SVEN F. CRONE
DEPARTMENT OF MANAGEMENT SCIENCE
LANCASTER UNIVERSITY MANAGEMENT SCHOOL
LANCASTER LA1 4YX, UNITED KINGDOM
S.CRONE@LANCASTER.AC.UK

Stepwise Selection of Artificial Neural Networks Models for Time Series Prediction

Abstract: Various heuristic approaches have been proposed to limit design complexity and computing time in artificial neural network modelling, parameterisation and selection for time series prediction. However, no single approach demonstrates robust superiority on arbitrary datasets, causing additional decision problems and a trial-and-error approach to network modelling. To reflect this, we propose an extensive modelling approach exploiting available computational power to generate a multitude of models. This shifts the emphasis from evaluating different heuristic rules towards the valid and reliable selection of a single network architecture from a population of models, a common problem domain in forecasting competitions in general and the evaluation of hybrid systems of computational intelligence versus conventional methods. Experimental predictions are computed for the airline passenger data using variants of a multilayer perceptron trained with backpropagation to minimize a mean squared error objective function, deriving a robust selection rule for superior prediction results.

Keywords: Multilayer Perceptron, Model Selection, Extensive Enumeration, Forecasting

1 INTRODUCTION

Artificial neural networks (ANN) 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 (Tang and Fishwick 1993; Thiesing 1998; Smith and Gupta 2000; Wong, Lai et al. 2000). ANNs promise attractive features to business forecasting, being a data driven learning machine, permitting universal approximation (Hornik, Stinchcombe et al. 1989) of arbitrary linear or nonlinear

functions from examples without a priori assumptions on the model structure, outperforming conventional statistical approaches of ARIMA- or exponential smoothing-methods on selected datasets.

Despite their theoretical capabilities, NN are not an established forecasting method in business practice. Scepticism on NN persist through inconsistent research findings and pessimistic reports on their performance (Chatfield 1993; Zhang, Patuwo et al. 1998), in part due to a modelling process which depends heavily on trial-and-error (Adya and Collopy 1998). ANNs offer many degrees of freedom in the modelling process, requiring a multitude of interdependent decisions on parameter-settings to assure valid and reliable performance. Since a complete enumeration of all parameter combinations induces high computation time, various heuristic modelling approaches, empirical guidelines, rules of thumb and simple tricks have been proposed since the late 1980s, suggesting alternative approaches to determine the architecture, guide the training process and select appropriate models to minimize the objective function (Lapedes, Farber et al. 1987; de Groot and Wurtz 1991; Weigend 1992; Zhang, Patuwo et al. 1998). Unfortunately, no single heuristic has demonstrated its ability to deliver valid and reliable forecasts on arbitrary datasets as opposed to single experiments, therefore extending the decision problem of modelling ANNs instead of limiting it. Consequently, the task of modelling ANNs for a particular prediction problem is considered as much an art as a science (Chatfield 1993; Zhang, Patuwo et al. 1998).

As no heuristic has demonstrated superior performance, we exploit available computational power to propose an extensive, simultaneous enumeration of the most influential modelling parameters of network size and depth, activation function, data sampling strategy, size of the data subsets, initialisation ranges of the trainable weights, learning parameters and early stopping schemes for a sufficient amount of initialisations while successively extending the input vector while analysing the network performance. The large amount and variety of models estimated and evaluated shifts the emphasis from determining sound heuristics to limit modelling complexity towards a valid and reliable selection of a single superior model from a large population of competing, nonlinear autoregressive ANNs. Following we employ a

stepwise selection approach, effectively modelling a miniature forecasting competition motivated from the experience and publications in the domain of business forecasting (Fildes, Hibon et al. 1998; Makridakis and Hibon 2000) to derive an adequate time series prediction.

Following a brief introduction to the use of ANNs in time series prediction and their degrees of freedom in modelling, section 3 assesses the extensive modelling and design decisions in ANN application. This is followed by our experimental design and results for the proposed selection approach on the airline passenger data in section 4. Conclusions are given in section 5.

2 MODELING MULTILAYER PERCEPTRONS FOR TIME SERIES PREDICTION

Neural Networks represent a class of distinct mathematical models originally motivated by the information processing in biological neural networks, many of them applicable to forecasting tasks. The Multilayer Perceptron (MLP) MLP represents a well researched non-recurrent ANN paradigm, which offers great flexibility in forecasting through flexibility in the number of input and output variables. Forecasting with MLPs allows the prediction of a single dependent variable $\hat{y}$ or simultaneous prediction of multiple dependent variables from either lagged realisations of the predictor variable y_{t-n} itself, explanatory variables x_i of metric, ordinal or nominal scale and lagged realisations thereof, $x_{i,t-n}$, as well as complex combinations thereof. Consequently, MLPs offer large degrees of freedom towards the forecasting model design, permitting explanatory or causal forecasting through estimation of a functional relationship

$$\hat{y} = f\left(x_1, x_2, ..., x_z\right) \quad , \qquad (\mathbf{1})$$

as well as general transfer function models or simple time series prediction of the form

$$\hat{y}_{t+1} = f\left(y_t, y_{t-1}, ..., y_{t-n+1}\right) \quad , \qquad (\mathbf{2})$$

using only the past history. Following, we present a brief introduction to the degrees of freedom in modelling MLPs for time series prediction; general discussions are given in (Bishop 1995; Kaastra and Boyd 1996; Makridakis, Wheelwright et al. 1998; Haykin 1999; Reed and Marks 1999; Zell 2000).

Forecasting time series following eq. (2) with a MLP is generally based on modelling the network by analogy to an non-linear autoregressive AR(p) model (Lapedes, Farber et al. 1987; Zhang, Patuwo et al. 1998). 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 MLP. The architecture of a feed-forward MLP of arbitrary topology is displayed in figure 1.

