Parameter Sensitivity of Support Vector Regression and Neural Networks for Forecasting

Sven F. Crone, Stefan Lessmann and Swantje Pietsch

Abstract— **Support Vector Regression (SVR) and artificial Neural Networks (NN) promise attractive features for time series forecasting. Despite their attractive theoretical properties, limited empirical studies using small or unbalanced parameter setups yield inconsistent results regarding their empirical accuracy. This paper investigates the accuracy of different configurations of NN and SVR parameters, paying particular attention to the common SVR kernels of polynomial, radial basis functions, sigmoid and linear functions through an exhaustive empirical comparison. We investigate the forecasting performance of alternative parameter setups with established benchmarks, evaluating all models on 36 artificial time series with archetypical patterns of level, trend, seasonality and trend-seasonality. As a result, we find that SVR and NN outperform statistical methods on particular time series patterns. Forecasting performance of SVR and NN is impacted by choice of parameters, indicating NN and SVR with the RBF kernels as robust choices on most time series forecasting problems.**

I. Introduction

Accurate corporate decisions in an uncertain future environment requires accurate forecasting [1, 2]. As a consequence, significant effort has been invested in developing forecasting methods with enhanced forecasting accuracy, extending established statistical approaches of Exponential Smoothing and ARIMA-methods towards nonlinear methods of ARCH, GARCH, STAR etc. and methods of computational intelligence [3]. While statistical methods are embedded in a methodology following iterative model building and parameterization based upon a prior theory, methods from computational intelligence such as support vector regression (SVR) and neural networks (NN) are semi-parametric, data-driven methods which capture the underlying linear and/or non-linear model form and suitable parameters directly from the data [4]. Thus they offer promising features to applications to various business forecasting domains where limited tangible knowledge on an underlying model form exists, such as accounting, finance, marketing, economics, production, tourism, transportation etc. [2, 5].

However, in contrast to statistical methods, NN as well as SVR have not been established in business practice despite their attractive theoretical properties. In particular, the limited empirical studies on time series forecasting with NN and SVR have provided only mixed results on their performance [6]. Substantial empirical criticism towards NN has raised doubts to their ability to forecast even simple time series patterns of seasonality or trends without adequate data pre-processing [4]. However, Hansen and Nelson indicate that these comparisons may be misleading, as they often aim to identify one true and superior method, which may be infeasible given the importance of the context of complex forecasting problems and different data properties [7]. Also, Liao and Fildes point out that missing documentation and unfixed parameter designs as well as unsuitable benchmarks provide only one-sided results with limited insights [8]. As a consequence, this paper investigates the accuracy of NN and SVR time series forecasting with various archetypical time series of seasonal, trend and trend-seasonal patterns with increasing levels of noise. In particular, we study the effectiveness of alternative parameters of SVR and kernel functions and NN architectures on their forecasting performance to derive the conditions under which they perform well. The exhaustive modelling evaluates a total of 33,120 NN and 4,327,152 SVR models using a consistent methodology [8] in comparison to a statistical benchmark method and on multiple established error measures to avoid evaluation biases.

This paper is organised as follows: next we provide a brief introduction to SVR and NN to time series forecasting. Section three presents an overview of the experimental design including the artificially generated time series and the obtained results, followed by conclusions.

II. Methods of Computational Intelligence

A. Artificial Neural Networks

Computational Intelligence methods are data-driven, self adaptive methods, which learn from examples and capture subtle functional relationships from the data, even if the underlying relationships are unknown or hard to describe [3]. They promise attractive features to business forecasting because of their inherent capability of learning arbitrary input–output mappings from examples without a priori assumptions on the model structure and their ability to generalize for future values of the dependent variables [3, 9].

NNs represent a class of distinct mathematical models originally motivated by the information processing in biological neural systems [10-13]. In this study a special class of NN, the Multi Layer Perceptron (MLP) is used. MLPs are feed forward neural networks which are typically

Sven F. Crone (corresponding author), Department of Management Science, Lancaster University Management School, Lancaster LA1 4YX, United Kingdom (phone +44.1524.5-92991; e-mail: s.crone@lancaster.ac.uk).
Stefan Lessmann, Swantje Pietsch, Institute of Information Systems, University of Hamburg, 20146 Hamburg, Germany (e-mail: Lessmann@econ.uni-hamburg.de; mailing@swantje-pietsch.de).

composed of several layers of nodes with unidirectional connections [3], often trained by back propagation [14]. MLPs are capable of predicting non linear time series patterns through the information processing in hidden layers [3, 15], frequently modeling a NARX(*p*) process or an ARX(*p*) processes equivalent to linear statistical forecasting models without nonlinear activation functions [3, 15].

The components of MLPs are the topology, the activation functions, the propagation functions, the connecting network, the input signal combination rules and the learning algorithms [12]. The many degrees of freedom in the modeling process require a number of decisions to fully specify a network architecture and to assure valid and reliable performance [8, 9]. This also includes decisions as choosing the accuracy measure [15], learning rates, sampling method, initializing weights, biases, cooling factor, initialization interval, activation function headroom, stopping rule and number of training iterations [8, 14, 16].

