A Review and Evaluation, *IEEE Transactions on Power Systems*, v.16, n.1, pp. 44-55.

Kantz, H., Schreiber, T. (1997). Nonlinear Time Series Analysis, Cambridge Nonlinear Science Series, n.7, Cambridge University Press.

Kennel, M.B., Brown, R., Abarbanel, H.D.I. (1992). Determining Embedding Dimension for Phase-space Reconstruction Using a Geometrical Construction, *Physical Review A*, v.45, n.6, pp. 3403-3411.

Mackay, D.J.C. (1992). Bayesian Methods for Adaptive Models, Ph.D. dissertation, California Institute of Technology, Pasadena, California, USA.

Mandal, P., Senjyu, T., Uezato, K., Funabashi, T. (2005). Several-Hours-Ahead Electricity Price and Load Forecasting Using Neural Networks, *IEEE PES General Meeting, San Francisco, USA.*

Neal, R.M. (1996). Bayesian Learning for Neural Networks, Lecture Notes in Statistics, n.118, Springer-Verlag, New York.

Ramanathan, R., Engle, R., Granger, C.W.J., Vahid-Araghi, F., Brace, C. (1997). Short-Run Forecasts of Electricity Loads and Peaks, *International Journal of Forecasting*, v.13, n.2, pp. 161-174.

Rulkov, N.F., Sushchik, M.M., Tsimring, L.S.; Abarbanel, H.D.I. (1995). Generalized Synchronization of Chaos in Directionally Coupled Chaotic Systems, *Physical Review E*, v.51, n.2, pp. 980-994.

Schölkopf, B., Smola, A.J., (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond, Cambridge, Massachusetts.

Takens, F. (1981). Detecting Strange Attractors in Turbulence, *In.: D.A. Rand, L.-S. Young (eds.), Dynamical Systems and Turbulence, Lecture Notes in Mathematics*, v.898, pp. 366-381, Springer-Verlag.

Tipping, (2001). Sparse Bayesian Learning and the Relevance Vector Machine, *Journal of Machine Learning Research*, v.1, pp. 211-244.

Vapnik (1998). Statistical Learning Theory, New York, John Wiley & Sons.