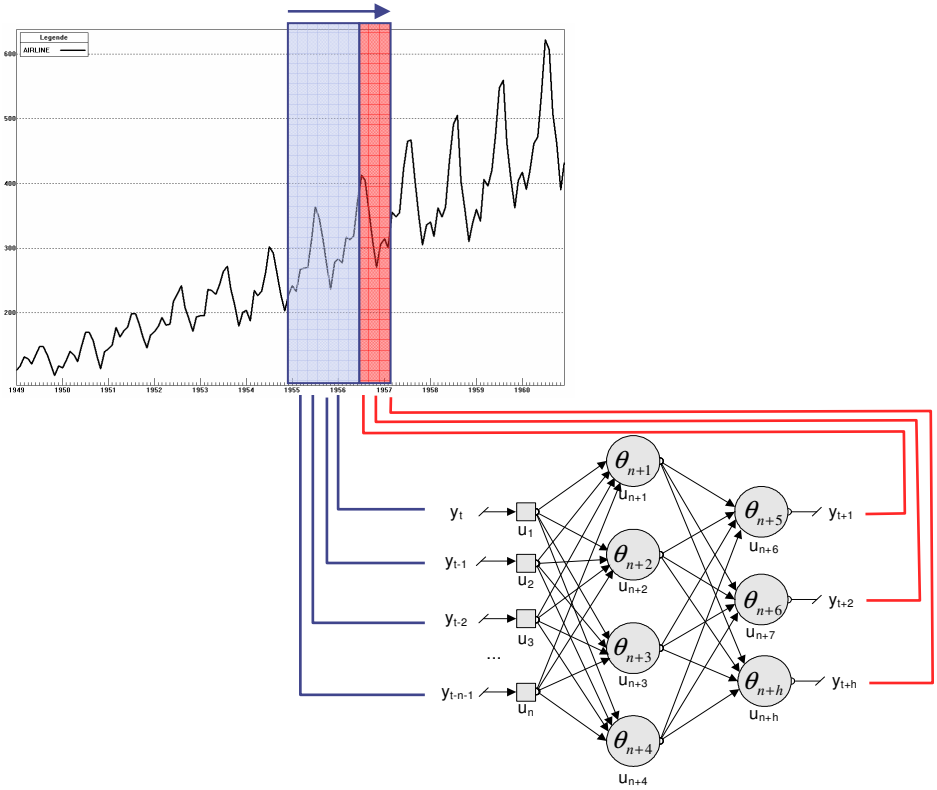

Figure 1. Autoregressive MLP application to time series forecasting with a (4-4-3)-MLP, using $n=4$ input nodes for observations in *t, t-1, t-2, t-3*, four hidden units, three output nodes for time period *t+1, t+2, t+3* and two layers of 28 trainable weights [22] The bias node is not displayed, resulting in 35 degrees of freedom.

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 network is subsequently presented with a new input vector value (Reed and Marks 1999).

The network paradigm of MLP offers extensive degrees of freedom in modeling for prediction tasks. Structuring the degrees of freedom (Alex 1998; Stahlbock 2002), each expert must decide upon the selection and sampling of datasets D, the degrees of data preprocessing P, the static architectural properties A, the signal processing within nodes U and the learning algorithm L in order to achieve the design goal, characterized through the objective function or error function O, calling for decisions upon $MLP=[P, A, L, U, D, O]$.

In data preprocessing, decisions upon correction of observations C, normalization N and scaling S must be made: $P=[C, N, S]$. The architectural properties or topology of the net is primarily determined through the size of the input vector N^I corresponding to the number of input nodes and the length of the sliding window, the size N^S and depth N^L of the hidden layers through the number of layers and number of nodes in each hidden layer, and the length of the output vector determined through the number nodes N^O in the output layer. In addition, the architecture is determined through the connectivity of the weight matrix K (fully or sparsely connected, shortcut connections etc.) and the activation strategy T (feedforward or with feedback), leading to $A=[N^I, N^S, N^L, N^O, K, T]$. The signal processing within nodes, is determined by input function F^I (weighted sum, product, distance measures etc.), activation function F^A (tanh, logistic, sin etc. with offsets, limits etc.) and output function F^O (linear, winner takes all, softmax etc.), leading to $U=[F^I, F^A, F^O]$. Decisions concerning the learning algorithm encompass the choice of learning algorithm G (backpropagation, one of its derivatives, higher order methods or heuristics etc.), the complete vector of learning parameters for each individual layer L and each different phase T of the learning process, $P^{T,L}$, the procedure I^P and number of initializations for each network I^N and the choice of the stopping method for the selection of the best network solution B. In addition, the objective of the training process must be specified through the objective or function O, although often neglected in MLP theory and practice

(Crone 2002). Consequently, the specification requires decisions upon $MLP=[[C, N, S],[N^I, N^S, N^L, K, T], [F^I, F^A, F^O], [G, P^T, I^P, I^N, B], D, O]$, with each parameter decision interacting with single or multiple other parameter recommendations. Consequently, we propose an extensive modelling approach based upon an enumeration of the relevant alternatives and a model selection procedure as is common practice in business forecasting competitions (Fildes and Ord 2002). In what follows, we discuss the modelling setup and selection process on the airline data.

3 EXAUSTIVE MODELLING APPROACH FOR MULTILAYER PERCEPTRON APPLICATION

3.1 Data Analysis

We select the well known time series of monthly totals of airline passenger data, first described by Brown (Brown 1959) and later extended by Box and Jenkins (Box and Jenkins 1970). Figure 2 gives an overview.

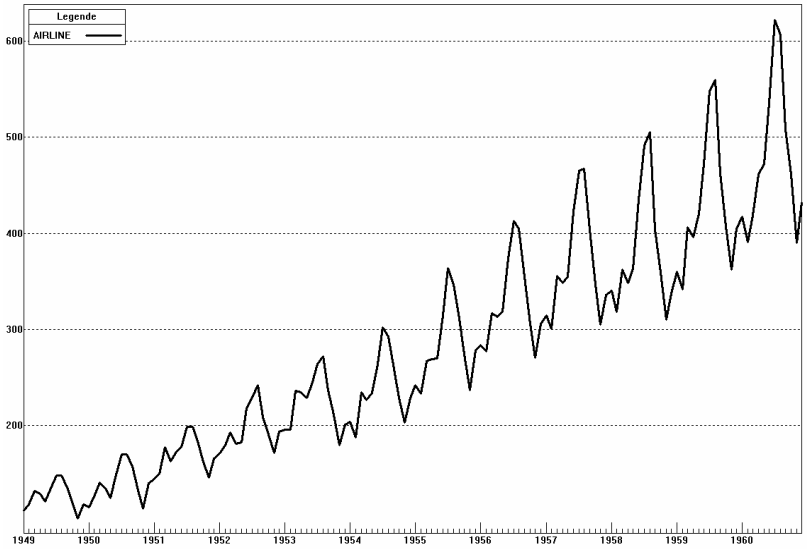

Figure 2. Monthly airline passengers in thousands

The dataset has been repeatedly analyzed (Tang and Fishwick 1993; Faraway and Chatfield 1995; Faraway and Chatfield 1998; Faraway and Chatfield 1998) and may serve as a benchmark in MLP forecasting. The data consist of 12 years of monthly values from 1949 to 1960, leading to 132 observations. The time series requires no data cleansing regarding structural breaks, correction of outliers, pulses or level shifts due to temporal external shocks etc. As the analysis of the linear autocorrelations in the lag structure reveals strong non-stationarity and seasonality in the time series, we analyze the partial linear autocorrelation coefficients of the integrated time series, as shown in figure 3.

