Prediction of the CATS benchmark using a Business Forecasting Approach to Multilayer Perceptron Modelling

Sven F. Crone
Department of Management Science
Lancaster University
Lancaster, UK
E-mail: crone@bis-lab.com

Heiko Kausch
Institute of Information Systems
University of Hamburg
Hamburg, Germany
E-mail: heiko.kausch@web.de

Dieter Preßmar
Institute of Information Systems
University of Hamburg
Hamburg, Germany
E-mail: pressmar@econ.uni-hamburg.de

Abstract - **Various heuristic approaches have been proposed to limit design complexity and computing time in artificial neural network modelling and parameterisation for time series prediction, with no single approach demonstrating robust superiority on arbitrary datasets. In business forecasting competitions, simple methods robustly outperform complex methods and expert teams. To reflect this, we follow a simple neural network modelling approach, utilising linear autoregressive lags and an extensive enumeration of important modelling parameters, effectively modelling a miniature forecasting competition. Experimental predictions are computed for the CATS benchmark using a standard multilayer perceptron to predict 100 missing values in five datasets.**

I. INTRODUCTION

Artificial neural networks (ANN) have found increasing consideration in forecasting theory, leading to successful applications in various forecasting domains, as well as time series and explanatory forecasting in corporate business [1-4]. ANNs promise attractive features to business forecasting, being a data driven learning machine as opposed to conventional model-based approaches, permitting universal approximation [5] of arbitrary linear or nonlinear functions, and therefore offering great flexibility in learning the generator of noisy data from examples and generalising structure from it without a priori assumptions. However, the nontrivial task of modelling an ANN for a particular prediction problem is still considered to be as much an art as a science [6, 7], as the combination of choices may significantly impact on the networks ability to extrapolate results. Various heuristic modelling approaches have been proposed for the popular paradigm of multilayer perceptrons (MLPs) alone, suggesting alternative approaches to determine the architecture and guide the training process [6, 8-11] to assure robust minimization of the objective function.

In business forecasting, a variety of methods from naïve to complex scientific methods have been developed and applied to minimize forecast errors on short, noisy time series with a low signal to noise ratio [12]. The competing models were evaluated in various time series competitions on different time series [13, 14], with simple methods surprisingly outperforming complex methods and statistical expert teams [15, 16]. Therefore, we attempt to follow a simple business forecasting approach to predict artificial time series outside the business domain, in order to validate these findings relevant for operations research and business forecasting.

Following, we employ a simple MLP approach to derive adequate time series predictions, exploiting the information derived from a preliminary analysis of the linear autoregressive (AR) components as in ARIMA modelling [11] for pre-processing and the selection of appropriate time lags for the input vector. As no heuristic has proven robust performance through valid and reliable results on arbitrary datasets, we exploit computational power available today and propose an extensive enumeration of the most important modelling parameters of network size and depth, activation functions, data sampling strategy, size of the data subsets, initialisation ranges and learning parameters for number of necessary initialisations while successively extending the input vector. Consequently, we model, train and compare a large variety of nonlinear AR-ANNs. The modelling and training of thousands of MLPs 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 models. 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 [13, 16].

Following a brief introduction to the use of MLPs in time series prediction, section 3 assesses the individual modelling and design decisions in MLP training in following a data mining modelling process. This is followed by our experimental results on the CATS benchmark in section 4. Conclusions are given in section 5.

0-7803-8359-1/04/$20.00 ©2004 IEEE

II. MULTILAYER PERCEPTRONS FOR TIME SERIES PREDICTION

Forecasting with non-recurrent ANNs may encompass prediction of a dependent variable $\hat{y}$ from lagged realisations of the predictor variable y_{t-n}, one or i explanatory variables x_i of metric, ordinal or nominal scale as well as lagged realisations thereof $x_{i,t-n}$. Therefore, ANNs offer large degrees of freedom towards the forecasting design, permitting explanatory or causal forecasting through estimation of a functional relationship of the form

