Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

A Naïve Support Vector Regression Benchmark for the NN3 Forecasting Competition

Sven F. Crone and Swantje Pietsch

Abstract—**Support Vector Regression is one of the promising contenders in predicting the 111 time series of the NN3 Neural Forecasting Competition. As they offer substantial degrees of freedom in the modeling process, in selecting the kernel function and its parameters, cost and epsilon parameters, issues of model parameterization and model selection arise. In lack of an established methodology or comprehensive empirical evidence on their modeling, a number of heuristics and ad-hoc rules have emerged, that result in selecting different models, which show different performance. In order to determine a lower bound for Support Vector Regression accuracy in the NN3 competition, this paper seeks to compute benchmark results using a naïve methodology with a fixed parameter grid-search and exponentially increasing step sizes for radial basis function kernels, estimating 43,725 candidate models for each of the 111 time series. The naïve approach attempts to mimic many of the common mistakes in model building, providing error as a lower bound to support vector regression accuracy. The in-sample results parameters are evaluated to estimate the impact of potential shortcomings in the grid search heuristic and the interaction effects of the parameters.**

I. Introduction

Time series forecasting with computational intelligence has received increasing attention in theory and practice. However, in order to prove their efficacy in forecasting their accuracy must be evaluated against established statistical forecasting methods on empirical datasets [1, 2]. The 2007 NN3 Forecasting Competition for computational intelligence methods provides this opportunity on a dataset of 111 empirical time series and a reduced subset of 11 time series in order to establish the forecasting accuracy of computational intelligence methods on business data.

Recently, the method of Support Vector Regression (SVR) has shown promising performance in various scientific forecasting domains [3-7], offering a non-parametric, data-driven and self-adaptive method that learns linear or nonlinear functional relationships directly from training examples [2, 4]. However, recent experiments have demonstrated that despite the promise of automatically estimating optimal predictors using statistical learning theory, SVR offer substantial degrees of freedom in forecasting, requiring a data dependent selection of the kernel function and its parameters from a set of potential functions, and two metric scaled parameters to control the cost and epsilon-insensitive margin. Hence, support vector regression share common and well known problems with competing forecasting methods such as artificial neural networks, offering near endless degrees of freedom in the choice of architecture parameters to be selected based upon their performance on short and noisy time series. Due to the lack of an established methodology, a number of modeling heuristics and ad-hoc rules-of-thumb have emerged in order to guide architecture decisions in SVR modeling. As different heuristics may lead to distinct models and varying performance, they require a systematic evaluation.

To further investigate the capability of SVR in time series forecasting, SVR is applied to forecast the 111 time series of the NN3 competition. In order to determine a lower bound for SVR predictions in the competition, this study seeks to compute benchmark results using a naïve methodology of a fixed parameter grid-search with exponentially increasing step sizes. The heuristic is limited to a radial basis function kernel, using a simple preprocessing of input-data, and following a simple 'pick-the-best' approach in model selection of the candidate with the lowest cross-validation mean squared error (MSE). Considering the naïve heuristics employed, the methodology actively neglects relevant modeling guidelines in SVR modeling, such as regarding data dependent kernel selection, candidate model selection using *k*-fold cross-validation for performance prediction, adequate scaling and preprocessing of data etc. Hence we develop a naïve benchmark utilizing available computational power as a lower bound to SVR performance. The naïve grid search estimates 43,725 candidate models for each of the 111 time series, computing a total of 4,853,475 models for the NN3 datasets. In addition to providing benchmark results, we analyse the interaction of different parameter choices and interpret the results.

The paper is organised as follows. First we provide a brief introduction to ε-SVR and the relevant model parameters in forecasting, followed by a short discussion on alternative methodologies for modelling SVR. Section 3 outlines the experimental setup of data pre-processing, SVR parameter selection and the SVR candidate selection, followed by the parameter results and an analysis of their interactions.