The training input data is in form of vectors [3] of the time lagged observations [7], whereas the number of input nodes corresponds to the number of lagged observations [4]. The input vector is almost composed as a sliding window over the time series observations [3, 9]. At this study a linear function for the input nodes is used. The incoming signal at the nodes of the next layer is combined by the summing method [12]. The nodes of the hidden layer use non linear hyperbolic tangent (TanH) activation functions, which is a sigmoid, continuous, fully differentiable function and is defined in]-1;1[[10]. It determines the relationship between the inputs and outputs of a node and a network [1]. Just one output neuron with identity function is used. The network is in matrix form and rules the signal traffic with arc weights [10], which are estimated from the data by minimizing the sum of squared errors (SSE) of the within sample one-step-ahead forecast errors over the validation set [3, 17]. The propagation function regulates the output signal with an on-neuron bias w_{co} [10]. To demonstrate an example, the functional form for a feed forward NN with one layer may be written as [14]

$$\hat{y}_{t+1} = f_{Id}\left(w_{co} + \sum_h w_{ho} f_{\tanh}\left(w_{ch} + \sum_i w_{ih} y_{t-j+1}\right)\right) \quad (1)$$

Before a NN can be used to perform any desired task, it must be trained. Basically, training is the process of determining the arc weights which are the key elements of an NN [3]. The training vector is given to the input nodes and transformed by the arc weights w_{ch} before they are accumulated at each node of the successive layer. This value is transformed by an activation function and weighted w_{ho} between this layer and output node [3]. The output is compared with the teaching input and the weights are adjusted by a gradient descent method, which is the back propagation method in this study. This is repeated with the goal to minimize the error between the actual and desired system behavior [2] and the knowledge is stored in the arcs and nodes in form of arc weights and node biases [3].

B. Support Vector Regression

Unlike the NN, the training problem of the SVR is a convex optimization problem without local minima [18, 19] and methodically bases upon the statistical learning theory by Vapnik [11]. Thus, it constructs a regression model, which minimizes some empirical risk while it simultaneoulys regularizes the capacity [19]. The SVR is an advancement of the SVM and applicable for time series prediction. The aim of the SVR is to estimate an unknown function from the training dataset $((\mathbf{x}_i, y_1),\ldots,(\mathbf{x}_\ell, y_\ell)) \subseteq (\mathbf{X} \times Y)^\ell$, that has at most deviation in the *ε*-tube and is also as flat as possible [20-22]. For time series prediction the vector $\mathbf{x}$ contains the lag structure and *y* is the prediction value at the target time [23]. Typically the estimated regression function is written as

$$f(\mathbf{x}_i) = \sum_{i=1}^{\ell} (\alpha_i - \alpha_i^*)\,\mathrm{k}(\mathbf{x_i} \cdot \mathbf{x_j}) + b \quad (2)$$

where the Lagrange multipliers α_i, α_i^* act as forces pushing and pulling the predictions towards the target value *y* [24]. To get the Lagrange multipliers, the empirical risk and the capacity control term are formulated as convex optimization problem, which yields to the following dual optimization problem

$$\mathit{maximize}\begin{cases} \frac{1}{2}\sum_{i,j=1}^{\ell}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(\mathbf{x_i}, \mathbf{x_j}) \\ -\varepsilon\sum_{i=1}^{\ell}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{\ell} y_i(\alpha_i - \alpha_i^*) \end{cases} \quad (3)$$

$$\mathit{subject\ to} \quad \sum_{i=1}^{\ell} \alpha_i^+ - \alpha_i^- = 0 \text{ and } \alpha_i^+, \alpha_i^- \in [0, C]$$

with $i = 1,\ldots,\ell$, which is here formulated for the *ε*-insensitive case [25-28]. The term $\mathrm{k}(\mathbf{x_i} \cdot \mathbf{x_j})$ is the kernel function, which enables the SVR to predict non linear data. Instead of explicitly mapping the non linear data into a high-dimensional feature space F, the data are implicitly transformed into F by a kernel function. This is just applicable, as SV algorithms only depend on dot products between patterns. The common kernel functions used are the linear (4), the radial basis function (RBF) (5), the polynomial (6) and the sigmoid (7) [28-30]:

$$k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}', \quad (4)$$

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\gamma |\mathbf{x} - \mathbf{x}'|^2\right), \gamma > 0, \quad (5)$$

$$k(\mathbf{x}, \mathbf{x}') = \left(\gamma\, \mathbf{x}^T \cdot \mathbf{x}' + r\right)^p, \gamma > 0, \quad (6)$$

$$k(\mathbf{x}, \mathbf{x}') = \tanh\left(\gamma\, \mathbf{x}^T \cdot \mathbf{x} + r\right), \quad (7)$$

Kernel functions enable dot products to be performed in high-dimensional feature space using low dimensional space data input without knowing the transformation explicitly. They must satisfy Mercer’s condition that corresponds to the inner product of some feature space. The RBF is most commonly used as the kernel for regression [31]. For the RBF case, the number of centers, the center-locations, the weights, and the threshold (*b*) are all produced automatically during parameterisation [28, 32]. As the number of hyper parameters influences the complexity of model selection, the polynomial kernel is often avoided due to its increased number of hyper parameters in comparison to the RBF and sigmoid kernel functions [29]. The sigmoid kernel was quite popular for support vector machines due to its origin from neural networks. Due to its conditionally positive definiteness, the quality of the local minimum solution in other parameters may not be guaranteed [33], making it potentially hard to select suitable parameters for the sigmoid kernel. Furthermore, all kernel functions corresponding to the inner product of some feature space, must satisfy the Mercer's conditions [28, 34, 35].