$$\hat{y} = f(x_1, x_2, ..., x_z) \quad , \tag{1}$$

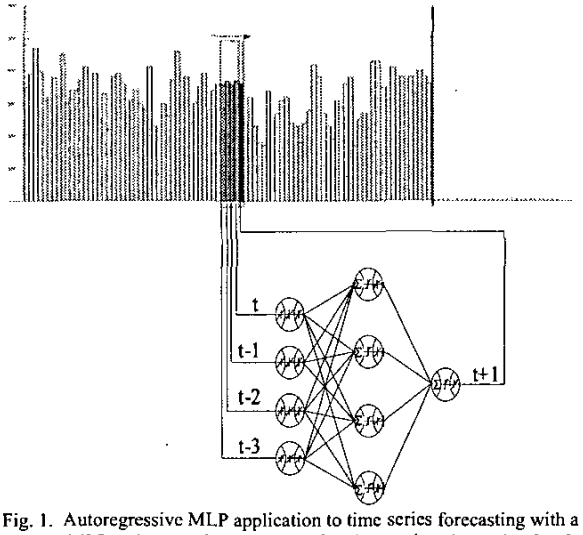

Fig. 1. Autoregressive MLP application to time series forecasting with a (4-4-1)-MLP, using $n=4$ input neurons for observations in t, t-1, t-2, t-3, four hidden units, one output neuron for time period $t+1$ and two layers of 20 trainable weights [22] The bias node is not displayed.

as well as general transfer function models and univariate time series prediction. Considering the objective of the CATS benchmark we will give a brief introduction to modelling ANNs for time series prediction. For a general discussion readers are referred to [12, 17-21]. Forecasting time series with ANN is generally based on modelling the network in analogy to an non-linear autoregressive AR(p) model [6, 8, 22]. At a point in time t, a one-step ahead forecast $\hat{y}_{t+1}$ is computed using $p=n$ observations $y_t, y_{t-1}, \cdots, y_{t-n+1}$ from n preceding points in time t, t-1, t-2, ...; t-n+1, with n denoting the number of input units of the ANN. This models a time series prediction of the form

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

The architecture of a feed-forward MLP, one of the original and well researched ANN paradigms, of arbitrary topology is displayed in figure 1. For a $t+n$ forecast of n steps ahead two approaches for MLP predictions are feasible. Either an iterative $t+1$ forecast is computed using one output node, predicting future realisations in $t+n$ V $n>1$ based upon previous forecasts used as observations y_{t+n-1}. Alternatively, a MLP may be trained to forecast $t+1, t+2, ..., t+n$ values simultaneously, using a multiple-step-ahead forecasting architecture of n output units.

Data is presented to the MLP as a sliding window over the time series observations. The task of the MLP is to model the underlying generator of the data during training, so that a valid forecast is made when the trained network is subsequently presented with a new value for the input vector [5]. Therefore the objective function used for ANN training determines the resulting system behaviour and performance.

III. EXAUSTIVE MODELLING APPROACH FOR MULTILAYER PERCEPTRON APPLICATION

A. Experiment and Training Objective

The objective of the Competition on Artificial Time Series (CATS) is to provide predictions with a minimum mean squared error (MSE) for 100 missing values in 5 subsets of a supplied dataset. The submitted methods will be ranked on the MSE computed on the missing values using

$$E = \frac{\sum_{t=981}^{1000} e_t^2 + \sum_{t=1981}^{2000} e_t^2 + \sum_{t=2981}^{3000} e_t^2 + \sum_{t=3981}^{4000} e_t^2 + \sum_{t=4981}^{5000} e_t^2}{100} \quad , \tag{3}$$

with $e_t^2 = (y_t - \hat{y}_t)^2$ determining the prediction objective and the objective function through the ex post evaluation measure (see II.D). It must be noted, that the MSE as all quadratic errors represents a biased error metric and objective function, penalizing errors of larger scale. Although the selection of appropriate, non quadratic error metrics has received large attention in business forecasting research [23], MSE error measures may be suitable if it represents the true objective of the problem domain, e.g. modelling filters in engineering domains where larger errors are unfavourable [24].