II. Support Vector Regression

A. Method Background

The method of Support Vector Regression (SVR) is based on statistical learning theory by Vapnik [8] and estimates a function $f(\mathbf{x})$ that minimizes the forecasting error on a training data set $((\mathbf{x}_1, y_1),...,(\mathbf{x}_t, y_t)) \subseteq (\mathbf{X} \times Y)^t$ while keeping

Sven F. Crone is with the Department of Management Science, Lancaster University Management School, Lancaster LA1 4YX, United Kingdom (+44.1524.5-92991, e-mail: sven.f.crone@crone.de).

Swantje Pietsch is with the Institute of Information Systems, University of Hamburg, 20146 Hamburg, Germany; (e-mail: mailing@swantje-pietsch.de).

the functional form as flat as possible [4, 9, 10]. SVR is formulated as convex optimization problem with slack variables ξ_i,ξ_i^* to allow for model errors [11, 12]

$$minimize \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\left(\xi_i + \xi_i^*\right)$$
$$subject\ to \begin{cases} y_i - (\mathbf{w}\cdot\mathbf{x_i}) - b = \varepsilon + \xi_i \\ (\mathbf{w}\cdot\mathbf{x_i}) + b - y_i = \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (1)$$

which controls the trade-off between overfitting and model complexity through a regularization parameter $C>0$ [8].
We employ ε-SVR, using an ε-intensive loss function, that assigns an error only to those observations $\xi_i,\xi_i^* \geq 0$ outside the ε-insensitive tube [13, 14], named support vectors [15], using the loss function

$$|\xi|_\varepsilon := \begin{cases} 0 & if\ |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & else \end{cases} \quad (2)$$

To handle non-linear functional relationships in forecasting problems, the data are mapped using a kernel function ϕ into a high dimensional feature space F, where they may be solved linear regression, which corresponds to nonlinear regression in a lower dimensional input space [5].In this study the radial basis kernel function (RBF) is applied, as it is commonly used in ε-SVR using just one parameter γ that to determined the RBF width a priori [3, 4, 16]. For the RBF, the number of centres, location of the centres, the weights and the thresholds are all determined whilst training [17]. Various basic tutorials exist to provide an introduction on the background and the mathematical properties of support vector machines [17] and SVR in particular [4].

In forecasting with SVR, the input vector contains the lag structure of the time series, which results in dot products after combining them with the support vectors in the kernel function. The quadratic optimization problem is solved to determine Lagrangian multipliers α_i,α_i^* that are used to determine the weights $\upsilon_i = \alpha_i - \alpha_i^*$ [4]. The dot products are then weighted by $\upsilon_i = \alpha_i - \alpha_i^*$ to calculate the one-step-ahead prediction value together with the threshold b [4]. The forecasting process can be visualised as in figure 2.

B. Methodologies in specifying SVR parameters

SVR is a data-driven and self-adaptive methods, which is capable of approximating linear and nonlinear functional relationships from data, unlike traditional model-based statistical methods [2, 4]. For applying ε-SVR in forecasting, a number of parameters must be determined a priori by determining the Costs C, the width of the epsilon-insensitive loss function ε, the kernel function and its kernel parameters γ [18] and the number and lags of independent variables to specify the input vector as a rolling window of fixed size over a time series. Recent experiments have shown that the forecasting performance of the ε-SVR is significantly impacted by an adequate a priori selection of their parameters [19], they are considered to be semi-parametric

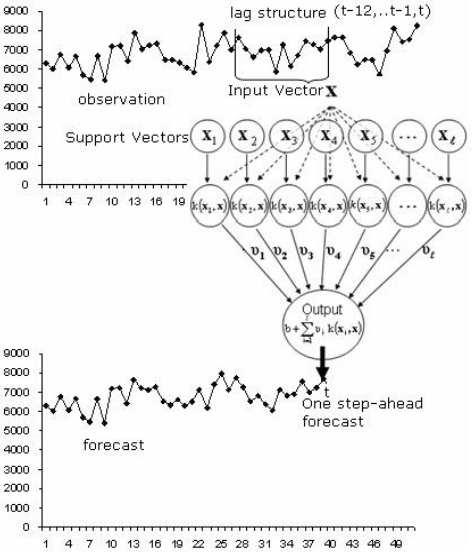

Fig. 2. Time series prediction with SVR

rather then non-parametric methods, similar to neural networks. Hence they require an expert to determine a priori model parameters that impact on the forecasting capabilities.