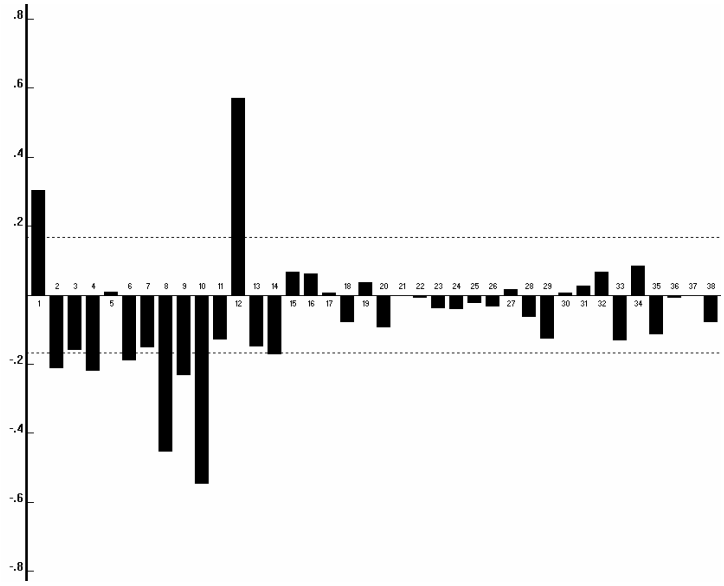

Figure 3. Partial **a**utocorrelation function of first order integrated time series of airline passenger data

We identify significant partial linear autocorrelations at lags t-1, 2, 4, 6, 8, 9, 10, 12, 14 of the integrated time series, with the most dominant autocorrelations at t-1, 8, 10, 12. These may serve as a starting point in MLP modeling to determine non-linear lag structures, following a stepwise extension of the input vector from [t-1, 8, 10, 12] to [t-1,…, 14] for input node selection.

3.2 Data Preprocessing

Subsequent training, MLPs requires adequate preprocessing of data through scaling and differencing of input and output vectors. Although linear autocorrelation analysis suggests that first order differencing of the time series may assist the neural network in learning the relevant patterns (Lapedes, Farber et al. 1987; Srinivasan, Liew et al. 1994), we analyze the original un-integrated time series, despite an ongoing discussion on the necessity of integrating time series data for nonlinear NN models, especially considering the reconstruction of multiplicative, nonlinear effects such as seasonality in full scope (Hill, O'Connor et al. 1996; Remus and O'Connor 2001) versus the reduction of noise to highlight linear lag structures.

Subsequent to transformation, we normalize the original data to avoid computational problems, to meet algorithm requirements and to facilitate network learning through speeding up of the training process (Azoff 1994). As various scaling methods allow linear equidistant or statistical scaling in arbitrary intervals (Weigend 1992; Zhang, Patuwo et al. 1998), we select a simple normalisation of

$$y_t' = \frac{y_t}{y_{\max} + h} \quad , \qquad (\mathbf{3})$$

to scale the data according to the natural origin of sales data (Lachtermacher and Fuller 1987). In accordance with the selected activation function we scale input and output data from [114000; 622000] to the interval of [0; 1] including app. 50% headroom, *h=378000,* to avoid saturation effects of observations close to the asymptotic limits of the functions (Zhang, Patuwo et al. 1998) and to account for trends in the test data outside the scale of training and validation data. To assure homogeneous scaling in all data sets we apply an external along channel normalization of the time series based upon the minimum and maximum values of training and validation data (Azoff 1994).

3.3 Architecture Selection and Node Processing

Architecture selection of a MLP offers some of the widest degrees of freedom in the modeling process (Zhang, Patuwo et al. 1998). The number of input nodes corresponds to number of lagged observations in the input vector to discover the underlying pattern in the time series for future forecasts. While too few nodes leave out relevant, exploitable information of the linear and nonlinear autoregressive lag-structure of the time series, too many nodes add uncorrelated noise obscuring input patterns. In want of a superior heuristic to determine the optimum input vector for an arbitrary dataset we follow an approach by Lattermacher and Fuller (Lachtermacher and Fuller 1987), exploiting information derived from the preliminary analysis of linear autoregressive (AR) components to select appropriate time lags for the input vector as in ARMA and ARIMA modelling (Box and Jenkins 1970; Makridakis, Wheelwright et al. 1998), consequently determining the input nodes. As a result, we model nonlinear AR-ANNs, distinct from a conventional Box-Jenkins ARIMA approach due to nonlinearities in the AR-terms and omitting MA-terms (Fildes and Makridakis 1988). We commence model building with the linear AR-lags determined in the data analysis and successively extend the input vector by including less relevant autocorrelation-lags until all past observations from lags *t-1,…,t-14* are used.

While the number of input nodes is pre-determined through variable selection and data structure, the number of output nodes N^O in the output layer is determined by the forecasting horizon of the time series forecasting problem. For a *t+n* prediction of *n* steps ahead, with *n>1,* two approaches are feasible. For forecasting horizons *t+n*, with *n>1*, we may model an MLP using one output node to model iterative *t+1* step ahead forecasts, successively using predictions as inputs for subsequent forecasting horizons as in conventional ARIMA prediction. Alternatively, we may forecast *t+1,…,t+n* values directly using a multiple-step-ahead forecasting architecture of *n* output units as proposed by (Zhang, Patuwo et al. 1998).

We have chosen to limit our evaluation to a t+1 one step-ahead forecast due to the nature of the prediction objective.

The number of hidden layers and corresponding hidden nodes in each layer is determined using an extensive enumeration, evaluating every combination of $l=1,\ldots,3$ hidden layers and a maximum of $n=1,\ldots,18$ hidden nodes, applying a step size of 2 nodes and limiting the structure of multi-layered ANNs to equal sized successive layers, to limit modeling complexity with regard to the scarce data.