B. Data Analysis and Pre-processing

The CATS benchmark is an artificial time series of 5,000 observations with 100 values omitted for out of sample evaluation. The omitted observations are divided in 5 blocks of 20 observations: elements 981 to 1,000, 1981 to 2000, 2981 to 3000, 3981 to 4000 and elements 4981 to 5000. Lacking information on the domain and structure of the time series no data cleansing regarding structural breaks, correction of outliers, pulses or level shifts due to temporal external shocks etc. may be performed. In absence of a heuristic lag structure derived from the typical seasonality of a quarterly, monthly or weekly time series, we analyze the complete dataset as well as each individual set for linear autocorrelations in the integrated data.

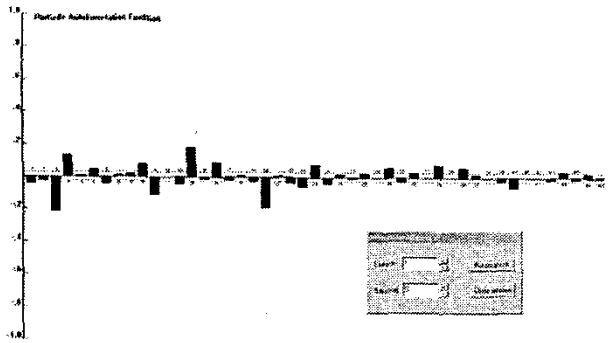

Fig. 2. Partial autocorrelation function of first order integrated time series over all partial dataset d1, ..., d5

Assuming the nature of an artificial time series, we expect to find an identical structural generator for the data hidden by noise in all independent samples of the CATS datasets. Detecting strong instationarity, we analyze the linear autocorrelation coefficients and the partial autocorrelation coefficients of the first order integrated time series, shown in fig. 2 and 3 respectively. Our analysis confirmed similarities in the linear lag-structure, with significant lags at *t-3, t-4, t-10, t-11, t-14* and *t-16* in all datasets. In addition, we detected significant partial autocorrelations at *t-1* in three, and *t-6, t-12* and *t-18* in two datasets attributed to different impact of random noise without significant lags larger than *t-18* within the separate datasets. These lags were utilized for a stepwise extension of the input vector. Although these findings support an approach to join the datasets and train each single MLP on all data available, thereby reducing the adverse influence of noise and possibly deriving robust results in the individual sets we attempted to use local information of time series level and noise. Following, all analysis, training, evaluation and model selection is computed on the five distinct datasets. Based on our autocorrelation analysis, first order differencing of the time series could reduce noise in the data structure, although there persists 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 [25, 26]. Following preliminary tests on a limited set of NN architectures we transform the input and output variables through first order integration of the complete data series, albeit loosing information on the scale of the observations in order to reduce noise, highlight important relationships and compact the distribution of the variables to assist the neural network in learning the relevant patterns [8, 22]

Subsequent to transformation, we normalize the integrated data in order to avoid computational problems, to meet algorithm requirements and to facilitate network learning and speeding up the training process [27] in accordance with the selected nonlinear hyperbolic tangent (tanh) activation function to (-0.8;0.8) including headroom h to avoid saturation effects of observations close to the asymptotic limits of the functions [4]. Various scaling methods allow linear equidistant scaling in arbitrary intervals [6] or statistical normalisation [9]. We select a simple normalisation

$$y_t' = \frac{y_t}{y_{\max} + h} \quad , \tag{4}$$

to scale the data symmetric to the origin to account for non-symmetric intervals of scale in positive and negative observations [28]. We assure homogeneous of all separated datasets through an external, along channel normalization of the complete time series [17].

C. *Architecture Selection*

Designing a MLP architecture encompasses the dataset dependent decisions [6] on the number of hidden layers, the number of nodes in input, each hidden and output layers as well as the functional processing within the nodes through input, activation and output function [6].