In order to determine the ε-SVR parameters various modelling heuristics exist. Gao, Gunn, Harris and Brown determine SVR parameters using a Bayesian framework for Gaussian SVR [20]. Chu, Keerthi and Ong follow another Bayesian approach, that combines the merits of SVR with the advantages of Bayesian methods for model adaptation [21]. Chang and Lin derived leave-one-out bounds for SVR parameters [22]. Heuristics can also lead to parameter combinations with inferior performance the even a simple parameter grid search [23], as in comparing a grid search with a Bayesian approach in Lin and Weng [24]. Momma and Bennett perform model selection by pattern search to reduce the number of parameter combinations that need to be tested [18, 25]. Kwok and Tsang [26] as well as Smola, Murata, Schölkopf and Müller [27] determine the parameter ε as a linear dependency on the noise of the training data, which requires a priori knowledge of the noise level [22]. In contrast, Cherkassky and Ma analyse the parameter interaction to limit the number of relevant parameters. The suggest that for a given ε, the value of C has only negligible effects on the generalization performance as long as C is larger than a certain threshold that can be determined from the training data [18].

A simple method to determine suitable ε-SVR parameters for each time series follows a systematic grid search over the parameter space [23, 28]. Instead of evaluating every possible parameter combination, which would be intractable for parameters of interval scale, a grid using equidistant steps in the parameter space limits the computational effort. However, different grids are applicable, using linear step sizes, exponential increasing sequences as in Hsu [23] or Luxburg [29] or logarithmic sequences as in Chang and Lin [22]. In addition, stepwise refinements of the grid size in parameter space are feasible, leading to an analytically

simple yet computationally expensive parameter selection approach. In this study we seek to explore the simple grid-based approach, using a brute-force, exhaustive enumeration of a representative parameter space.

III. Experimental Design of naïve SVR

A. Specifying SVR input vectors

The forecasting accuracy of any method depends largely on providing adequate input information to learn from. In time series forecasting this takes the form of specifying significant timed lags of the dependent variable y_{t-n} and excluding irrelevant ones, hence determining the length of the input vector. Multiple methods exist in specifying input vectors, based on simple heuristic rules, statistical autocorrelation analysis to determine the order of autoregressive (AR), integrated (I) and moving average (MA) processes or mixed ARIMA-processes of lagged realisations of the dependent variable [2] or using spectral analysis to detect multiple overlying seasonal patterns. To pursue a naïve modelling approach we select a simple heuristic decision rule based on the observation interval of the time series, using a constant lag structure of the past 12 monthly observations in a year to account for possible seasonality of the months or quarters. The same lag structure was used for all 111 time series, despite the possibility of different lag structures for different time series, the necessity to include 13 lags for seasonal integrated autoregressive processes SARIMA$(p,d,0)(P,D,0)_s$ or the approximation of MA processes of SARIMA$(0,0,q)(0,0,Q)_s$. by extending the input vector to multiples of the yearly seasonality.

B. Data Pre-Processing

The dataset contains 111 monthly time series from the complete dataset of the NN3 competition. Of these series 11 monthly time series form a reduced data subset of the competition. The NN3 time series are heterogeneous, show various seasonal and non-seasonal patterns and noise levels, and vary in length between 49 and 126 observations. The prediction objective is to forecast the next 18 observations in a multiple step ahead forecast as accurately as possible.