The degrees of freedom in the modeling process of SVR are often considered to be less than for a NN modeling process, but also must be defined ex ante. The Parameter *C* determines the trade of between the model capacity and the amount of outliers of the ε-tube [27, 36]. If *C* is selected too large, the objective is to minimize the empirical risk only, without regard to the model capacity [37]. In this study the ε-insensitive loss function is used, where only data outside the ε-tube are considered [27, 36], which signify that only the data points are support vectors, for which the Lagrange multipliers are non-zero [31]. Thus, the parameter ε affects the number of support vectors. As the support vectors are the data points that describe the searched function [20, 38], a larger ε allows for fewer support vectors to construct the regression function, which results in a flat regression model [37]. Another parameter, which has to be selected a priori, is the kernel parameter γ which reflects the input range of the training data [37]. In addition, for the polynomial kernel function the degree of the polynomial *d* has to be specified as well.

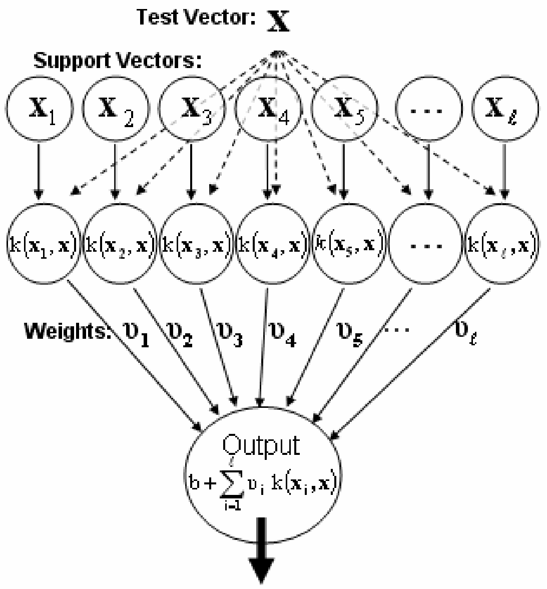
Test Vector: $\mathbf{x}$
Support Vectors:
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
$\mathbf{x}_4$
$\mathbf{x}_5$
...
$\mathbf{x}_\ell$
$k(\mathbf{x}_1,\mathbf{x})$
$k(\mathbf{x}_2,\mathbf{x})$
$k(\mathbf{x}_3,\mathbf{x})$
$k(\mathbf{x}_4,\mathbf{x})$
$k(\mathbf{x}_5,\mathbf{x})$
...
$k(\mathbf{x}_\ell,\mathbf{x})$
Weights: υ_1 υ_2 υ_3 υ_4 υ_5 ... υ_ℓ
Output
$b+\sum_{i=1}^{\ell}\upsilon_i\, k(\mathbf{x}_i,\mathbf{x})$

Figure1: This figure demonstrates the SVR prediction process [39].

The kernel function determines the dot products between the input pattern and the support vectors, which are multiplied by the weights $\upsilon_i = \alpha_i - \alpha_i^*$ and become accumulated. This plus the threshold b yield in the final prediction value. As most parameters are of metric scale, the degrees of freedom in ex ante parameter selection appear as large as for NN.

III. Empirical Simulation Experiment

A. Time Series Patterns

We evaluate SVR and NN model candidates on 36 artificial time series, which are derived from an empirical dataset of retail data from Zhang and Qi [6]. The time series components of trend and season are decomposed and recombined individually, building the patterns according to the Pegels-Gardner framework of trend-seasonal patterns [15]. The time series features are stationary (E), linear trend (LT), exponential trend (ET), degressive trend (DT), additive seasonality (AS) and multiplicative seasonality (MS), which are combined to E, LT, ET, DT, LT_AS, LT_MS, ET_AS, ET_MS, DT_AS, DT_MS patterns with [6] three increasing levels of normal distributed noise, $N(0,\sigma^2)$, with $\sigma^2 = 1, 25, 100$ [39]. Each monthly time series consist of 228 observations which are split into 180 for the training set, 48 for the validation set and 48 for the test set, i.e. [60%, 20%, 20%]. Thus 48 periods are withheld for out of sample, ex ante comparison. The statistical benchmarks of ARIMA or exponential smoothing are computed using the expert software system ForecastPro, utilizing 180 observations for model selection and parameterization, as no validation data set is needed.

B. Experimental Setup

Developing SVR or NN models for a particular forecasting application is a non-trivial task which is critical for the performance of the competing methods, because of the potential impacts on learning and generalization [5]. As no generic methodology for modelling either SVR or NN exists for arbitrary datasets, we evaluate a range of candidate models with alternative parameterisations across the 36 time series.

SVR modeling also requires setting and tuning of multiple parameters ex ante, so that a naïve strategy of a complete enumeration in the parameter space becomes intractable. Consequently, we apply a "grid-search" using exponentially growing sequences of parameters as proposed in Hsu [29], Yong [40] and Luxburg [41]. The grid search evaluates parameter combinations and picks the one with the best accuracy on the validation set [29]. The "grid-search" promises valid and reliable results for this investigation, as different kernel functions and similar parameter combinations have to be compared to guarantee a balanced comparison.

TABLE 1

Parameter Grids for Model Selection with SVR

	RBF	Sigmoid	Linear
C	$[2^{-5},2^{-4.5},\ldots,2^{15}]$		
ε	$[2^{-10},2^{-9.5},\ldots,2^{-5}]$	$[2^{-4.75},2^{-4.5},\ldots,2^{0}]$	
γ	$[2^{-20},2^{-19},\ldots,2^{14}]$	$[2^{-13.5},2^{-13},\ldots,210]$	$[2^{-9.75},2^{-9.5},\ldots,2^{0}]$
d	[2, 3, 4, 5]		

Furthermore, we apply a shrinking technique to reduce the training time, because if most variables are finally at bounds the shrinking technique reduces the size of the working problem by considering only free variables [26, 42].