Information processing functions within nodes are set homogeneously based upon experience and empirical evidence. We considered the summation as the input function, a linear output function and the hyperbolic tangent (tanh) as a nonlinear activation function in all hidden and output nodes, due to advantages in error propagating behavior (Zell 2000). The approach could be extended, however, to include alternative activation functions. The bias in each node is modeled as an 'on-node' outputting a constant of $o_0=1$ connected to all nodes in all hidden and output layers with trainable weights. Generally, we consider only fully connected feed forward architectures - no recurrent, sparsely connected networks or networks with shortcut connections are evaluated.

3.4 Training Process

Hetero-associative training of a MLP is the task of adjusting the weights of the links w_{ij} between units j and adjusting their thresholds to minimize the error δ_j between the actual and desired system behaviour (Rumelhart, McClelland et al. 1986) using various training algorithms for supervised online-training. We apply a simple derivative of the standard backpropagation gradient descent algorithm, applying a stepwise reduction of learning rate without momentum term to assure robust minimization of the objective function. We minimize a standard objective function of mean squared error (MSE), despite its theoretical and practical limitations (Reed and Marks 1999; Crone 2002) and the importance of selecting appropriate

error metrics (Zellner 1986; Fildes and Makridakis 1988), modelling the conditional distribution of the output variables similar to statistical regression problems.

Within the learning process, we evaluate various combinations of different learning rates, $\eta = \{0.85; 0.4\}$, and cooling rates for their stepwise reduction of $\varpi = \{0.95; 0.98\}$ per epoch, deriving a variety of alternative learning schemes. Each network topology is trained for up to 150000 iterations with the weight configuration causing the lowest MSE on the validation set saved for future use. To limit computation time, we apply an early stopping paradigm, evaluating the relative reduction of the network error in percent after every epoch to a 0.001% threshold for 13000 iterations. To account for random initialisation of the connection weights, we initialise each MLP 15 times prior to training. No additional heuristic pruning or growing algorithms are applied.

Due to changing input vector sizes the overall number of patterns in the dataset may vary. We divide the dataset into three distinct sub-samples of a training set to parameterize the weights, a validation set to guide early stopping and prevent overfitting and a test set to evaluate generalization on a hold-out set. Various subsample ratios are evaluated, leaving the test set constant through all experiments.

3.5 Model Evaluation and Selection

Following the training of various architectures a single model must be selected from all generated models for use in the final prediction. In accordance with the objective function and final evaluation criteria of MSE, the model with the lowest MSE on the validation data set is routinely selected. However, this simple 'pick-the-best'-approach does not guarantee robust selection of a superior model (Swanson and White 1997; Qi and Zhang 2001). Due to severe inconsistencies between performance on the validation set and the out-of-sample performance on the test set in our experiment an alternative approach must be followed. In consequence of the limited publications on valid and reliable ANN model selection

we propose a stepwise selection approach, selecting the model with the lowest validation error within the superior sub-group of model variants with regard to sampling strategy, initialisation interval, learning parameters etc. over all experiments. This approach exceeds the conventional pick-the-best approach of selecting the MLP architecture with the lowest validation error, which neglects low correlations between validation and test errors, error variance, robustness of parameterization and the probability of selecting a network model with high generalisation ability within its different initialisations. Our motivation for this approach stems from experiences in the domain of business forecasting competitions (Fildes, Hibon et al. 1998; Makridakis and Hibon 2000; Fildes and Ord 2002) and is exemplified and evaluated in the following empirical experiment.

4 SIMULATION EXPERIMENT

4.1 Experiment Structure

We evaluate a total of 14400 MLPs as variations of selected modelling parameters. To automate this extensive experiment we apply a prototype MLP simulator developed within our research group for extensive or complete enumeration in MLP time series competitions. Each of the 36 architectures is initialised 15 times to account for randomized starting weights. In combination, we evaluate 4 variants of initialisation ranges, 3 variants of data splitting between training and validation set and 2 variants of backpropagation learning schemes. The networks were trained applying early stopping in 13716 cases if no error decrease of 0.01% in 300 epochs of 96 iterations took place, leading to a median of 63960 iterations or 666 epochs, with the final network parameters saved after a median of 24960 iterations or 260 epochs.

Total computation time was 19 hours on a Pentium IV, 2400 MHz, 1GB RAM, with an average time of 5 seconds for parameterization of the network and saving all training errors, results and parameters to

completely reevaluate the experiment at any time. To support our exhaustive evaluation approach based upon available computational power, it should be noted that saving the experiment data exceeded the time of actual experimental calculations per network in the total computation time.

4.2 Experimental Results

We analyze the results of all 14400 experiments regarding their performance on all data sets. A selection of results ranked by validation error presented in Table 1.

Tab. 1. Errors on all data sets by MLP topology ranked by validation error

Rank by validation error	Data Set Errors			MLP Topology (Internal comp. ID)
	Training	Validation	Test	
overall lowest	0,009207	0,011455	0,017760	
overall highest	0,155513	0,146016	0,398628	
1st	0,010850	0,011455	0,043413	39 (3579)
2nd	0,009732	0,012093	0,023367	10 (5873)
…	…	…	…	…
25th	0,009632	0,013650	0,025886	8 (919)
…	…	…	…	…
14400th	0,014504	0,146016	0,398628	33 (12226)

According to early stopping we should select the MLP architecture with the lowest validation error for future applications. However, already the MLP ranked 2nd shows a significantly decreased test error, questioning the validity and reliability of the selection rule of lowest overall validation error. The ambiguity of the selection criteria becomes evident in plotting training, validation and test error in Figure 4.