Data is scaled before training in order to speed up the computation process and to avoid numerical difficulties [23]. Each time series observation x_t is linearly scaled as z_t into the interval of [-0.5; 0.5], using the scaling function of

$$z_t = -0{,}5 + \frac{(x_t - x_{t\,\min})}{(x_{t\,\max} - x_{t\,\min})} \quad , \quad (3)$$

on the minimum $x_{t\,min}$ and maximum value $x_{t\,max}$ of x_t on the training and validation set and by applying headroom of 50% to avoid possible saturation effects of the nodes on instationary time series patterns in the unseen test data.

C. SVR training and parameters

We employ a simple grid search of costs C, epsilon ε and the parameter of a Gaussian kernel γ with exponentially growing sequences that cover a vast range of value combinations. Figure 2 shows an example grid with exponentially growing sequences.

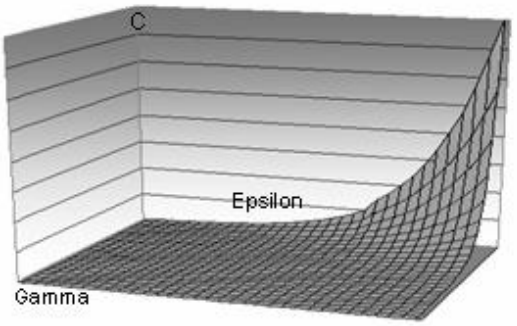

Fig. 2. A grid with exponentially growing sequences

However, employing a grid search methodology requires the setting of valid and reliable lower and upper parameter bounds that define the search space of the grid.

The regularisation parameter C determines the trade-off between the model capacity, reflected in the flatness of the approximated function, and the amount to which deviations larger then the ε-insensitive tube are tolerated [13]. A larger value for C reduces the error contribution but yields a more complex forecasting function that is more likely to overfit on the training data [18]. Hence it appears reasonable to evaluate parameters of C between a very small lower bound to create SVR-models with simple, flat functions to handle strong noise and a large upper bound to also consider SVR-models that describe more complex time series structures. In other experiments, Chang and Lin used parameter bounds between $e^{-8} \le C \le e^{8}$ [23] while Hsu and Lin used bounds between $2^{-2} \le C \le 2^{12}$ [30]. To follow an exhaustive approach, we set the lower bound to $C_l = 2^{-10}$ and the upper bound to $C_l = 2^{16}$, exceeding the parameter range of previous experiments in order to provide the capability for a sufficient trade off for the different time series patterns. This implements an exponential grid with 36 steps of $2^{0.5}$ to evaluate the parameter values of $C = [2^{-10}, 2^{-9.5}, \ldots, 2^{16}]$.

The ε-parameter controls the size of the ε-insensitive tube and hence the number of support vectors and the error contributions of observations lying outside it [9, 15]. As ε corresponds to the level of noise in a time series, large values of ε allow an approximation of the structure of the underlying functional relationship of a time series with high noise as opposed to overfitting to the noise. Chang and Lin [22] use margin values of $e^{-8} \le \varepsilon \le e^{-1}$, with Lin and Weng using similar margins [24]. We extend these search spaces and use a lower margin of 2^{-8} with an upper margin of 2^{0}. We use exponential grid steps of $2^{0.25}$, evaluating 32 parameter values of $\varepsilon = [2^{-8}, 2^{-7.75}, \ldots, 2^{0}]$ for different noise.

The kernel parameter γ defines the width of the kernel to reflect the range of the training data in feature space and therefore the ability of an SVR to adapt to the data [17, 18, 31]. Chang and Lin [22] used parameter bounds of $e^{-8} \le \gamma \le e^{8}$

and Lin and Weng [24] used bounds of $2^{-8} \leq \gamma \leq 2^{1}$. We select an exponential grid with steps of $2^{0.5}$, evaluating 30 parameter values of $\gamma=[2^{-12},2^{-11.5},\ldots,2^{0}]$ to provide feasible kernel parameters for the scaled time series data.