For MLPs the selection of an adequate topology is problem-dependent and influences the forecasting performance. Early investigations of Tang and Fishwick already show that the number of nodes in the topology affect the forecasting performance significantly [3]. The hidden nodes enable the NN to capture the pattern in the data and perform the non-linear mapping between input and output variables, and they correspond to the number of free parameters to model the time series [14]. Several heuristics exist to determine for the topology selection but non of them are established across arbitrary empirical datasets, as the guidelines and heuristics were developed in limited experiments [3]. Although in theory a single hidden layer is sufficient for NNs to approximate any arbitrary non-linear, fully differentiable function, most authors use only one hidden layer for forecasting purposes. However, two or more hidden layers may provide enhanced accuracy for non-differentiable time series. To estimate an accurate MLP architecture, a set of 920 candidate models for each time series were evaluated. All used parameters for the experiments with NN are shown in the following table:

TABLE 2
PARAMETER GRID FOR MODEL SELECTION WITH NN

Topology	Input nodes	13
	All hidden Nodes have the tanh activation function	Hidden Layer 1: 0,2…,30 nodes Hidden Layers1+2: Sum 0,2,…30 nodes Hidden Layers1+2+3: Sum 0,2,…30 nodes
	Output node	1 with identity function
Learning Parameter	Iterations	1000 Epochs
	Algorithm	Back-Propagation
	Learning Rate	0.1
	Cooling Rate	0,99
	Cooling Cycle	1 Epoch
Early Stopping Criteria	Error evaluated every Epoch	
	Error function	MSE
	Early Stopping if there is no improvement on the validation set after 50 Epochs	
Init Interval	]-0,88;0,88[	
Sampling	Without Replacement	
Initializations	10	
Scaling:	Linear Scaling of Min & Max into	
Target Function	f(x)=x	

s

The approach evaluates every combination of hidden layer 0,…,3 and nodes 0,…,30 in step of 2 nodes limiting the models to pyramidal topologies with an equal or smaller number of nodes like in Crone et. al. [16]

For NN and SVR all data is scaled to avoid numerical difficulties and to speed up the training process [13, 29]. Linear scaling including a headroom of 50% is used, to avoid saturation effects of observations close to the asymptotic limits of the activation functions [3, 13]. As the length of the input vector corresponds to the autocorrelation lag structure in the data, a constant lag structure of 13 inputs is chosen for all time series and SVR and NN alike [3, 43]. To guarantee an objective model selection, the SVR and NN models with the lowest MSE on the validation data set are selected.

C. *Evaluation Criteria*

Comparisons of errors across time series can be measured in different ways and the selection of an error measure is dependent upon the situation [44]. As the forecasting performance can be influenced by the evaluation criteria [8, 44], three different criteria are used in this study. These are the mean absolute error (MAE), Root mean squared error (RMSE) and mean absolute percentage error (MAPE), which are explicitly explained in [15, 44]. The MAE is an easy interpretable measure that shows the relative absolute error. The RMSE has been used frequently to draw conclusions about forecasting methods [44], as it gives much more weight to large errors than to smaller ones [44]. The unit free measure MAPE show the percentage error. But as it has evaluation problems with time series vacillate around the zero-point. For these time series the MdAPE is used, which aims at reducing the bias in favor of low forecasts, thus offering advantage over the MAPE [15, 44]. The forecasting errors are measured on the out-of-sample data test set for each *t+1* forecast on a rolling origin.

D. *Experimental Results*

Table 3 provides the results of the forecasting performance of the investigated methods and different SVR kernel functions in comparison to the statistical benchmark. All error measures provided represent the out of sample set. We evaluate the forecasting accuracy on the test set for all 12 time series patterns across the increasing noise levels of low, medium and high noise, providing mean and median errors per method and noise level to show robustness of the results, across different patterns of stationary, seasonal, trended and trend-seasonal time series, linear versus non-linear patterns and across all series.

The results indicate that although in some cases alternative error measures indicate different candidates as superior, most results are robust and consistent across error measures. Therefore we will limit our discussion to the MAE performance. For low and medium noise, RBF SVR outperforms all other methods on mean and median accuracy, followed closely by the NN who outperform SVR on median error of medium noise level. On a high noise level the NN outperform all other methods, indicating robust learning of the underlying relationship for increasing levels of noise. Over all patterns and noise levels, NN outperform all other methods significantly, followed by RBF SVR, SIG SVR, LIN SVR and the statistical benchmark methods. Poly SVR show consistently inferior forecasting accuracy across noise levels, and time series patterns.

Differentiating the analysis by time series patterns, NN outperform all other methods including SVR and statistical benchmarks on seasonal, trended and trend-seasonal patterns, but not on stationary time series. Here RBF SVR outperforms NN and other methods, which is also the method with the second best performance, often even outperforming NN on median accuracy measures.

