A study on the ability of Support Vector Regression and Neural Networks to Forecast Basic Time Series Patterns

Sven F. Crone[1], Jose Guajardo[2], and Richard Weber[2]
1 Lancaster University, Department of Management Science, Lancaster LA1 4YX, Lancaster, UK s.crone@lancaster.ac.uk
2 University of Chile, Department of Industrial Engineering, Republica 701, Santiago, Chile {jguajard,rweber}@dii.uchile.cl

Abstract. Recently, novel learning algorithms such as Support Vector Regression (SVR) and Neural Networks (NN) have received increasing attention in forecasting and time series prediction, offering attractive theoretical properties and successful applications in several real world problem domains. Commonly, time series are composed of the combination of regular and irregular patterns such as trends and cycles, seasonal variations, level shifts, outliers or pulses and structural breaks, among others. Conventional parametric statistical methods are capable of forecasting a particular combination of patterns through ex ante selection of an adequate model form and specific data preprocessing. Thus, the capability of semi-parametric methods from computational intelligence to predict basic time series patterns without model selection and preprocessing is of particular relevance in evaluating their contribution to forecasting. This paper proposes an empirical comparison between NN and SVR models using radial basis function (RBF) and linear kernel functions, by analyzing their predictive power on five artificial time series: stationary, additive seasonality, linear trend, linear trend with additive seasonality, and linear trend with multiplicative seasonality. Results obtained show that RBF SVR models have problems in extrapolating trends, while NN and linear SVR models without data preprocessing provide robust accuracy across all patterns and clearly outperform the commonly used RBF SVR on trended time series.

1 Introduction

Support Vector Regression (SVR) and Artificial Neural Networks (NN) have found increasing consideration in forecasting theory, leading to successful applications in time series and explanatory forecasting in various domains, including business and management science [1, 2]. Methods form computational intelligence promise

attractive features to business forecasting, being data driven, semi-parametric learning machines, permitting universal approximation of arbitrary linear or nonlinear functions from examples without a priori assumptions on the model structure, often outperforming conventional statistical approaches of ARIMA- or exponential smoothing- methods.

Despite their theoretical capabilities, NN as SVR are not established forecasting methods in business practice. Recently, substantial theoretical criticism of NN has raised skepticism regarding their ability to forecast even simple time series patterns of seasonality or trends without prior data preprocessing [3]. While all novel methods must ultimately be evaluated in an objective experiment using a number of empirical time series, adequate error measures and multiple origins of evaluation [4], the fundamental questions to their ability to approximate and generalize basic time series patterns must be evaluated beforehand. Time series can generally be characterized by the combination of basic regular patterns: level, trend, season and residual errors. For trend, a variety of linear, progressive, degressive and regressive patterns are feasible. For seasonality, an additive or multiplicative combination with level and trend further determines the shape of the final time series. Consequently, we evaluate SVR and NN on a set of artificially created time series derived from previous publications. We evaluate the comparative forecasting accuracy of each method to reflect their ability of learning and forecasting fundamental time series patterns relevant to empirical forecasting tasks.

This paper is organized as follows. First, we provide a brief introduction to SVR and NN in forecasting time series of observations. Section three presents the artificially generated time series and the experimental design. This is followed by the experimental results and their discussion. Conclusions are given in section 4.

2 Modelling SVR and NN for Time Series Prediction

2.1 Support Vector Regression

We apply the common Support Vector Regression (SVR) algorithm as proposed by Vapnik [5], which uses an ε-insensitive loss function for predictive regression problems. This function allows a tolerance degree to errors not greater than ε. The description is based on the terminology used in [6, 7]. Let $\{(x_1,y_1),\ldots..,(x_\ell,y_\ell)\}$, where $x_i \in R^n$ and $y_i \in R$, be the training data points available to build a regression model. The SVR algorithm applies a transformation function Φ to the original data points from the initial Input Space, to a higher-dimensional Feature Space F. In this new space, we construct a linear model, which corresponds to a non-linear model in the original space[1]:

[1] When Φ is the identity function, the Feature Space is equivalent to the Input Space, and the model constructed is linear in the original space.

$$\Phi : R^n \to F, w \in F$$
$$f(x) = \langle w, \Phi(x) \rangle + b$$

The goal when using the ε-insensitive loss function is to find a function that fits current training data with a deviation less or equal to ε, and at the same time is as flat as possible. This means that one seeks for a small weight vector w; one way to do that is e.g. by minimizing the quadratic norm of the vector w [6]. As this problem could be infeasible, slack variables $\xi i, \xi i^*$ are introduced to allow error levels greater than ε, arriving to the formulation proposed in [5]:

$$Min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{\ell}(\xi_i + \xi_i^*)$$
$$s.t. y_i - \langle w, \Phi(x_i) \rangle - b \le \varepsilon + \xi_i$$
$$\langle w, \Phi(x_i) \rangle + b - y_i \le \varepsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \ge 0, i = 1,2,...,\ell$$

This is known as the primal problem of the SVR algorithm. The objective function takes into account generalization ability and accuracy in the training set, and embodies the structural risk minimization principle [8]. Parameter C measures the trade-off between generalization ability and accuracy in the training data, and parameter ε defines the degree of tolerance to errors. To solve the problem stated above, it is more convenient to represent the problem in its dual form. For this purpose, a Lagrange function is constructed, and once applying saddle point conditions, it can be shown that the following solution is obtained [8]:

$$w = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)\Phi(x_i)$$

$$f(x) = \sum_{i=1}^{\ell}(\alpha_i - \alpha_i^*)K(x_i, x) + b$$

Here, α_i and α_i^* are the dual variables, and the expression $K(x_i,x)$ represents the inner product between $\Phi(x_i)$ and $\Phi(x)$, which is known as the kernel function [8]. The existence of such a function allows us to obtain a solution for the original regression problem, without explicitly considering the transformation $\Phi(x)$ applied to the data. In our experiments we use radial basis functions (RBF) and linear kernel functions.

Limited research has been conducted to investigate the ability of SVR for predicting different time series patterns. Experiments performed by Hansen et. al [9] compare SVR performance with 3 statistical methods (e.g. ARIMA) on predicting 9 different patterns present in real world time series. Among other patterns, they tried trends, seasonality, cycles, and combinations of them. Their experiments show SVR models outperforming the other methods on 8 of the 9 patterns; particularly, they obtained very good results using SVR for extrapolating linear and non linear trends. Guajardo et al. [10] compared SVR with ARMAX models for predicting seasonal time series in a weekly sales forecasting domain for 5 different products. Their experiments show that SVR were slightly better than ARMAX models, succeeding in extrapolating seasonal patterns (without trends) with SVR.

2.2 Neural Networks

Forecasting with non-recurrent NN may encompass prediction of a dependent variable $\hat{y}$ from lagged realizations of the predictor variable y_{t-n}, 1 or *i* explanatory variables x_i of metric, ordinal or nominal scale as well as lagged realizations thereof, $x_{i,t-n}$. Therefore, NNs offer large degrees of freedom towards the forecasting design, permitting explanatory or causal forecasting through estimation of a functional relationship of the form $\hat{y} = f(x_1, x_2, ..., x_z)$, as well as general transfer function models and simple time series prediction. Following, we present a brief introduction to modelling NN for time series prediction; a general discussion is given in [11, 12].

Forecasting time series with NN is generally based on modelling the network in analogy to a non-linear autoregressive AR(*p*) model [2, 13]. At a point in time *t*, a one-step ahead forecast $\hat{y}_{t+1}$ is computed using *p=n* observations $y_t, y_{t-1}, \ldots, y_{t-n+1}$ from *n* preceding points in time *t, t-1, t-2, ..., t-n+1*, with *n* denoting the number of input units of the NN. This models a time series prediction as of

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, ..., y_{t-n+1}) .$$

The architecture of a feed-forward Multilayer Perceptron (MLP), a well researched NN paradigm, of arbitrary topology is displayed in figure 1.