In total we evaluate 43,725 parameter combinations for each time series. As a grid search of this magnitude is a time intensive approach of parameter selection, we reduce the training time by applying a shrinking technique to speed up the decomposition used to solve the SVR optimization problem. It iteratively removes bounded components, so that reduced problems are solved [28, 30]. See Fan Chen and Lin for details [32]. To summarise, a total of 4,853,475 ε-SVR candidate models are computed on the 111 time series using YALE [33] and the LIBSVM libraries [28]

D. Selection of SVR candidate models

Depending on the combination of model parameters, a SVR is capable of approximating the underlying data generating process of a time series to different degrees of accuracy, permitting overfitting to the training data though the combination of sub-optimal parameters and therefore limiting its ability to generalize on unseen data [4], see figure 3.

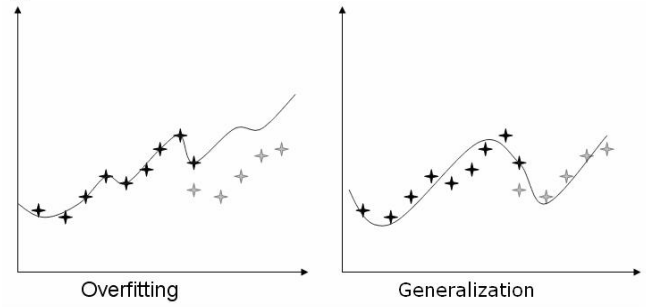

Fig.3. The difference between overfitting and generalization

Hence the selection of a robust SVR candidate for each time series requires particular attention. To select the 'best' SVR candidate model from the different parameter setups, which are 43,725 per time series in this study, each time series is split into two subsets of 65% training data and 35% validation data [34]. Considering the length of the series the validation set is selected to roughly match the undisclosed test set in length, serving as a first estimate of a quasi-out-of-sample accuracy. Each candidate ε-SVR is trained exclusively on the training data and is selected on its validation dataset.

As only a short validation dataset is used for selecting the best candidate model for that time series, overfitting on the validation set frequently occurs if the validation subset does not adequately represent the true data generating process, which cannot be expected from small data sub-samples. Multiple approaches are feasible to avoid overfitting to the validation data set in model selection and to derive an unbiased estimator on unseen data. Comprehensive methods for data sub-sampling may be considered, including k-fold cross validation using different numbers of data folds or leave-one-out cross validation [35] To adhere to the naïve approach, while avoiding the grave mistake of selection of the best candidate model on the training data itself, we compute only single cross-validation errors and select the best model on the prefixed validation set. So all SVR candidates are parameterised exclusively on the training set, while the forecasting capability of the models is evaluated on the validation set and the candidate model with the lowest validation error is selected [36, 37].

Empirical simulation experiments have proven that error measures play an important role in calibrating and refining, model selection and ex post evaluation of competing forecasting models in order to determine the competitive accuracy and rank candidate models [36, 38]. Although they should be selected with care, we apply the quadratic error measure of the root mean squared error (RMSE), weighting each error deviation by the quadratic distance using:

$$RMSE = \frac{1}{n}\sqrt{\left(\sum_{t=1}^{\ell}\left(e_t\right)^2\right)} \tag{4}$$

Using quadratic error measures emphasises the influence of large forecast errors over small ones, e.g. from outliers, and should normally be avoided in the evaluation of model performance, but according to Armstrong and Collopy [38] practitioners and academicians used the RMSE frequently to draw conclusions about forecasting methods. Furthermore, they are frequently used due to their relation with conventional least-squares-estimators and their mathematical simplicity. This is also noteworthy, as the selection criteria diverges from the final forecasting error metric in the NN3 competition, which is using a symmetric mean absolute percent error (SMAPE) [38, 39]:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{\left|y_t - \hat{y}_t\right|}{\left(y_t + \hat{y}_t\right)/2}\cdot\left(100\right) \tag{5}$$