TABLE 3

FORECASTING ERROR MEASURE FOR THE OUT-OF-SMAPLE TEST DATA SET

	MAE						RMSE						MAPE					
Time Series	NN	RBF	SIG	POLY	LIN	STAT	NN	RBF	SIG	POLY	LIN	STAT	NN	RBF	SIG	POLY	LIN	STAT
E_low	**0.77**	0.78	0.81	0.84	0.79	0.82	0.97	0.97	**0.95**	1.05	0.98	1.02	46.70	**45.40**	56.00	49.40	49.00	49.84
AS_low	0.88	0.89	1.11	1.03	1.12	**0.85**	1.12	1.12	1.40	1.26	1.40	**1.05**	0.92	0.93	1.15	1.07	1.16	**0.87**
MS_low	1.38	**1.16**	1.27	1.81	1.35	1.26	1.69	**1.44**	1.58	2.33	1.63	1.81	1.51	1.37	1.47	2.00	1.53	**1.35**
LT_low	1.28	**0.84**	1.33	11.60	0.86	0.85	1.67	1.05	1.67	13.20	1.06	**1.04**	0.57	**0.37**	0.58	5.17	0.38	0.38
ET_low	1.70	**0.92**	5.75	29.90	1.07	11.20	2.04	**1.14**	8.41	31.60	1.33	14.39	2.70	**1.54**	7.40	53.70	1.73	15.42
DT_low	**0.82**	0.85	0.98	5.24	0.95	4.79	**1.03**	1.05	1.20	5.86	1.17	5.42	**0.37**	0.38	0.44	2.33	0.42	2.13
LT_AS_low	1.18	1.34	1.14	25.60	2.36	**0.86**	1.48	1.71	1.43	32.40	2.94	**1.07**	0.53	0.60	0.51	10.00	1.08	**0.38**
LT_MS_low	1.46	1.26	1.30	3.55	1.40	**0.84**	1.81	1.60	1.62	4.73	1.79	**1.06**	0.65	0.57	0.59	1.50	0.62	**0.39**
ET_AS_low	**1.29**	2.08	1.45	84.90	25.10	35.00	**1.60**	2.76	1.79	107.0	27.70	47.13	**0.34**	0.51	0.38	21.70	6.32	8.22
ET_MS_low	6.92	9.11	16.70	59.70	**6.72**	34.10	12.70	12.10	20.38	77.80	**12.10**	50.37	2.16	3.08	6.14	20.40	**2.09**	10.00
DT_AS_low	3.87	**2.82**	2.12	7.54	2.09	3.44	4.90	**3.45**	2.59	8.81	2.55	4.02	0.85	**0.62**	0.48	1.69	0.46	0.76
DT_MS_low	2.26	**1.21**	2.34	1.81	2.48	2.02	2.91	**1.63**	2.79	2.09	3.06	2.44	1.09	**0.58**	1.14	0.84	1.21	0.90
Low Noise Mean	1.98	**1.94**	3.03	19.46	3.86	8.00	2.83	**2.50**	3.82	24.01	4.81	10.90	4.87	**4.66**	6.36	14.15	5.50	7.55
Low Noise Median	1.34	**1.19**	1.32	6.39	1.38	1.64	1.68	**1.52**	1.65	7.34	1.71	2.13	0.89	**0.61**	0.87	3.75	1.19	1.13
E_medium	3.93	**3.78**	3.90	**3.78**	**3.78**	3.81	4.79	**4.68**	4.85	**4.68**	**4.68**	4.71	115.0	87.40	101.0	87.40	**85.90**	89.54
AS_medium	**3.72**	4.06	5.67	4.24	5.69	3.84	**4.79**	5.16	7.02	5.49	7.04	4.82	**3.77**	4.21	5.79	4.20	5.83	3.96
MS_medium	**5.50**	6.79	5.89	6.86	6.29	5.95	**6.88**	8.27	7.34	8.49	7.39	7.87	6.38	8.15	6.83	8.89	7.23	6.39
LT_medium	4.61	4.46	7.87	19.00	4.60	**3.81**	5.89	5.51	9.18	22.10	5.89	**4.71**	2.06	1.98	3.43	8.43	2.07	**1.70**
ET_medium	5.44	**5.24**	6.56	30.90	5.61	14.90	7.10	**6.22**	8.06	33.00	6.75	18.38	8.48	**8.38**	9.80	54.80	6.05	20.66
DT_medium	**4.10**	4.29	4.85	5.84	4.53	9.75	**5.15**	5.30	5.93	7.31	5.59	11.20	**1.82**	1.89	2.13	2.59	2.00	4.28
LT_AS_medium	6.35	6.42	**3.51**	25.80	6.47	4.04	7.59	7.81	**4.59**	33.00	8.05	4.98	2.86	2.85	**1.59**	10.20	2.87	1.83
LT_MS_medium	6.21	6.45	6.62	26.50	6.60	**3.99**	7.47	7.78	8.20	35.50	8.19	**4.96**	2.87	2.96	3.03	10.50	3.03	**1.88**
ET_AS_medium	15.60	17.20	16.50	105.0	**5.95**	51.70	18.60	20.90	20.84	126.0	**7.33**	64.57	4.04	4.40	4.07	27.60	**1.61**	12.45
ET_MS_medium	17.00	**12.00**	14.70	109.0	32.30	53.40	23.90	**16.00**	19.82	155.0	39.00	70.99	5.34	**4.51**	5.28	39.60	10.70	16.53
DT_AS_medium	6.77	7.07	7.77	13.60	8.11	**4.45**	8.55	8.59	9.28	18.00	9.81	**5.36**	1.51	1.57	1.74	3.01	1.83	**1.00**
DT_MS_medium	6.28	**6.19**	7.95	10.20	106.0	3.91	7.88	**7.80**	9.80	12.10	111.0	4.95	**2.77**	2.82	3.64	4.58	52.10	1.80
Medium Noise Mean	7.13	**7.00**	7.65	30.06	16.33	13.63	9.05	**8.67**	9.58	38.39	18.39	17.29	13.08	**10.93**	12.36	21.82	15.10	13.50
