Input-variable Specification for Neural Networks - an Analysis of Forecasting low and high Time Series Frequency

Sven F. Crone and Nikolaos Kourentzes

Abstract - **Prior research in forecasting time series with Neural Networks (NN) has provided inconsistent evidence on their predictive accuracy. In management, NN have shown only inferior performance on well established benchmark time series of monthly, quarterly or annual frequency. In contrast, NN have shown preeminent accuracy in electrical load forecasting on daily or hourly time series, leading to successful real life applications. While this inconsistency has been traditionally attributed to the lack of a reliable methodology to model NNs, recent research indicates that the particular data properties of high frequency time series may be equally important. High frequency time series of daily, hourly or even shorter time intervals pose additional modelling challenges in the length and structure of the time series, which may abet the use of novel methods. This analysis aims to identify and contrast the challenges in modelling NN for low and high frequency data in order to develop a unifying forecasting methodology tailored to the properties of the dataset. We conduct a set of experiments in three different frequency domains of daily, weekly and monthly data of one empirical time series of cash machine withdrawals, using a consistent modelling procedure. While our analysis provides evidence that NN are suitable to predict high frequency data, it also identifies a set of challenges in modelling NN that arise from high frequency data, in particular in specifying the input vector, and that require specific modelling approaches applicable to both low and high frequency data.**

I. Introduction

The past years have seen a resurgence of interest modelling artificial neural networks (NN) for time series prediction, both in research and practice [1]. A recent literature survey reveals over 5,000 publications on NN in time series prediction, with successful applications across various forecasting domains (see e.g. [2, 3]), in academic research [4, 5] and in practice [6]. However, in management research, the majority of publications have limited their evaluation of NN to predicting low frequency data. A literature review identified that 74 of 102 publications (73%) analysed the performance of NN on low frequency time series, i.e. time series of annual, quarterly, monthly or weekly observation intervals. In contrast, the evaluation of NN in predicting time series of higher frequency has received lesser attention, despite the widespread existence of high-frequency data in electrical load forecasting [7-9], traffic predictions [10, 11], finance [12-14] and macroeconomics [15]. While no common agreement exists on what constitutes low and high frequency data across domains, time series of daily or shorter time intervals are generally characterised as high frequency data [16]. While all time series essentially consist of combined archetypical time series patterns of seasonality, trends, levels, structural breaks, outliers and calendar effects, it is argued that high frequency data poses a new set of forecasting problems, that make conventional methods inappropriate [17] and demand new approaches regarding methods, methodologies and computational resources [7]. Recent research in econometrics and finance by Markham and Rakes [18, 19] as well as Hu et al. suggests that NN can perform particularly well on high frequency data due to the particular data properties, which has been supported by some empirical evidence in electrical load forecasting [6]. However, NN have not been analysed regarding their adequacy and challenges in predicting data of different time frequencies, leaving both fields of low-frequency and high-frequency time series disconnected with inconsistent findings.

The aim of this study is to explore the accuracy and modelling challenges of NN that arise from different levels of time series frequency. We conduct a set of experiments to predict an empirical time series of daily cash withdrawals taken from the NN5 competition (www.neural-foecasting-competition.com), which is aggregated to daily, weekly and monthly levels of time frequency. The stepwise aggregation enables an analysis of the changes in accuracy and of the appearance of novel challenges in the modelling process during the transition from low to high frequency data. While methodologies to specify the number of hidden layers, number of hidden nodes in each layer, activation functions, learning parameters etc. (commonly based on wrapper approaches) remain largely unaffected by the time series frequency, the data properties show a direct impact on the specification and length of the input vector. Consequently, we evaluate a set of alternative heuristic and statistical techniques for selecting the time-lagged input variables and their impact on forecasting accuracy. The accuracy of the NN is compared to statistical benchmark methods in each of the frequency domains and in a bottom-up aggregation of the daily predictions to weekly and monthly levels in order to evaluate potential increases in accuracy in lower time frequency from predictions using high-frequency data.

The paper is organised as follows: section II briefly introduces the methods and different methodologies of input-

Manuscript received January 15, 2009. Nikolaos Kourentzes (corresponding author) and Sven F. Crone are with the department of Management Science at Lancaster University Management School, Lancaster, LA1 4YX, United Kingdom. (phone: +44.1524.5-92991; fax: +44.1524.844885; e-mail: {n.kourentzes; s.crone}@lancaster.ac.uk).

978-1-4244-3553-1/09/$25.00 ©2009 IEEE

vector specification for NN, followed by information on the time series and the experimental design in section III. Section IV discusses the results for each frequency domain and across frequency domains using a bottom-up comparison. In section V we identify characteristic modelling challenges of NN on different time frequencies, followed by conclusions and further research in section VI.