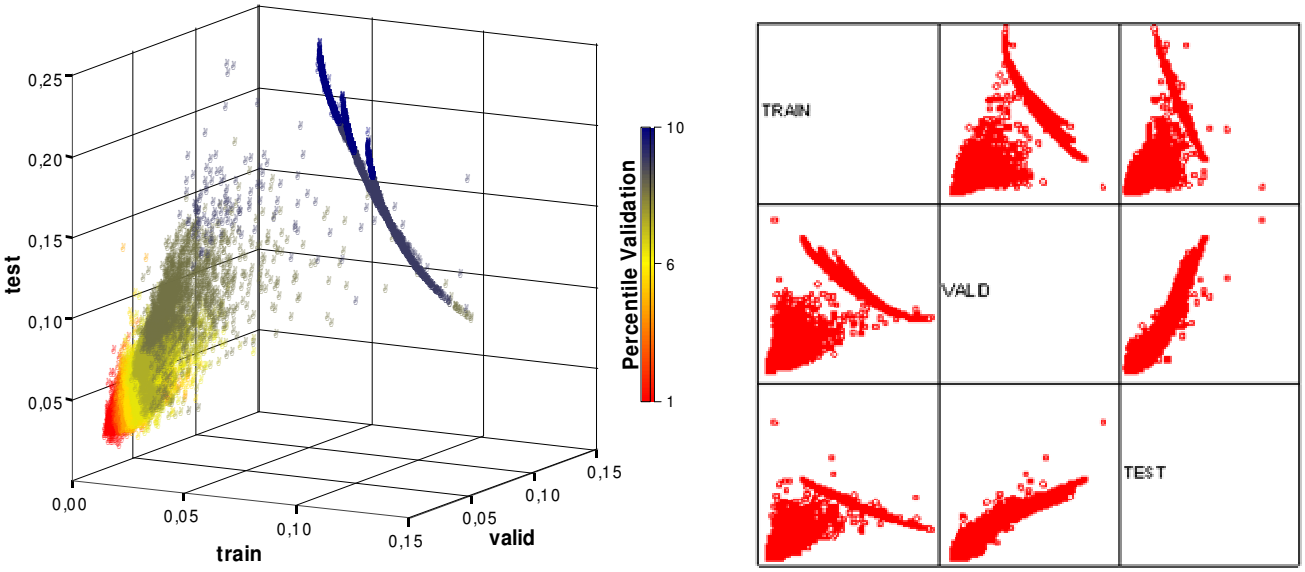

Figure 4. Correlation Plot of training, validation and test er[illegible]

Although displaying a general tendency of correlation between validation and test error, a large variance exists for low validation errors. This is supported by decreasing, significant correlation coefficients in an evaluation of all 14400, the top 1000 and the top 100 MLPs ranked by validation error in Table 2.

Tab. 2. Correlation coefficients of datasets and data ranges

	Correlation between datasets		
Data included	Train - Validate	Validate - Test	Train - Test
14400 MLPs	0,7786**	0,9750**	0,7686**
top 1000 MLPs	0,2652**	0,0917**	0,4204**
top 100 MLPs	0,2067**	0,1276**	0,4004**

The ability of generalisation through positive correlations between validation and test error within all MLPs does not apply for a stratified subsample of the top performing MLPs. For further analysis, we visualise the top percentile in an s-diagram in Figure 5, sorting MLPs by validation error to graph the ranked training, validation and test-error to show performance, variance and correlation through symmetric development of increasing error values in all sets. While the validation error must follow a

steady upwards trend as the ranking criteria, we detect no similar pattern from the test-error, indicating limited correlation between the selection criteria of a minimum validation and test error. In addition, the graph highlights significant variance within the top percentile of ranked validation error, questioning the selection criteria of validation error further.

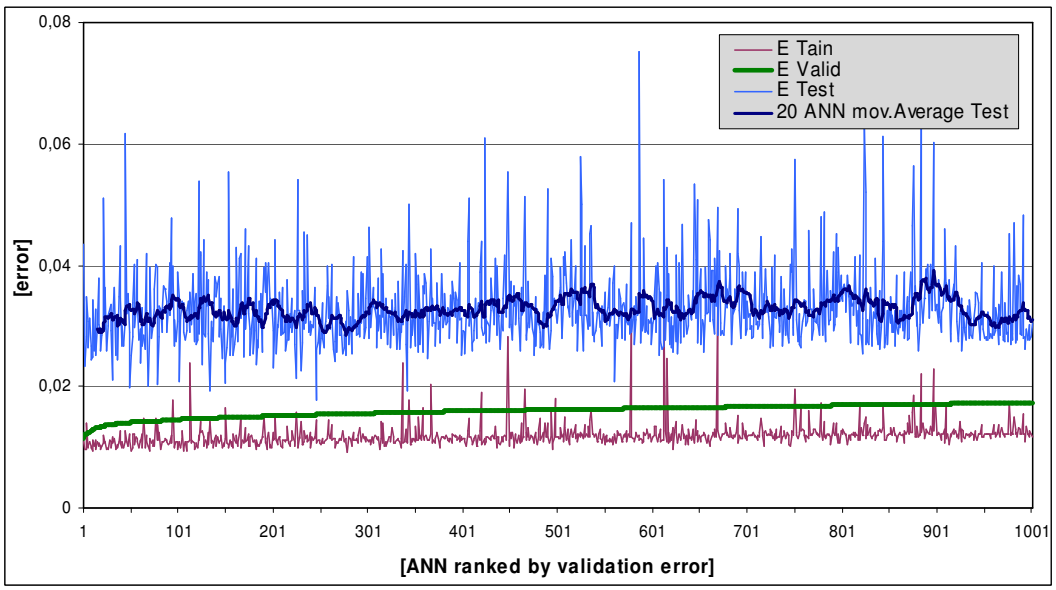

Figure 5. S-Diagram of top percentile of MLPs ranked by validation error

Consequently, we have to extend our selection criteria in order to determine a network architecture capable of robust generalization from just using a single minimum validation error. In order to limit the variance in the test error we incorporate additional selection rules in a stepwise model selection process, aiming for a valid and reliable, robust model selection.

4.3 Experimental Model Selection

In order to structure the MLP selection process to allow robust selection of a single MLP architecture, we evaluate the performance of each individual modelling variant within its group, in order to identify variants with superior performance through lower errors or lower variance of errors over all experiments.

If dominant variants are identified, we combine each chosen variant and select the MLP with the lowest validation error within the group of the dominant variant-combination. In order to analyse the sensitivity of each variant, we evaluate all groups of parameter variations for minimum, maximum and mean error as well as variance for each training, validation and test-dataset to determine a superior combination.

First, we evaluate four different initialisation ranges of network weights {[-0.01; 0.01], [-0.33, 0.33], [-0.66, 0.66], [-1, 1]} of 3510 MLPs, each for their impact on performance. Figure 6 reveals the superiority of the Init3 [-0.66, 0.66] initialisation through complete dominance of all other initialisations through the lowest minimum, mean, maximum and variance of error within training and validation set.