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 [11]. There exist no dominant heuristic to determine the optimum number for an arbitrary dataset, although this issue is essential to the quality of the following model building process [6]. Too few nodes leave out relevant information from the exploitable information in the underlying linear and nonlinear autoregressive structure of the time series, while too many nodes may add noise to the input patterns. We employ a autoregressive modelling approach following Lattermacher and Fuller [28], exploiting the information derived from a preliminary analysis of the linear autoregressive (AR) components as in ARIMA modelling [17] to select the appropriate time lags for the input vector and consequently 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 while omitting MA-terms. We commence model building with the linear AR-lags determined in the data analysis in five groups of increasing length, successively extending the input vector by including less relevant autocorrelation-lags until all past observations from lags *t-1,...,t-18* are used.

The number of output nodes is determined by the forecasting horizon of the time series forecasting problem. 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 n output nodes as proposed by [24, 29]. We evaluate both approaches in a preliminary evaluation, with the iterative $t+1$ predictions routinely outperforming the multiple step ahead forecasts. As a

consequence, we limit our presented results to those from the iterative $t+1$ predictions.

The number of hidden layers and corresponding hidden nodes in each layer is determined using an extensive approach, evaluating every combination of $l=1,...,3$ hidden layers and $n=1,...,20$ hidden neurons in steps of 2, limiting the models to pyramidal topologies with an equal or smaller number of nodes in successive layers and a maximum number of 20 nodes in total, leading to 54 different topologies. We consider only fully connected, non-recurrent feedforward architectures without shortcut connections.

Information processing within nodes is set homogeneously throughout all experiments. For the processing functions we considered the summation as an input function, a linear output function and the hyperbolic tangent (tanh) as a nonlinear activation function in all hidden and output nodes, due to presumed advantages in error propagating behaviour [30]. The bias in each node is modelled as an on-node connected to all nodes in all hidden and output layers with trainable weights.

D. *Training Process*

Training a MLP is the task of adjusting the weights of the links w_{ij} between units j and their thresholds to minimize the error δ_j between the actual and desired system behaviour [16] 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. The objective function in ANN training

$$E = f(t_j, o_j) \quad , \tag{5}$$

determines the size attributed to each δ_j the MLP outputs, and may therefore be interpreted as measuring the significance of an over-prediction or an under-prediction for each point in time t or pattern p. The impact of selecting an appropriate objective function according to the learning problem and the ex post measure has received limited attention in the business forecasting domain [31]. As different error measures imply different error weights to deviations of the predicted values to the observations, a MLP should be trained, selected and evaluated on its ability to adhere to the objective [30]. As the objective function should reflect the design goal of the competition, we minimize the MSE. For one output node the MSE equals the popular sum of squared error (SSE) objective function as proposed by Rumelhart, Hinton and Williams [19, 20]:

$$E_p = \frac{1}{2}\sum_j (t_{pj} - o_{pj})^2 \quad . \tag{6}$$

The use of the modified *SSE* in hetero-associative MLP training parallels statistical regression problems, modelling the conditional distribution for normally distributed output variables. In time series point prediction, the single network output o_p corresponds to the forecast $\hat{y}_t$ of a network, the teaching input t_p represents the actual value of the observation y_t and the forecast error e_t represents the networks error δ_j in the output-layer.

We apply various variants of different learning rates and cooling rates, with four variants of learning rates between $\eta = 0.05,...,0.45$ in steps of 0.2 and three variants of stepwise reductions of $\varpi = 0.980, 0.990, 0.995$ per epoch, deriving a variety of 9 alternative learning schemes. Each network topology is trained for up to 600.000 iterations, with the weight configuration causing the lowest MSE on the validation set saved for future use. To reduce computation time, we apply an early stopping paradigm, evaluating the relative reduction of the network error in percent after every epoch to a 0% threshold in 15 epochs. To account for random initialisation of the connection weights, we initialise each MLP 10 times prior to training. We apply three different initialisation ranges of [-0.33; 0.33], [-0.66, 0.66] and [-0.99; 0.99]. No pruning or growing heuristics were applied.