The model with the lowest RMSE on forecasting 18 t+1 step-ahead forecasts on the validation set is selected and applied to predict the next 18 data points as multiple-step-ahead forecasts t+1, t+2, …, t+18 on the NN3 competition data sets. It is apparent, that this gives rise to another mismatch, as a method may show adequate accuracy on forecasting one step into the future, yet another set of parameters may perform better in forecasting multiple steps ahead. As this is commonly not aligned in previous studies, we comply with this malpractice in the naïve methodology, introducing further potential for misspecification errors.

No true observations for the final 18 data points of the NN3 competition are available, so no evaluation of out-of-sample accuracy may be conducted. More thorough evaluations of the naïve methodology would be feasible by splitting the available data into training, validation and test set, but these are not conducted due to the obvious sub-optimality of the naïve approach. This limits the following investigation to an analysis of correlation between training and validation data and the ranges of 'optimal' parameters.

IV. Experimental Results

To derive insight about the potential quality of the estimated SVR models, we investigate the one-step-ahead predictions for all time series on training and validation data using the SMAPE. Table 1 shows the SMAPE for the 111 NN3 time series including the 11 time series from the reduced dataset from NN100 to NN101. An analysis of the errors on training and validation sets indicates various problems of high errors, overfitting in the training process as indicated by significantly smaller errors on the training data then on the validation data set, and overfitting to the validation set, indicated by significantly smaller errors on the validation set then on the training set. The numerical analysis therefore confirms that various misspecifications may have occurred, which however cannot be attributed to a particular model choice or the uncontrollable structure dataset. To support the numerical evidence of the expected low validity and reliability of the naïve benchmark approach, we conduct a visual inspection of the highlighted time series to validate the assumptions.

The visual inspection confirms that the SVR models for time series NN009, NN010, NN019 and NN059 overfit on the training data, showing limited generalisation through significantly lower accuracy on the validation set. Time series NN40 to NN49 show extreme noise and no apparent generalisation of the underlying structure. For most of these time series the SVR model is either very flat or significantly overfitted to the noise. Time series NN022, NN037, NN043, NN054, NN063 and NN069 show a significant error increase on the validation set, which can be explained by visual apparent outliers in the validation set. Time series NN093 and NN009 show a structural break, explaining the high forecasting error. The SVR models for time series NN027, NN029, NN043 and NN045 are very flat and hence explain the limited fit to the noisy time series.

V. Conclusions

We compute a naïve heuristic, making use of most frequent mistakes in SVR modeling for time series prediction in order to establish a lower bound for support vector accuracy in the NN3 forecasting competition. The naïve heuristic evaluates an extensive grid search of 43,725 combinations of cost parameters, epsilon-parameters and kernel parameters for each time series, calculating a total of 4,853,475 ε-SVR candidate models. In aiming for a lower bound, we neglect the necessity to identify a significant input vector per time series, evaluate different scaling schemes, evaluate different kernel functions, control for overfitting in model selection form the validation data using k-fold cross-validation, conducting model selection and evaluation on a representative error metric for the ex post evaluation of the performance or the true cost of the decision, and computing and evaluating one step ahead predictors instead of multiple-step-ahead predictors as required in the final test evaluation. The naïve heuristic identifies a set of parameters for each of the 111 time series, which are subsequently used to forecast the next 18 steps into the future for unseen data.

While we hope to demonstrate the general ability of SVR to forecast linear and nonlinear time series with seasonal and non-seasonal patterns, the discussion of the naïve heuristic methodology also aims at drawing attention to the most common mistakes in SVR as well as neural network model building. Further research, systematic evaluations of forecasting accuracy on multiple empirical time series are required to establish a valid, reliable and robust methodology for automatic SVR model building. Until then, the naïve grid search heuristic may serve as a warning benchmark which pitfalls to avoid in SVR model building.