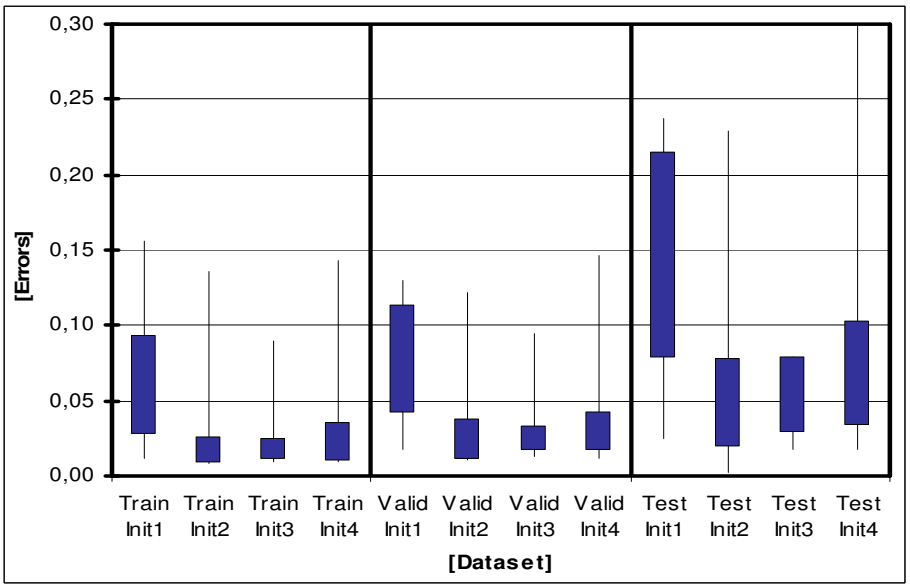

Figure 6. Box plot of minimum, maximum and mean plus and minus one standard deviation of errors for variants of initialisation range

The impact of different data samplings was analysed assigning different partitions of data to training and validation sets, while leaving the test set size constant at 20%, equalling two years of observations. We evaluate three variants {[40%, 40%, 20%], [50%, 30%, 20%], [60%, 20%, 20%]], with the results of mean errors permitting the analysis of different sample sizes across the dissimilar dataset experiments in Figure . The test errors highlight the dominance of a [60%, 20%] distribution of training to validation data

in dataset sample 3, offering the most information for parameterisation through a larger training set. The next step analyses the impact of different backpropagation learning schemes through combinations of parameter values of learning rates η=[0.85; 0.4] and step-size reductions of ω=[0.95; 0.98]. Each network was trained for a maximum of 150000 iterations, with an early stopping criteria evaluating the validation error after each epoch and stopping if the error did not decrease by 0.001% in 13000 iterations. The variants of the learning parameters demonstrated only limited impact on the results of mean error and variance, but with lower maximum values in dominant learning scheme 3, also displayed in Figure 7.

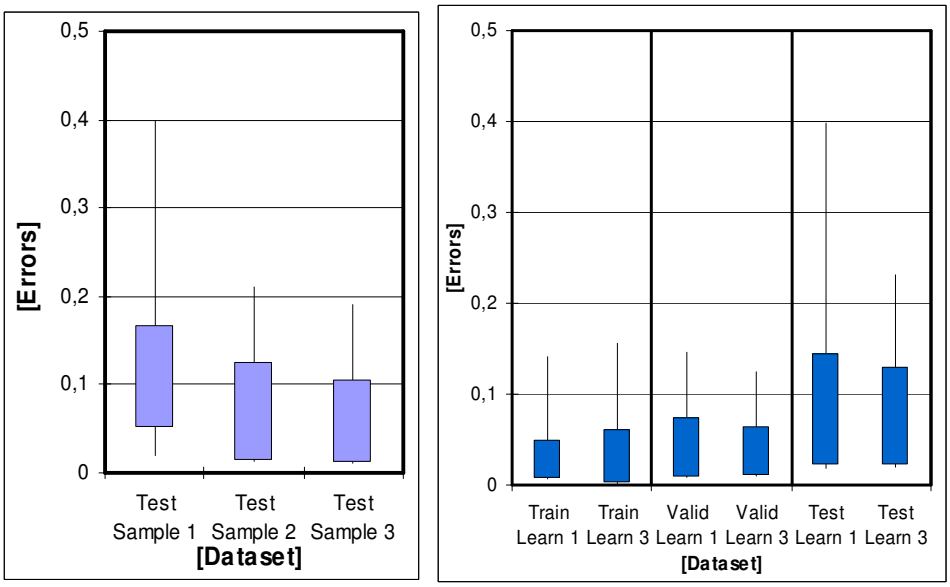

Figure 7. Box plot of minimum, maximum and mean plus and minus one standard deviation of errors for variants of data sampling (left) and learning parameters for backpropagation (right)

Finally, we analyse the topologies of 10 single, 18 double and 8 three-hidden-layered network architectures regarding their mean errors on training, validation and test set over all 360 experiments each to identify superior architectures for the dataset. The results in Figure 8 show a superior group of single layer architectures with 14 (MLP8), 16 (MLP9) and 18 (MLP10) hidden nodes, displaying low validation and test errors and correlation. Generally, one-layered MLPs outperformed multi-layered architectures in the experiment.

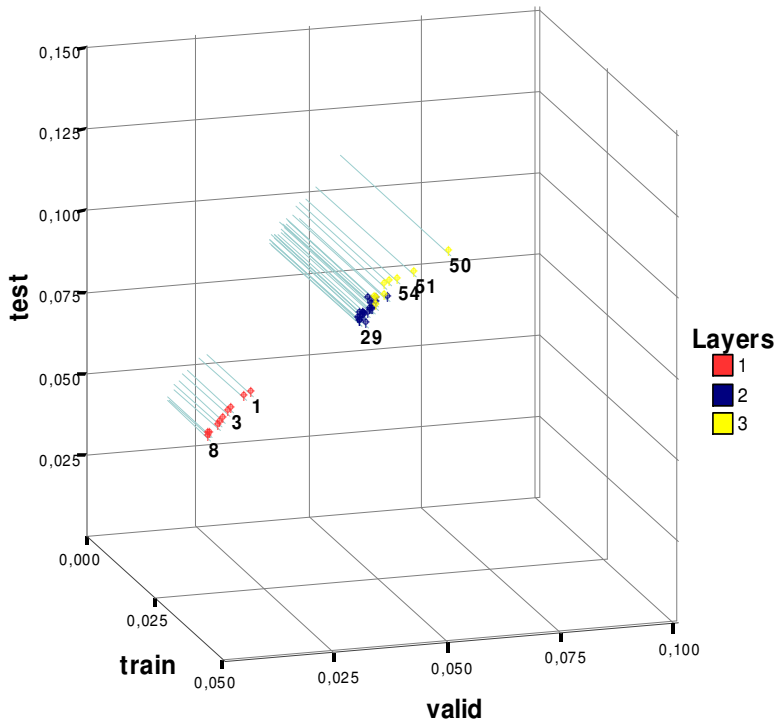

Figure 8. Mean training, validation and test errors by MLP topology over all 360 variants each

Combining all evaluated modelling variants, we select the MLP with the lowest validation error within the combined group. The ranked errors of the selected subgroup for all 15 initializations are displayed in Figure 9.