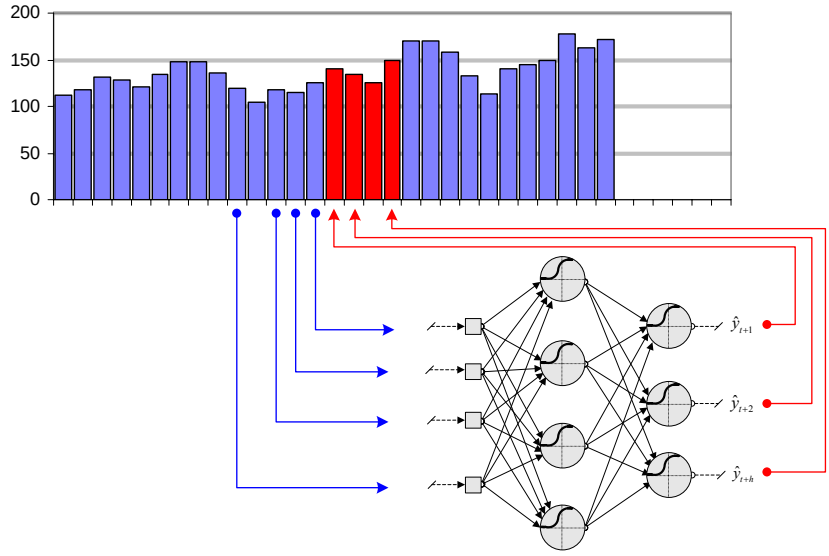

Fig. 1. Autoregressive MLP application to time series forecasting with a MLP of arbitrary topology, using *n* input neurons for observations in *t, t-1, t-2, ..., t-n-1*, *m* hidden units, *h* output units for time periods *t+1, t+2, ..., t+h* and a two layers of trainable weights. The bias is displayed within the units.

Data is presented to the MLP as a sliding window over the time series observations. The task of the MLP is to model the underlying generator of the data during training, so that a valid forecast is made when the trained NN is subsequently presented with a new input vector value.

The network paradigm of MLP offers extensive degrees of freedom in modelling for prediction tasks. Structuring the degrees of freedom, each expert must decide upon the selection and sampling of datasets, the degrees of data preprocessing, the static architectural properties, the signal processing within nodes and the learning algorithm in order to achieve the design goal, characterized through the objective function or error function. For a detailed discussion of these issues and the ability of NN to forecast univariate time series, the reader is referred to [2].

3 Experiments and Results

3.1 Description of the Artificial Time Series

We evaluate a set of five artificial time series of monthly retail sales motivated from Pegel's original classification, later extended by Gardner to incorporate degressive trends. Time series are composed of regular patterns of different forms of linear, progressive, degressive or regressive trends T, additively or multiplicatively combined with seasonality S, a constant level L and residual noise E. In addition, empirical time series are impacted by irregular patterns such as level shifts and pulses, which are disregarded. To evaluate the ability of different computational intelligence methods we create a set of benchmark time series for the most common regular time series patterns: linear trend and different forms of seasonality. Consequently, we create individual time series patterns and combine them accordingly, overlaying each with additive noise.