Due to changing input vector sizes the overall number of patterns in the dataset varies. We divide the dataset into three distinct sub-samples of training set to parameterize the trainable weights, a validation to guide early stopping and prevent overfitting and a test set to evaluate generalization on a hold-out set. Three different subsample ratios were drawn, using a split of [60%, 40%], [70%, 30%] and [80%, 20%], omitting the generalization set for final training due to the high correlation of the structure between subsamples of the datasets. Each data pattern is drawn randomly from the sample sets until all patterns are used to complete an epoch.

E. *Model Evaluation and Selection*

Following the training of various architectures, a presumably optimum model must be selected from all available models for the final prediction. In accordance with the objective function and final evaluation criteria of MSE, we select the model with the lowest MSE on the validation data set. It must be noted, that the naïve selection of the network model with the lowest validation error does not seem to guarantee robust results. Therefore we followed a stepwise approach by analysing the mean errors of each parameter variant of sampling strategy, initialisation range etc. and iteratively selecting the ANN model group with the lowest Median of all MSEs on the validation dataset. However, the noise in results of the best MLPs sorted by validation error suggests that a pure selection based upon validation error is questionable to assure a valid and reliable model selection. Therefore, future research should consider the development of sound selection methods instead of additional heuristic guidelines for network modelling.

To evaluate our selection process in the lack of generalisation data for the CATS competition, we analyze

the autocorrelation structure of the iterative predictions for each dataset. Despite the low signal to noise ratio on a sample size of 20 forecasts, we would anticipate a similar autoregressive pattern in the predictions, which was not always evident. As the true observations of the competition are not available at the time of writing, we anticipate a limited accuracy of our predictions, further highlighting the problem of robust model selection.

IV. Simulation Results of Exhaustive Modelling on the CATS Benchmark

A. *Selected MLP Architecture*

We train and evaluate a total of 43740 different MLPs for each input lag-structure and t+1 forecasts. Total computation time was appx. 10 days on a Pentium IV, 2400 MHz, 1GB RAM, with an average time of 4 seconds for network training and saving of the results. To automate this extensive design and model selection process we apply *NeurLab*, a prototype simulator for ANN time series experiments developed specifically for extensive time series predictions within our research group. Table 1 gives an overview.

TABLE I
ARCHITECTURES EVALUATED FOR MODEL SELECTION

ARCHITECTURE	EVALUATED PARAMETERS
Input Nodes / Lag Structure	3,4,10,11,14,16 1,3,4,10,11,14,16,18 1,3,4,5,6,9,10,11,12,14,16,18 1,2,3,4,56,9,10,11,12,14,16,18 1,...,18
Hidden Nodes	hidden layer 1: 1,...,20 hidden layer 1 + 2: SUM 1,...,20 hidden layer 1+ 2 + 3: SUM 1,...,20
Output Nodes	1 (iterative t+1 prediction) 20 (t+1, ..., t+20 prediction)
Activation Function	TanH
TRAINING PROCESS	
Iterations	600000
Initialisations	10
Weight Initialisation Range	[-0.33; 0.33], [-0.66; 0.66], [-0.99;0.99]
Learning rate	0.05, 0.25, 0.45
Learning rate decay	0.98, 0.99, 0.999
Early Stopping criteria	error decrease of 0%
Early stopping cycle	evaluation every epoch trained
Early stopping evaluation	training continued for 15 epochs
Data Splitting [Train/Valid]	[60%; 40%], [70%, 30%], [80%, 20%]
Data Sampling	without replacement

Of all evaluated architectures, we selected a 12-8-1 architecture named BISLAB.a, using the lags *t-1...6,9...12,14,16,18* for input nodes. The MLP was trained for 60000 iterations using a starting learning rate of 0.05 being reduced by a cooling factor of 0.995 (0.5%) every epoch, with early stopping after 15 evaluation cycles with no relative improvement of the MSE after every epoch on the validation data. The dataset was split into 70% training data and 30% validation data to train the found MLP architecture to generate the final forecasts. The architecture was found on dataset 1 and successively trained and applied to all datasets.

B. *Simulation Results*

Following, we present the numerical and selected graphical predictions of the chosen MLP-model for the datasets of the CATS competition.