Medium Noise Median	**5.86**	6.31	6.59	16.30	6.12	4.25	**7.29**	7.79	8.13	20.05	7.36	5.17	**3.32**	3.59	3.86	9.55	4.43	4.12
E_high	9.39	**8.50**	9.88	8.81	8.72	8.51	11.50	**10.70**	12.20	11.40	11.00	10.90	115.0	102.0	100.0	110.0	**98.50**	111.0
AS_high	9.45	13.90	12.90	14.20	13.00	**8.91**	12.10	17.10	16.10	18.10	16.10	**11.80**	10.00	16.20	14.20	15.10	14.20	**9.89**
MS_high	**8.54**	12.10	13.70	14.70	13.60	12.30	**10.70**	15.30	17.21	19.90	17.20	16.18	**11.20**	16.50	18.10	20.60	18.10	15.10
LT_high	8.76	14.70	9.26	20.50	8.95	**8.47**	11.60	18.10	11.70	23.30	11.40	**10.80**	3.87	6.66	4.13	9.12	4.02	**3.85**
ET_high	**12.90**	13.30	21.10	27.90	19.60	16.60	**16.40**	17.00	26.80	31.40	23.40	20.95	23.90	**22.30**	38.60	49.80	29.20	24.81
DT_high	**8.51**	9.03	9.72	12.40	9.79	8.62	**10.80**	11.50	12.49	15.40	12.50	11.25	**3.77**	4.00	4.29	5.41	4.30	3.87
LT_AS_high	13.20	13.30	13.30	17.00	13.30	**8.92**	16.40	16.50	16.80	23.40	16.50	**11.80**	6.00	6.05	6.03	7.33	6.07	**4.15**
LT_MS_high	13.80	15.60	95.80	29.20	14.40	**9.14**	17.06	20.40	101.0	37.90	17.70	**12.20**	6.48	7.11	45.20	12.36	6.63	**4.45**
ET_AS_high	19.30	**19.00**	23.40	135.0	19.90	47.90	26.40	**24.50**	33.67	183.0	24.40	61.21	5.19	**5.02**	5.69	33.30	5.09	11.44
ET_MS_high	26.30	28.50	**15.80**	85.30	39.10	49.50	34.30	38.60	**22.60**	118.0	49.30	67.99	8.68	10.00	**5.58**	31.70	12.70	15.12
DT_AS_high	14.20	**13.40**	15.20	22.60	15.30	13.60	17.40	**16.60**	18.65	31.00	18.70	16.36	3.15	**3.01**	3.36	5.00	3.37	3.10
DT_MS_high	11.20	13.60	14.20	14.00	14.20	**9.23**	14.74	16.30	17.77	17.20	17.70	**12.20**	5.18	6.34	6.44	6.20	6.42	**4.49**
High Noise Mean	**12.96**	14.58	21.19	33.47	15.82	16.81	**16.62**	18.55	25.58	44.17	19.66	21.97	**16.87**	17.10	20.97	25.49	17.38	17.61
High Noise Median	**12.05**	13.50	13.95	18.75	13.90	9.19	**15.57**	16.80	17.49	23.35	17.45	12.20	**6.24**	6.89	**6.24**	13.73	6.53	7.19
Mean Stationary Patterns	4.70	**4.35**	4.86	4.48	4.43	4.38	5.75	**5.45**	6.00	5.71	5.55	5.54	92.23	78.27	85.67	82.27	**77.80**	83.46
Mean Seasonal Patterns	**4.91**	6.48	6.76	7.14	6.84	5.52	**6.21**	8.07	8.44	9.26	8.46	7.26	**5.63**	7.89	7.92	8.64	8.01	6.26
Mean Trend Patterns	**5.35**	5.96	7.49	18.14	6.22	8.78	**6.85**	7.43	9.49	20.35	7.68	10.90	**5.28**	**5.28**	7.87	21.26	5.57	8.57
Mean Trend-Seasonal P.	**9.62**	9.81	14.43	43.13	17.88	18.67	12.54	**12.50**	17.42	56.83	20.99	24.65	**3.32**	3.48	5.61	13.75	6.90	5.49
Median Stationary	3.93	**3.78**	3.90	**3.78**	**3.78**	3.81	4.79	**4.68**	4.85	**4.68**	**4.68**	4.71	115.0	87.40	100.0	87.40	**85.90**	89.54
Median Seasonal	**4.61**	5.43	5.78	5.55	5.99	4.90	**5.84**	6.72	7.18	6.99	7.22	6.35	**5.08**	6.18	6.31	6.55	6.53	5.18
Median Trend	4.61	**4.46**	6.56	19.00	4.60	8.62	5.89	**5.51**	8.41	22.10	5.89	11.20	2.70	**1.98**	4.13	8.43	2.07	3.87
Median Trend-Seasonal	**6.85**	8.09	10.63	25.70	10.71	9.03	10.63	**10.35**	13.30	32.70	14.30	12.00	**2.87**	2.99	3.50	10.10	3.20	3.63
Mean Linear Patterns	5.29	6.08	5.89	12.70	5.80	**4.47**	6.66	7.53	7.32	15.78	7.25	**5.73**	25.61	22.89	24.53	26.45	**22.59**	23.12
Mean Non-linear Patterns	**8.39**	8.72	12.99	35.14	15.10	16.98	**10.92**	11.09	15.83	45.39	17.80	22.22	**4.60**	4.90	7.58	17.50	7.70	7.77
All Noise Levels Mean	**7.36**	7.84	10.62	27.66	12.00	12.81	**9.50**	9.91	12.99	35.52	14.29	16.72	11.60	**10.90**	13.23	20.49	12.66	12.89
All Noise Levels Median	**6.25**	6.44	7.20	14.45	6.54	8.49	**7.53**	7.81	8.80	19.00	8.12	10.85	**3.46**	3.54	4.21	9.56	4.16	4.22