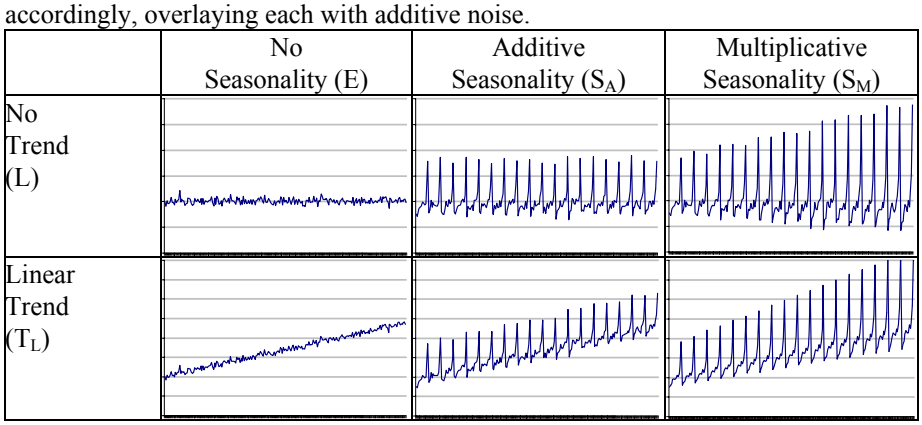

Fig. 2. Basic time series patterns of artificial time according to the Pegels- and Gardner-classification, combining Level, Trend and Seasonality with a medium additive noise level.

In contrast to Pegel's classification, a time series with multiplicative seasonality $L+S_M+E$ cannot display an increasing seasonality in the absence of level changes, it equals the pattern of additive seasonality and was consequently omitted from further analysis. Consequently, we create a set of five time series including a stationary time series L+E (E), seasonality without trend $L+S_A+E$ (S_A), linear trend $L+T_L+E$ (T_L), linear trend with additive seasonality $L+T_L+S_A+E$ (T_LS_A) and linear trend with multiplicative seasonality depending on the level of the time series $L+T_L*S_M+E$

(T_LS_M). The residual error term follows a Gaussian distribution $N(0,\sigma^2)$ applying a medium level of noise $\sigma^2 = 25$. The original time series data was taken from the experiments of [3] and represent monthly retail sales. All time series considered an additive noise term to allow an estimation of final forecasting accuracy in relationship to the original noise level. Each time series consists of 228 observations.

3.2 Experimental Design

This research investigates whether the five patterns described above can be accurately predicted with RBF SVR, Linear SVR and NN models. For each series, we defined a lag structure including the 13 previous observations as attributes for predicting the next series value (one period ahead prediction); thus, a total of 215 data points remain to build and parameterize models. Data was sequentially divided into training, validation and test sets using 119, 48 and 48 observations respectively; training data is used to build the model, validation data for parameter selection purposes, and test data to evaluate the accuracy on a hold-out data set. All models are parameterized using only training and validation data, withholding all information in the test set (also for scaling etc.) to assure valid ex ante testing. Data was transformed only by applying linear scaling into a [-0.5, 0.5] interval to avoid saturation effects, using minimum and maximum values only from the training and validation data. No other preprocessing procedures such as deseasonalization or detrending were carried out.

As mentioned in section 2.1., SVR models require setting of two parameters: *C* and *ε*. In addition, one needs to select an appropriate kernel function to carry out the transformation to a higher dimensional feature space. The RBF kernel function, which is the kernel function most widely utilized for regression (see e.g. [6, 14, 15]), requires the definition of an additional parameter σ. Our heuristic approach for RBF SVR parameter selection can be summarized as follows:

- First, we determine starting values for the C and ε parameters on each time series by using the empirical rules proposed by Cherkassky and Ma [14], leading to E {C=0.67538; ε=0.020373}, S_A {C=0.86224; ε=0.0056657}, T_L {C=0.70709; ε=0.0043011}, T_LS_A {C=0.74641; ε=0.0064901} and T_LS_M {C=0.76968; ε=0.0064652}.
- Second, we search for 'good' values of the RBF kernel parameter σ using the predetermined parameters C and ε, and evaluate 45 different alternatives for σ={0.001; 0.01; 0.03; 0.05; 0.08; 0.1; 0.3; 0.5; 0.8; 1; 1.3; 1.5; 1.8; 2; 2.3; 2.5; 2.8; 3; 3.3; 3.5; 3.8; 4; 4.3; 4.5; 4.8; 5; 5.3; 5.5; 5.8; 6; 7; 8; 9; 10; 15; 20; 25; 50; 80; 100; 200; 300; 400; 500; 1000}. The value of σ which generates the model with the lowest mean absolute error (MAE) in the validation set is defined as the base parameter for the kernel function. As result, we now have heuristic starting values for the three parameters of the SVR model, C', ε' and σ'.
- Third, we define a grid around base parameters C', ε' and σ', and retain the best combination of parameters to be the final values used in the SVR model. In our experiments, we tried five different values for each parameter C, ε, σ (factors