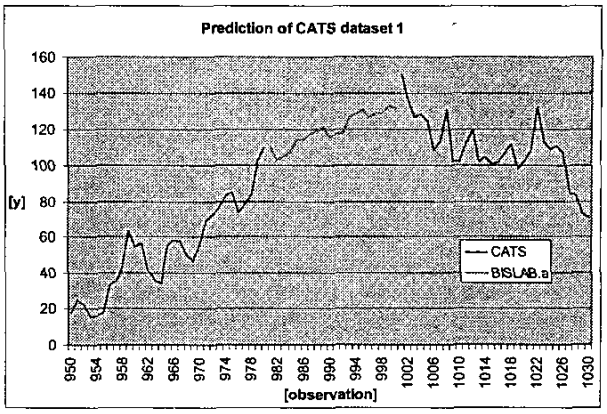

Fig. 4. Results of BISLAB.a prediction on the CATS dataset 1.

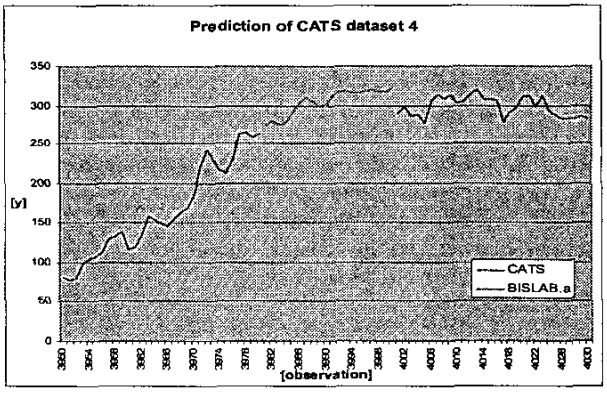

Fig. 6. Results of BISLAB.a prediction on the CATS dataset 4.

Graphic evaluations of time series graphs do not necessarily reveal the performance of a predictor, especially considering the unknown level of noise in the time series. However, a 'sanity check' of the level and signal to noise ratio of the predicted values suggests a medium forecasting performance, especially for datasets 2 and 3.

Table 2 displays the numerical predictions of the trained model on all datasets, rounded to two decimals. The actual predictions are submitted in full numerical detail.

TABLE II
PREDICTION RESULTS ON THE CATS DATASETS FOR BISLAB.A

	BISLAB.a Rounded Predicted Values
CATS dataset 1	110.34, 103.07, 104.24, 107.78, 113.84, 113.79, 117.38, 118.98, 121.36, 115.49, 117.16, 117.73, 127.16, 128.57, 131.00, 126.41, 129.14, 129.09, 133.31, 131.78
CATS dataset 2	407.55, 407.73, , 407.16, 391.03, 394.53, 388.77, 399.53, 397.76, 395.47, 399.93, 391.30, 392.93, 381.80, 386.38, 384.44, 386.29, 389.14, 382.52, 381.36, 374.44
CATS dataset 3	91.69, 88.73, 92.58, 83.19, 77.07, 71.49, 71.64, 71.19, 71.86, 66.65, 74.76, 69.82, 75.31, 64.25, 71.51, 65.08, 70.16, 63.43, 64.08, 57.08
CATS dataset 4	274.86, 280.78, 275.72, 279.14, 291.03, 303.91, 310.44, 305.88, 299.58, 300.80, 312.10, 316.99, 318.57, 316.13, 316.35, 318.72, 320.02, 317.80, 317.04, 321.36
CATS dataset 5	-65.35, -75.76, -68.94, -64.03, -63.07, -73.28, -68.88, -64.85, -55.43, -63.09, -62.45, -66.71, -60.97, -65.85, -63.49, -70.85, -67.26, -68.42, -66.20, -73.26

V. CONCLUSION

We have predicted values for the CATS forecasting competition benchmark applying an extensive architecture evaluation and selection approach motivated by business forecasting competitions. The limitations of an extensive approach of long computational time and the need to fully automate the complete training, evaluation, selection and forecasting process of thousands of networks are no longer dominant through available computational power. However, the training of thousands of neural network architectures shifts the emphasis in neural network modelling from limiting the degrees of freedom in parameter estimation towards the problem of a valid and reliable selection criterion to determine superior network topologies. Future research is needed to derive sound selection rules exceeding a naïve pick-best on a single error measure.