In distinguishing between linear and nonlinear combinations of the archetypical time series patterns, NN significantly outperform all other methods on non-linear patterns, closely followed by RBF SVR. However, on linear time series patterns the statistical benchmark methods clearly outperform all methods from computational intelligence. As we evaluated only a single expert system approach in the form of ForecastPro, this may not serve as a general proof of potential problems with linear methods to forecast non-linear patterns, but rather as an indication to the sub-optimal model selection and parameterization scheme incorporated in the software, which does include models of DT but not of ET to forecast the corresponding non-linear time series patterns.

To summarize, both NN and SVR show pre-eminent performance in comparison to established statistical benchmark methods of ARIMA and Exponential Smoothing across time series patterns, noise levels and linear and non-linear patterns. The inferior performance of SVR with polynomial kernels raises caution to its use for future applications. MLPs and SVR with RBF kernels demonstrate superior performance allowing valid and reliable forecasting without ex ante identification of the underlying time series patterns and specification of the correct model form in contrast to established statistical methods. Although alternative methods may outperform NN and RBF SVR on selected time series patterns, NN and RBF always demonstrate near comparative accuracy, indicating these methods as a robust alternative if only a single method is evaluated or applied in forecasting.

IV. Conclusions

This empirical comparison analyzed the performance of competing forecasting methods of SVR with four commonly used kernel functions and NN in comparison with established statistical benchmarks. The forecasting performance was measured on 36 time series consisting of 12 basic time series patterns overlaid with three levels of noise. The results were evaluated by using three established error measures on out-of-sample accuracy.

The experiments demonstrated the superior forecasting performance of NN and SVR with RBF kernels and indicated their ability to learn an adequate model form and parameters directly from the data. While NN are well known in forecasting applications, the recently developed method of SVR provides a recent alternative to time series forecasting, offering attractive properties through the underlying statistical learning theory. However, even through SVR in theory promise global minima and therefore provide an attractive alternative to the theoretical shortcomings of NN with local minima, their forecasting performance is not superior to NN. However, this mixed performance may be the result from suboptimal hyper-parameter combinations and a biased model selection approach, which provides fertile ground for further investigation. Also, the forecasting performance of SVR appears to be highly influenced by the choice of its kernel function. The RBF kernel function performed superior on time series forecasting problems, which stands in contrast to other recent investigations of the impact of SVR parameterisation. Although this investigation does not attempt to derive any general superiority of a particular forecasting method for all potential forecasting applications and time series patterns, NN and SVR have demonstrated per-eminent performance. To say the least, their accuracy was under no circumstances significantly worse then established methods, calling for increased research activity in these areas.

For future research, further evidence on the sensitivity of NN and SVR parameters on time series forecasting accuracy is required. We seek to extend our analysis towards multiple real empirical time series, evaluating results across multiple origins and for different forecasting horizons.

References

[1] G. Zhang, "Linear and Nonlinear Time Series Forecasting with Artificial Neural Networks," vol. Doctor of Philosophy: Kent State Graduate School of Management, 1998, pp. 152.

[2] S. F. Crone, "Prediction of White Noise Time Series using Artificial Neural Networks and Asymmetric Cost Functions," presented at International Joint Conference on Neural Networks, Portland (U.S.A.), 2003.

[3] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with Artificial Neural Networks: The State of the Art," *International Journal of Forecasting*, vol. 14, pp. 35-62, 1998.

[4] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A Simulation Study of Artificial Neural Networks for Nonlinear Time - Series Forecasting," *Computers & Operations Research*, vol. 28, pp. 381-396, 2001.

[5] G. P. Zhang, "Neural Networks in Business Forecasting: An Overview," in *Neural Networks in Business Forecasting*, G. P. Zhang, Ed. Hershey (U.S.A.): Idea Group Publishing, 2004, pp. 1-22.

[6] G. P. Zhang; and M. Qi, "Neural Network Forecasting for Seasonal and Trend Time Series," *European Journal of Operation Research*, vol. 160, pp. 501 - 514, 2003.

[7] J. V. Hansen; and R. D. Nelson, "Neural Networks and Traditional Time Series Methods: A Synergistic Combination in State Economic Forecasts," *IEEE Transactions on Neural Networks*, vol. 8, pp. 863-873, 1997.

[8] K.-P. Liao; and R. Fildes, "The Accuracy of a Procedural Approach to Specifying Feedforward Neural Networks for Forecasting," *Computers & Operations Research*, vol. 32, pp. 2121-2169, 2005.

[9] S. F. Crone, "Stepwise Selection of Artificial Neural Networks Models for Time Series Prediction," University of Lancaster, Lancaster (UK) 2004.

[10] A. Zell, *Simulation neuronaler Netze*, vol. 1. Aufl. Bonn: Addison - Wesley Verlag, 1994.

[11] V. N. Vapnik, "An Overview of Statistical Learning Theory," *IEEE Transactions on Neural Networks*, vol. 10, pp. 988-1000, 1999.

[12] R. Callan, *Neuronale Netze im Klartext*. München: Pearson Studium, 2003.

[13] S. F. Crone, "Stepwise Selection of Artificial Neural Network Models for Time Series Prediction," *Journal of Intelligent Systems*, vol. 14, pp. 23, 2005.

[14] S. D. Balkin; and J. K. Ord, "Automatic Neural Network Modelling for Univariate Time Series," *International Journal of Forecasting*, vol. 16, pp. 509-515, 2000.

[15] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting Methods and Applications*, 3 ed. New York: John Wiley & Sons, 1998.

[16] S. F. Crone, H. Kausch, and D. Preßmar, "Prediction of the CATS benchmark using a Business Forecasting Approach to Multilayer Perceptron Modelling," presented at IJCNN'04, Budapest (Hungary), 2004.