TABLE 1
SMAPE Prediction Error on Training And Validation Set or the 111 SVR Models on the NN3 Competition Dataset

Series	Train	Valid	Series	Train	Valid	Series	Train	Valid	Series	Train	Valid	Series	Train	Valid	Series	Train	Valid
NN001	7.72	8.17	NN021	6.84	15.60	NN041	13.60	11.90	NN061	2.50	2.04	NN081	3.19	4.60	NN101	0.55	0,49
NN002	7.14	9.11	NN022	7.00	25.80	NN042	12.50	12.40	NN062	7.05	7.75	NN082	0.41	0.47	NN102	3.22	2,50
NN003	7.15	10.62	NN023	8.51	10.41	NN043	25.70	32.20	NN063	2.62	2.47	NN083	1.84	2.00	NN103	18.5	12,46
NN004	6.65	7.49	NN024	16.4	17.34	NN044	39.10	39.50	NN064	1.53	1.66	NN084	1.33	1.53	NN104	5.52	5,40
NN005	7.82	5.73	NN025	58.1	47.27	NN045	29.90	27.70	NN065	4.21	4.73	NN085	3.20	11.11	NN105	0.69	0,70
NN006	5.07	8.24	NN026	24.7	32.29	NN046	23.90	31.75	NN066	1.23	1.81	NN086	1.93	1.16	NN106	2.07	3,38
NN007	14.3	16.50	NN027	17.9	17.33	NN047	21.30	32.04	NN067	6.38	4.69	NN087	3.83	2.31	NN107	1.29	1,21
NN008	14.3	16.50	NN028	8.14	9.70	NN048	13.40	14.40	NN068	1.25	2.75	NN088	3.85	4.26	NN108	8.58	9,35
NN009	2.05	9.13	NN029	30.00	20.44	NN049	50.10	33.16	NN069	8.20	10.22	NN089	2.14	1.15	NN109	3.66	2,54
NN010	7.47	14.87	NN030	3.67	7.24	NN050	7.92	12.75	NN070	2.18	1.40	NN090	1.23	0.87	NN110	11.50	12,70
NN011	10.1	10.51	NN031	58.00	56.37	NN051	1.80	1.93	NN071	4.35	4.01	NN091	0.18	0.28	NN111	3.92	4,82
NN012	7.67	9.28	NN032	14.9	12.04	NN052	1.53	1.28	NN072	2.51	1.98	NN092	0.34	0.25			
NN013	7.93	8.74	NN033	12.2	15.29	NN053	2.72	1.83	NN073	7.16	5.06	NN093	18.70	11.74			
NN014	7.70	5.65	NN034	8.82	13.49	NN054	2.72	1.83	NN074	2.10	1.67	NN094	1.45	2.26			
NN015	6.74	7.72	NN035	12.1	12.79	NN055	1.84	1.70	NN075	1.24	1.19	NN095	5.46	5.02			
NN016	5.08	6.30	NN036	8.86	12.89	NN056	0.95	1.43	NN076	3.43	4.67	NN096	15.10	19.61			
NN017	6.66	6.44	NN037	9.57	14.93	NN057	0.43	0.45	NN077	0.39	0.31	NN097	3.49	2.87			
NN018	8.65	9.92	NN038	18.1	13.98	NN058	1.05	0.92	NN078	5.54	4.10	NN098	4.60	4.16			
NN019	4.35	14.44	NN039	8.44	6.09	NN059	4.79	2.76	NN079	2.26	2.14	NN099	5.75	5.36			
NN020	12.90	17.40	NN040	9.77	9.07	NN060	1.01	1.03	NN080	0.61	0.71	NN100	2.95	4.24			

♣ High error e, with e_{valid}, $e_{train}>x$; ✧ Overfitting on the trainng set, with $e_{valid}>e_{train}$; * Overfitting on the Validation set for model selection, with $e_{valid}<e_{train}$

ACKNOWLEDGMENT

To ensure objective and true ex ante results, all computations were conducted by the second author, without access to the undisclosed NN3 test set data at any time. None the less, the paper is exempt from officially participating in the award of the NN3 competition.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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 2006 IEEE WCCI, Vancouver (Canada), 2006.