0.5, 0.75, 1, 1.25 and 1.5 over the initial values), thus creating a grid of 125 possible parameter settings. The parameter candidate of the grid is selected by using the lowest MAE on the validation set as before.

The scheme for Linear SVR is very similar, but without considering parameter σ. Thus, second step for base parameter σ is not carried out, and the third step involves only 25 different combinations for C and ε. (for additional details see [10]).

For NN models, we used the backpropagation algorithm to train multiple candidates of multilayer perceptron (MLP) networks. The network topology was obtained using a grid search of different hidden nodes {0, 2,…, 20} and activation functions {sigmoid; tanh} with fixed number of input and output nodes, selecting the architecture with the lowest MAE on the validation set. The final model was initialized 20 times using an (13-8-1) architecture comprised of 13 input nodes, 8 hidden nodes and a single output node for *t+1* predictions, applying a sigmoid transfer function between the input and hidden layers, and a linear function between hidden and output layers. As for SVR models, we selected the network with the lowest validation mean absolute error (MAE) to calculate the test error results.

3.3 Experimental Results and Discussion

To evaluate our models we used the root mean squared error (RMSE), mean absolute error (MAE), and mean absolute percentage error (MAPE). Test set errors obtained using SVR and MLP models for each one of the five series analyzed in this paper are shown in Table 1. As can be seen from Table 1, RBF SVR has the best performance (denoted in bold) on a level time series superimposed with white noise (E) and additive seasonality (S_A) patterns across all error measures. Linear SVR is the best method for predicting linear trend (T_L) and linear trend with multiplicative seasonality patterns (T_LS_M), while MLPs provide best results for linear trends with additive seasonality (T_LS_A) pattern.

Table 1. Forecasting accuracy on the test set for RBF and linear SVR models and MLP

Series	RBF SVR			Linear SVR			MLP		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Series E	**4.670**	**3.776**	1.387	5.036	4.108	1.496	4.851	3.946	**1.264**
Series S_A	**4.746**	**3.739**	**0.039**	6.961	5.637	0.058	5.787	4.766	0.048
Series T_L	11.501	10.408	0.046	**5.876**	**4.811**	**0.021**	6.058	4.966	0.022
Series T_LS_A	21.267	17.678	0.075	7.915	6.280	0.028	**7.083**	**5.878**	**0.027**
Series T_LS_M	14.758	10.842	0.043	7.927	**6.305**	**0.029**	**7.673**	6.454	0.030
Sum	56.942	46.443	1.590	33.715	27.141	1.632	**31.452**	**26.010**	**1.391**

Since we evaluated artificially constructed time series we can estimate the part of the forecasting errors caused by the artificially created noise, which due to its random nature cannot be forecasted. This permits an analysis to what extent each method was capable of separating noise from structure of varying complexity on the unbiased error measure of MAE. In applying the true mean of the Gaussian residuals

as an optimal forecast, we estimate a MAE of 3.801 as a lower bound forecast error for all time series on the test set. It becomes apparent, that RBF SVR exceeds even a 'perfect' forecast for series E and S_A, which can be attributed to the randomness of the data inherent in all ex ante evaluations of forecasting experiments. In contrast, RBF SVR significantly underperform on trended time series patterns, indicating inadequacies of the chosen kernel function. On the contrary, linear SVR shows a more robust prediction of all time series patterns. As the forecasts deviates only slightly from the lower bound in comparison to the level of the time series, as would be reflected in the MAPE, SVR with linear kernel functions may be considered a robust method in forecasting arbitrary time series patterns without preprocessing. Similarly, MLPs forecast all time series patterns robustly and without preprocessing with a comparative high accuracy close to linear SVR and the lower bound.