[17] J. Faraway; and C. Chatfield, "Time Series Forecasting with Neural Networks: A Case Study," University of Bath, Bath (United Kingdom), Research Report 95-06 of the statistics group Juni 1995.

[18] M. Welling, "Support Vector Regression," Department of Computer Science, University of Toronto, Toronto (Kanada) 2004.

[19] J. Bi; and K. P. Bennett, "A Geometric Approach to Support Vector Regression," *Neurocomputing*, vol. 55, pp. 79-108, 2003.

[20] N. Cristianini; and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based Learning Methods*. Cambridge (United Kingdom): Cambridge University Press, 2000.

[21] M. Anthony; and N. Biggs, *Computational Learning Theory*. Cambridge (United Kingdom): Cambridge University Press, 1992.

[22] A. J. Smola; and B. Schölkopf, "A Tutorial on Support Vector Regression," Australian National University / Max-Planck-Institut für biologische Kypernetik, Canberra / Thübingen 2003.

[23] H. Yang, K. H. L. Chan, T. King, and M. R. Lyu, "Outliers Treatment in Support Vector Regression for Financial Time Series Prediction," presented at ICONIP, 2004.

[24] H. Yang, K. Huang, L. W. Chan, K. C. I. King, and R. M. Lyu, "Outliers Treatment in Support Vector Regression for Financial Time SeriesPrediction"." *Computer Science*, vol. 3316, pp. 1260-1265, 2004.

[25] K.-R. Müller, A. J. Smola, G. Rätsch, B. Schölkopf, J. Kohlmorgen, and V. Vapnik, "Predicting Time Series with Support Vector Machines," in *Advances in Kernel Methods — Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge: MIT Press, 1999, pp. 243-254.

[26] C.-C. Chang; and C.-J. Lin, "LIBSVM: a Library for Support Vector Machines," National Science Council of Taiwan, Taipei (Taiwan) 17. April 2005.

[27] S. R. Gunn, "Support Vector Machines for Classification and Regression," University of Southampton, Technical Report. Image, Speech and Intelligent Systems Group January 1998.

[28] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," in *Data Mining and Knowledge Discovery*, vol. 2, U. Fayyad, Ed. Boston (U.S.A.): Kluwer Academic Publishers, 1998, pp. 121–167.

[29] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," National Tawain University, Taipei (Taiwan) 2003.

[30] B. Funt; and W. Xiong, "Estimating Illumination Chromaticity via Support Vector Regression," presented at IS&T/SID Twelfth Color Imaging Conference, 2004.

[31] C.-H. Wu, C.-C. Wei, D.-C. Su, M.-H. Chang, and J.-M. Ho, "Travel Time Prediction with Support Vector Regression," presented at IEEE Intelligent Transportation Systems Conference, 2003.

[32] B. Schölkopf, P. Y. Simard, A. J. Smola, and V. Vapnik, "Prior Knowledge in Support Vector Kernels," in *Advances in Neural Information Processings Systems*, vol. 10, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds. Cambridge (United Kingdom): MIT Press, 1998, pp. 640-646.

[33] H.-T. Lin; and C.-J. Lin, "A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods," Technical report, Department of Computer Science and Information Engineering, National Taiwan University. March 2003.

[34] B. Schölkopf, K.-R. Müller, and A. J. Smola, "Lernen mit Kernen: Support-Vektor-Methoden zur Analyse hochdimensionaler Daten," *Informatik, Forschung und Entwicklung*, vol. 14, pp. 154–163, 1999.

[35] A. J. Smola; and B. Schölkopf, "A Tutorial on Support Vector Regression," *Statistics and Computing*, vol. 14, pp. 199-222, 2004.

[36] A. Smola, "Regression Estimation with Support Vector Learning Machines," Technische Universität München, 1996.

[37] V. Cherkassky; and Y. Ma, "Practical Selection of SVM Parameters and Noise Estimation for SVM Regression," *Neural Networks*, vol. 17, pp. 113-126, 2004.

[38] B. Schölkopf, "Support Vector Learning." Berlin: Technische Universität, 1997.

[39] S. Pietsch, "Computational Intelligence zur Absatzprognose - Eine Evaluation von Künstlichen Neuronalen Netzen und Support Vector Regression zur Zeitreihenprognose," in *Institut für Wirtschaftsinformatik*. Hamburg: Universität Hamburg, 2006.

[40] Q. Yong, Y. Jie, Y. Lixiu, and Y. Chenzhou, "An Improved Way to Make Large-Scale SVR Learning Practical," *EURASIP Journal on Applied Signal Processing*, vol. 8, pp. 1135-1141, 2004.

[41] U. v. Luxburg, O. Bousquet, and B. Schölkopf, "A Compression Approach to Support Vector Model Selection," *Journal of Machine Learning Research*, vol. 5, pp. 293-323, 2004.

[42] C.-W. Hsu; and C.-J. Lin, "A Comparison of Methods for Multiclass Support Vector Machines," *IEEE Transactions on Neural Networks*, vol. 13, pp. 415-425, 2002.

[43] S. F. Crone, S. Lessmann, and S. Pietsch, "Forecasting with Computational Intelligence - An Evaluation of Support Vector Regression and Artificial Neural Networks for Time Series Prediction," presented at IEEE World Congress on Computational Intelligence, Vancouver (Canada), 2006.

[44] J. S. Armstrong; and F. Collopy, "Error Measures for Generalizing About Forecasting Methods: Empirical Comparisons," *International Journal of Forecasting*, vol. 8, pp. 69-80, 1992.