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

[9] 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.

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

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

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

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

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

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

[16] H. Yang, I. King, and L. Chan, "Non-Fixed and Asymmetrical Margin Approach to Stock Market Prediction Using Support Vector Regression," presented at 9th Conference on Neural Information Processing, Singapore (Malaysia), 2002.

[17] 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, 1998, pp. 121-167.

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

[19] S. F. Crone, S. Lessmann, and S. Pietsch, "Parameter Sensitivity of Support Vector Regression and Neural Networks for Forecasting," presented at International Conference on Data Mining, Las Vegas (U.S.A.), 2006.

[20] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown, "A probabilistic framework for SVM regression and error bar estimation," *Machine Learning*, vol. 46, pp. 71-89, 2002.

[21] W. Chu, S. S. Keerthi, and C. J. Ong, "Bayesian Support Vector Regression Using a Unified Loss function," *IEEE Transactions on Neural Networks*, vol. 15, pp. 29-44, 2002.

[22] M.-W. Chang; and C.-J. Lin, "Leave-One-Out Bounds for Support Vector Regression Model Selection," *Neural Computation*, vol. 17, pp. 1188-1222, 2005.

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

[24] C.-J. Lin; and R. C. Weng, "Simple probabilistic predictions for support vector regression," National Taiwan University, Taipei 2004.

[25] M. Momma; and K. P. Bennett, "A pattern search method for model selection of support vector regression," presented at Second SIAM International Conference of Data Mining, 2002.

[26] J. T. Kwok; and I. W. Tsang, "Linear Dependency Between epsilon and the Input Noise in epsilon -Support Vector Regression," *IEEE Transactions on Neural Networks*, ICANN 2001, pp. 405-410, 2003.

[27] A. Smola, N. Murata, B. Schölkopf, and K.-R. Müller, "Asymptotically optimal choice of epsilon-loss for support vector machines," presented at Proceeding of the International Conference on Artificial Neural Networks, 1998.

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

[29] 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.

[30] 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.

[31] C.-C. Hsu, C.-H. Wu, S.-C. Chen, and K.-L. Peng, "Dynamically Optimizing Parameters in Support Vector Regression: An Application of Electricity Load Forecasting," presented at 39th International Conference on System Sciences, 2006.

[32] R.-E. Fan, P.-H. Chen, and C.-J. Lin, "Working Set Selection Using Second Order Information for Training Support Vector Machines," *Journal of Machine Learning Research*, vol. 6, pp. 1889-1918, 2005.

[33] I. Mierswa, M. Wurst, R. Klingenberg, M. Scholz, and T. Euler, "YALE: Rapid Prototyping for Complex Data Mining Tasks," presented at Proceeding of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.

[34] P. Cunningham, "Overfitting and Diversity in Classification Ensembles based on Feature Selection," Trinity College Dublin, Dublin (Ireland), Computer Science Technical Report: TCD-CS-2000-07, 2000.

[35] D. Opitz; and R. Maclin, "Popular Ensemble Methods: An Empirical Study," *Journal of Artificial Intelligence Research*, vol. 11, pp. 169-198, 1999.

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

[37] A. Foka, "Time Series Prediction Using Evolving Polynomial Neural Networks," in *Institute of Science and Technology*. Manchester (United Kingdom): University of Manchester, 1999.

[38] 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.

[39] R. J. Hyndman; and A. B. Koehler, "Another Look at Measures of Forecast Accuracy," Monash University, Working Paper 13/05, 20 Mai 2005.