In summarizing over all time series, applying an equal weight to each of the time series patterns, MLPs robustly outperform RBF SVR on all three error measures of MAE, MAPE and RMSE, whereas MLPs also moderately outperform linear SVR. This indicates that while particular kernel functions enable the SVR to outperform alternative parameterizations, MLPs or linear SVR may prove a more robust alternative in using a single method to forecast a set of time series of different patterns. In addition to these distance based error measures, we evaluate the relative performance by ranking each method by the individual error measure, provided in Table 2.

Table 2. Forecasting accuracy measured by ranks of methods for each error measure

	Rank by RMSE			Rank by MAE			Rank by MAPE		
	SVR RBF	SVR linear	MLP	SVR RBF	SVR linear	MLP	SVR RBF	SVR linear	MLP
Series E	**1**	3	2	**1**	3	2	2	3	**1**
Series S_A	**1**	3	2	**1**	3	2	**1**	3	2
Series T_L	3	**1**	2	3	**1**	2	3	**1**	2
Series T_LS_A	3	2	**1**	3	2	**1**	3	2	**1**
Series T_LS_M	3	2	**1**	3	**1**	2	3	**1**	2
Sum of Ranks	11	11	**8**	11	10	**9**	12	10	**8**

The findings by ranked error measures confirm little differences between linear SVR and MLPs, with MLPs providing the best results for the limited test design provided across all error measures. SVR with RBF kernel, the most frequently used implementation in time series prediction with SVR to date, performs significantly worse than the other methods.

As must be expected, different error measures identify different 'best' methods. In particular, RMSE and MAPE are considered to be biased error measures. To limit biases in the absence of a true objective function which could motivate the use of a particular error measure, we assume equal weight to each error and focus our conclusions on the MAE. To confirm the results of model accuracy from a statistical point of view, we performed a paired-samples t test on the absolute values of the residuals over the test set data points. Results obtained show that differences between

model errors are statistically significant when comparing RBF SVR to Linear SVR ($t=7.337$; $df=239$; $p<0.001$), and RBF SVR to NN ($t=6.999$; $df=239$; $p<0.001$), although not when comparing NN to Linear SVR ($t=-0.989$; $df=239$; $p=0.324$). This indicates that no significant difference in forecasting accuracy between the methods of linear SVR and MLP may be derived from these experiments. Consequently, we need to extend this evaluation on additional time series and variations of MLPs. Results suggest that RBF SVR can predict seasonal patterns but no trends, while linear SVR and NN seem to be able to extrapolate trend as well as seasonal patterns accurately and without preprocessing. By examining the residuals of the models, it can be observed that RBF SVR systematically underestimate hold-out sample observations for trended series, which corresponds to saturation effects.

4 Conclusion

We have examined the ability of RBF SVR, linear SVR and MLP for predicting five basic artificial time series patterns: stationary, seasonality, linear trends, linear trend with additive seasonality, and linear trend with multiplicative seasonality. Results obtained using multiple error measures show that while RBF SVR outperform other methods on non-trended data, they do not provide robust results across all patterns. For time series with trend components, linear SVR and MLP significantly outperform RBF SVR models, which severely underestimate out-of-sample observations, consistently lagging behind upward trends. RBF SVR errors have shown to be statistically significantly higher than linear SVR and NN errors. MLP demonstrate robust performance, providing the highest overall forecasting accuracy in across time series and different statistical error measures and rank based metrics.