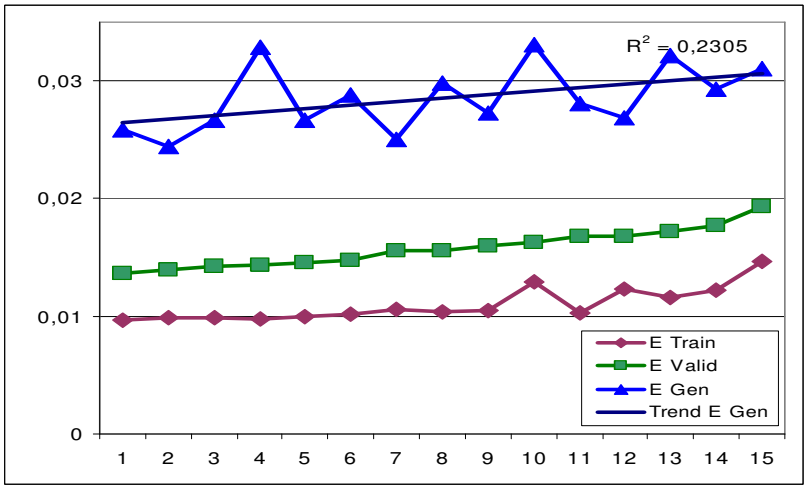

Figure 9. S-diagram of training, validation and test error over 15 initializations of selected MLP8 architecture ranked by validation error

The MLP architectures show a consistent error development within the subgroup, supported by positive correlations of 0,8778 between training and validation, 0,4475 between validation and test and 0,4762 between training and test set. Consequently, the selection of the MLP with the lowest validation error may be considered robust, although not optimal.

Of all evaluated the architectures, we selected MLP 919, a 14-14-1 architecture of variant 8 for it lowest validation error. The MLP was trained for 150000 iterations, without reaching the early stopping criteria. As determined by the selected variants, the MLP was trained using a starting learning rate of 0.85 and decreasing by 0.98 every epoch for 50000 iterations, followed by a starting learning rate of 0.4 and decreasing by 0.99 for 100.000 iterations, an initialization range of [-0.66, 0.66] and a split of the dataset of [60%, 20%, 20%], computing errors of [0.009632; 0.001365; 0.025886] on training, validation and test dataset. The network's t+1 output are shown in Figure10.

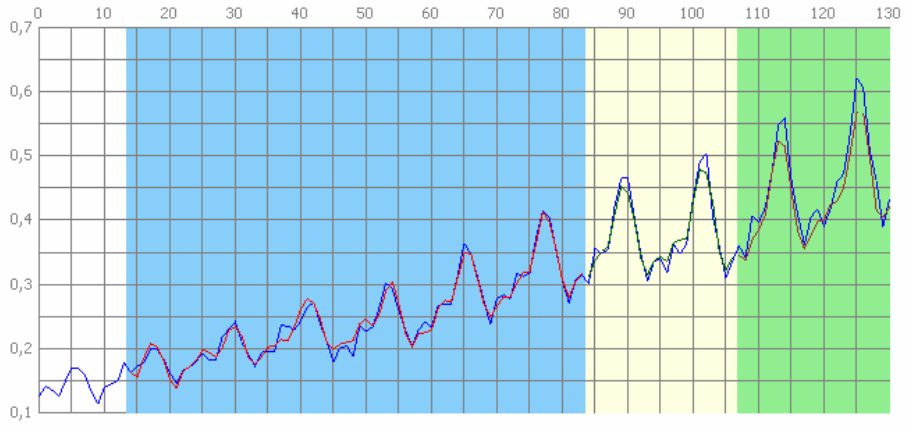

Figure 10. Time series and predictions of MLP 919 on training, validation and test data set

However, the MLP selected does not represent the MLP with lowest test error used as the routine measure of generalization ability, as seen in Tab. . In fact, the MLP is ranked 9th on training, 25th on validation and 102nd on generalization error (see Table 1), and with single realizations of the same architecture in

different variants achieving lower ranks. Therefore the selection strategy does achieve a “pick-best”-solution but aims to assure a robust selection process to derive valid and reliable results.

5 CONCLUSION

We proposed a complete or extensive enumeration of MLPs, avoiding problems in selecting acceptable heuristics and trial-and-error modelling approaches. However, the selection of a single MLP architecture based on naïve selection rules such as the lowest error on the validation dataset may lead to inconsistent, invalid and unreliable results. The spurious selection of superior models based upon a naïve selection criteria severely questions the results of common ANN practice, ensemble methods as well as the comparison of single or hybrid methods from computational intelligence with conventional approaches. In this paper, we propose a more robust selection approach, evaluating a variety of modelling variants and limiting the MLP selection to the combined group of dominant variants. However, the approach introduces yet another alternative heuristic into the neural network modelling process, allowing no direct exploitation of the available computational power for an automatic ANN modelling process free of heuristics.

Due to software limitations in the prototype simulator, we had to limit our experimental design, achieving only an extensive enumeration of the most relevant parameters. Further extensions of our experiments will incorporate additional modelling degrees of freedom, e.g. different activation functions in hidden and output layer, shortcut connections, different learning paradigms etc. and heuristics to analyse the validity of the selection criteria in more detail. In addition, we need to extend the simultaneous enumeration of all possible variants to a variety of time series from benchmark datasets and evaluating additional error measures and information criteria by Akaike and Schwartz etc. regarding their cross correlations, combinations of error measures and different early-stopping regimes based on mixed errors on validation and training-set in order to extend our results towards a general selection approach.

In general, the challenge of valid and reliable model selection underlies all empirical competitions in forecasting, data mining etc. and results in reoccurring discussions on the actual performance of competing methods. Therefore interdisciplinary research directed to solve the model selection problem within the ANN domain may also contribute to the valid selection of appropriate methods in real world applications, and consequently a more objective evaluation of artificial neural networks, establishing them as a valid, reliable and precise method for forecasting applications.