However, the actual performance of our approach may only be evaluated by the accuracy of our predictions in this and future forecasting competitions.

REFERENCES

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

[2] F. M. Thiesing and O. Vornberger, "Forecasting Sales Using Neural Networks Computational Intelligence," Proceedings of International Conference, 5th Fuzzy Days, Dortmund, Germany, 1997.

[3] F. M. Thiesing and O. Vornberger, "Sales forecasting using neural networks," International Conference on Neural Networks, 1997.

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

[5] K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks are Universal Approximators," *Neural Networks*, vol. 2, pp. 359 - 366, 1989.

[6] 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.

[7] C. Chatfield, "Neural Networks - Forecasting Breakthrough or Passing Fad," *International Journal of Forecasting*, vol. 9, pp. 1-3, 1993.

[8] A. Lapedes, R. Farber, and Los Alamos National Laboratory., "Nonlinear signal processing using neural networks prediction and system modelling." Los Alamos, NM: Los Alamos National Laboratory, 1987.

[9] A. S. Weigend, "Predicting Sunspots and Exchange Rates with connectionist Networks," in *Nonlinear modeling and forecasting*, M. Casdagli and S. Eubank, Eds. New York, 1992, pp. 395-432.

[10] C. de Groot and D. Wurtz, "Analysis of univariate time series with connectionist nets: A case study of two classical examples," *Neurocomputing*, vol. 3, pp. 177-192, 1991.

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

[12] S. G. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting : methods and applications*. New York: Wiley, 1998.

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

[14] S. G. Makridakis, "The M2-Competition : a real-time judgmentally based forecasting study," *International journal of forecasting*, vol. 9, pp. 5-22, 1993.

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

[16] R. Fildes, M. Hibon, S. Makridakis, and N. Meade, "Generalising about univariate forecasting methods: further empirical evidence," *International Journal of Forecasting*, vol. 14, pp. 339-358, 1998.

[17] A. Zell, *Simulation Neuronaler Netze*. Munich: R. Oldenbourg Verlag, 2000.

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

[19] S. S. Haykin, *Neural networks : a comprehensive foundation*, 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 1999.

[20] C. M. Bishop, *Neural networks for pattern recognition*. Oxford: Clarendon Press, 1995.

[21] I. Kaastra and M. Boyd, "Designing a neural network for forecasting financial and economic time series," *Neurocomputing*, vol. 10, pp. 215-236, 1996.

[22] D. Srinivasan, A. C. Liew, and C. S. Chang, "A Neural-Network Short-Term Load Forecaster," *Electric Power Systems Research*, vol. 28, pp. 227-234, 1994.

[23] J. S. Armstrong, "Principles of forecasting: a handbook for researchers and practitioners," in *International series in operations research & management science ; 30*. Boston ; London: Kluwer Academic, 2001, pp. xii, 849.

[24] R. Fildes and S. Makridakis, "Forecasting and loss functions," *International Journal of Forecasting*, vol. 4, pp. 545-550, 1988.

[25] T. Hill, M. O'Connor, and W. Remus, "Neural network models for time series forecasts," *Management Science*, vol. 42, pp. 1082-1092, 1996.

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

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

[28] G. Lachtermacher and J. D. Fuller, "Backpropagation in time series forecasting," *Journal of Forecasting*, pp. 381-393, 1987.

[29] A. Zellner, "A tale of forecasting 1001 series : The Bayesian knight strikes again," *International Journal of Forecasting*, vol. 2, pp. 491-494, 1986.

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

[31] D. E. Rumelhart, J. L. McClelland, and University of California San Diego. PDP Research Group., *Parallel distributed processing : explorations in the microstructure of cognition*. Cambridge, Mass.: MIT Press, 1986.