Our results confirm previous findings by Guajardo et. al [10], demonstrating accurate forecasts of seasonal time series without trends using RBF SVR, even outperforming established statistical methods such as ARIMAX. Also, they confirm results by Hansen et. al [9], who accurately predicted both linear and nonlinear trends using SVR, outperforming other methods such as ARIMA on several patterns. We assume that Hansen et al. also used linear kernels, as they did not fully document the kernel functions applied. A preliminary hypothesis for our poor results obtained with RBF SVR in extrapolating trend patterns lies in the linear nature of this trend. Previous publications report similar problems of closely related RBF-neural networks in predicting trends and instationary time series. While SVR with linear kernel functions and MLP with linear activation functions in the output units may be particularly suited to extrapolate linear trends, we did not conduct experiments as to their ability to extrapolate non-linear trends.

These issues will be evaluated in an extended set of experiments currently under investigation by the authors, increasing the number of time series patterns and considering additional kinds of trend patterns, also evaluating results against established statistical forecasting methods as benchmarks. Additionally, we will

evaluate the influence of preprocessing procedures such as deseasonalization to evaluate alternative perspectives on the problem of extrapolating time series patterns.

Acknowledgement: This work has been supported in part by the Millennium Nucleus "Complex Engineering Systems" (www.sistemasdeingenieria.cl).

References

1. K. P. Liao and R. Fildes, The accuracy of a procedural approach to specifying feedforward neural networks for forecasting, Computers & Operations Research 32 (8) (2005) 2151-2169.
2. G.P. Zhang, B.E. Patuwo, and M.Y. Hu, Forecasting with artificial neural networks: The state of the art, *International Journal of Forecasting*, 1, **14,** 35-62 (1998)
3. G.P. Zhang and M. Qi, Neural network forecasting for seasonal and trend time series, *European Journal of Operational Research* **160,** 501-514 (2005).
4. L. J. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, *International Journal of Forecasting* 16 (4) (2000) 437-450.
5. V.Vapnik, *The nature of statistical learning theory* (Springer, New York, 1995).
6. A.J. Smola and B. Schölkopf, A Tutorial on Support Vector Regression, NeuroCOLT Technical Report NC-TR-98-030, 1998 (Royal Holloway College, University of London, UK).
7. K. Müller, A. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, in: Advances in Kernel Methods: Support Vector Learning/ Using Support Vector Machines for Time Series Prediction, edited by B. Schölkopf, J. Burges, and A. Smola (MIT Press, 1999) , pp. 243-254.
8. V.Vapnik, *Statistical Learning Theory* (John Wiley and Sons, New York, 1998).
9. J.V. Hansen, J.B. McDonald, and R.D. Nelson, Some evidence on forecasting time-series with Support Vector Machines, *Journal of the Operational Research Society*, 1, 1-11, 2005.
10. J. Guajardo, J. Miranda, and R. Weber, A Hybrid Forecasting Methodology using Feature Selection and Support Vector Regression, 5th International Conference on Hybrid Intelligent Systems HIS 2005 (Rio de Janeiro, Brazil, 2005), pp. 341-346.
11. C. M. Bishop, Neural networks for pattern recognition. Clarendon Press; Oxford University Press, Oxford, 1995.
12. S. Haykin, Neural networks: a comprehensive foundation, 2nd ed. Prentice Hall, Upper Saddle River, N.J., 1999.
13. A. Lapedes, R. Farber, and Los Alamos National Laboratory, Nonlinear signal processing using neural networks: prediction and system modelling, Los Alamos National Laboratory, Los Alamos, N.M. LA-UR-87-2662, 1987.
14. V. Cherkassky and Y. Ma, Practical selection of SVM parameters and noise estimation for SVM regression, *Neural Networks* **17**(1), 113-126 (2004).
15. D. Mattera and S. Haykin, in: Advances in Kernel Methods: Support Vector Learning/ Support Vector Machines for Dynamic Reconstruction of a Chaotic System, edited by B. Schölkopf, J. Burges and A. Smola (MIT Press, 1999), pp. 211-242.