6 REFERENCES

Adya, M. and F. Collopy 1998. "How effective are neural networks at forecasting and prediction? A review and evaluation." *Journal of Forecasting* **17**(5-6): 481-495.

Alex, B. 1998. *Künstliche neuronale Netze in Management-Informationssystemen : Grundlagen und Einsatzmöglichkeiten*. Wiesbaden, Gabler.

Azoff, E. M. 1994. *Neural network time series forecasting of financial markets*. Chichester ; New York, Wiley.

Bishop, C. M. 1995. *Neural networks for pattern recognition*. Oxford, Clarendon Press.

Box, G. E. P. and G. M. Jenkins 1970. *Time series analysis; forecasting and control*. San Francisco,, Holden-Day.

Brown, R. G. 1959. *Statistical forecasting for inventory control*. New York [u.a.], McGraw-Hill.

Chatfield, C. 1993. "Neural Networks - Forecasting Breakthrough or Passing Fad." *International Journal of Forecasting* **9**(1): 1-3.

Crone, S. F. 2002. Training Artificial Neural Networks using Asymmetric Cost Functions. *Computational Intelligence for the E-Age*. L. Wang, J. C.Rajapakse, K.Fukushima and X. Y. S.-Y.Lee. Singapore, IEEE**:** pp. 2374-2380.

de Groot, C. and D. Wurtz 1991. "Analysis of univariate time series with connectionist nets: A case study of two classical examples." *Neurocomputing* **3**(4): 177-192.

Faraway, J. and C. Chatfield 1995. Time series forecasting with neural networks: A case study. Research report 95-06 of the statistics group. University of Bath**:** 1-21.

Faraway, J. and C. Chatfield 1998. "Time series forecasting with neural networks: A comparative study using the airline data." *Journal of the Royal Statistical Society Series C-Applied Statistics* **47**: 231-250.

Faraway, J. and C. Chatfield 1998. "Time series forecasting with neural networks: A comparative study using the airline data." *Applied statistics* **47**(2): 20.

Fildes, R., M. Hibon, et al. 1998. "Generalising about univariate forecasting methods: further empirical evidence." *International Journal of Forecasting* **14**(3): 339-358.

Fildes, R. and S. Makridakis 1988. "Forecasting and loss functions." *International Journal of Forecasting* **4**(4): 545-550.

Fildes, R. and Ord 2002. Forecasting competitions: their role in improving forecasting practice and research. *A companion to economic forecasting*. Malden, Mass. [u.a.], Blackwell.

Haykin, S. S. 1999. *Neural networks : a comprehensive foundation*. Upper Saddle River, N.J., Prentice Hall.

Hill, T., M. O'Connor, et al. 1996. "Neural network models for time series forecasts." *Management Science* **42**(7): 1082-1092.

Hornik, K., M. Stinchcombe, et al. 1989. "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks* **2**(5): 359 - 366.

Kaastra, I. and M. Boyd 1996. "Designing a neural network for forecasting financial and economic time series." *Neurocomputing* **10**(3): 215-236.

Lachtermacher, G. and J. D. Fuller 1987. "Backpropagation in time series forecasting." *Journal of Forecasting*(14): 381-393.

Lapedes, A., R. Farber, et al. 1987. Nonlinear signal processing using neural networks prediction and system modelling. Los Alamos, NM, Los Alamos National Laboratory.

Makridakis, S. and M. Hibon 2000. "The M3-Competition: results, conclusions and implications." *International Journal of Forecasting* **16**: 451-476.

Makridakis, S. G., S. C. Wheelwright, et al. 1998. *Forecasting : methods and applications*. New York, Wiley.

Qi, M. and G. P. Zhang 2001. "An investigation of model selection criteria for neural network time series forecasting." *European Journal of Operational Research* **132**(3): 666-680.

Reed, R. D. and R. J. Marks 1999. *Neural smithing : supervised learning in feedforward artificial neural networks*. Cambridge, Mass., The MIT Press.

Remus, W. and M. O'Connor 2001. Neural networks for time-series forecasting. *Principles of forecasting: a handbook for researchers and practitioners*. J. S. Armstrong. Boston ; London, Kluwer Academic**:** 245-258.

Rumelhart, D. E., J. L. McClelland, et al. 1986. *Parallel distributed processing : explorations in the microstructure of cognition*. Cambridge, Mass., MIT Press.

Smith, K. A. and J. N. D. Gupta 2000. "Neural networks in business: techniques and applications for the operations researcher." *Computers & Operations Research* **27**(11-12): 1023-1044.

Srinivasan, D., A. C. Liew, et al. 1994. "A Neural-Network Short-Term Load Forecaster." *Electric Power Systems Research* **28**(3): 227-234.

Stahlbock, R. 2002. *Evolutionäre Entwicklung künstlicher neuronaler Netze zur Lösung betriebswirtschaftlicher Klassifikationsprobleme*. Berlin, Verlag für Wissenschaft und Kultur (WiKu-Verlag) Dr.Stein.

Swanson, N. R. and H. White 1997. "A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks." *Review of Economics and Statistics* **79**(4): 540-550.

Tang, Z. and P. A. Fishwick 1993. "Feed-forward Neural Networks as Models for Time Series Forecasting." *ORSA Journal on Computing* **5**(4): 374-386.

Thiesing, F. M. 1998. *Analyse und Prognose von Zeitreihen mit neuronalen Netzen*. Aachen, Shaker.

Weigend, A. S. 1992. Predicting Sunspots and Exchange Rates with connectionist Networks. *Nonlinear modeling and forecasting*. M. Casdagli and S. Eubank. New York**:** 395-432.

Wong, B. K., V. S. Lai, et al. 2000. "A bibliography of neural network business applications research: 1994-1998." *Computers & Operations Research* **27**(11-12): 1045-1076.

Zell, A. 2000. *Simulation Neuronaler Netze*. Munich, R. Oldenbourg Verlag.

Zellner, A. 1986. "A tale of forecasting 1001 series : The Bayesian knight strikes again." *International Journal of Forecasting* **2**(4): 491-494.

Zhang, G., B. E. Patuwo, et al. 1998. "Forecasting with artificial neural networks: The state of the art." *International Journal of Forecasting* **14**(1): 35-62.