II. Forecasting with Neural Networks

A. Multilayer Perceptrons for Time Series Prediction

We limit our evaluation to the common multilayer perceptron (MLP), which represents the most widely employed NN architecture [1]. The advantage of MLPs is that they are well researched regarding their properties and their proven abilities in time series prediction to approximate and generalise any linear or nonlinear functional relationship to any degree of accuracy [20] without any prior assumptions about the underlying data generating process [21], providing a powerful forecasting method for linear or non-linear, non-parametric, data driven modelling. In univariate forecasting feed-forward architectures of MLPs are used to model nonlinear autoregressive NAR(p)-processes, using only time lagged observations of the time series as input variables to predict future values [22], or intervention modelling of NARX(p)-processes using binary dummy variables to code exogenous events as explanatory intervention variables. Data are presented to the network as disjunct vectors of a sliding window over the time series history. The neural network learns the underlying data generating process by adjusting the connection weights $w = (\beta, \gamma)$ to minimise an objective (squared error loss) function on the training data to make valid forecasts on unseen future data [23]. We employ a single hidden layer MLP to forecast a future value $\hat{x}_{t+h|t}$:

$$\hat{x}_{t+h|t} = f(X, w) = \beta_0 + \sum_{h=1}^{H} \beta_h g\left(\sum_{i=0}^{I} \gamma_{hi} x_i\right) \quad , \qquad (1)$$

with t denoting the point in time, h the forecasting horizon and $X = [x_0, x_1, \ldots, x_n]$ the input vector of the time lagged observations of the time series X_t. The parameters I ($i = 1, \ldots, I$) and H ($h = 1, \ldots, H$) specify the number of input and hidden units of the network architecture, and $g(\cdot)$ is a non-linear transfer function in the hidden layer nodes [24].

Modelling a NN for time series data requires decisions on a number of architectural parameters, including the number of input nodes, hidden layers, nodes per hidden layers, activation functions, training parameters of learning algorithm, learning rates, early stopping criteria etc. An adequate NN architecture is routinely determined by using simulations on the time series: a set of candidate MLPs is trained using different architectural parameters and the architecture with the lowest in sample error is selected.

B. Input Variable Selection for Time Series Prediction

While the specification of NN architectures is still under discussion in research [1] multiple publications have identified the adequate selection of the input vector as one of the most important decisions for accuracy. As time series of different frequency may display varying time series patterns, including the appearance of multiple levels and forms of seasonality, changes in the magnitude of seasonality, trend and randomness, a suitable the input vector must be identified for each time series frequency. For the stationary and seasonal time series of the NN5, modelling the information contained in the seasonality is potentially necessary [25]. Consequently, we seek to evaluate multiple architectures and different approaches of input variable selection for each time series of a specific time frequency.

Different methodologies to specify input vectors have been suggested and explored for low frequency data, but without adequate evaluation on high-frequency data. In order to reflect possible interactions of the time series frequency with the input vector methodology and the resulting number of input nodes we evaluate the four most popular analytical approaches to specify the input vector (not selecting an input vector of arbitrary size), as identified by a literature review. The most common approach of input variable selection for NN applies a stepwise linear regression model with statistical testing to identify significant time lags and use those to specify the input vector for the NN [26-28], despite evidence in econometrics and time series modelling that this may lead to suboptimal and misspecified input variables.

Alternatively, we may specify the input vector following the popular statistical Box-Jenkins methodology of ARIMA modelling, which has demonstrated promising results [23, 29]. The autocorrelation function (ACF) and the partial autocorrelation function (PACF) of the time series is analysed in order to identify and select significant time-lagged realisations. As a feed-forward MLP models an autoregressive NAR(p)-process (without explicit MA(q) components of a moving average process), we may limit the analysis to the PACF. The conventional algorithm to calculate the PACF utilises the Yule-Walker equations, which estimate the true PACF by minimising the forward regression error in the least squares sense. Alternatively, the PACF may be estimated using the Burg algorithm, by minimising both the forward and backward error, thereby providing a more accurate estimation of the autoregressive structure of the time series [30] at the cost of being computationally more intensive.

A simple expansion of these algorithms combines the autoregressive lag-structure identified by the PACF with the information on moving average processes contained in the ACF function, selecting all lags that are statistically significant in both ACF and PACF, as suggested in [23]. These four methods warrant further evaluation in order to establish their comparative accuracy in time series prediction with MLPs, as we will show in later experiments.

Across all methodologies for input-variables specification, including stepwise Regression and PACF / ACF analysis, one problem arises in the mandatory ex ante specification of the maximum number of lags one should include in the evaluation. For non-stationary time series, specifying the input-vector via ACF/PACF will lead to a large number of significant autocorrelations, essentially requiring a maximum cut-off of a maximum lag to be considered. Despite the

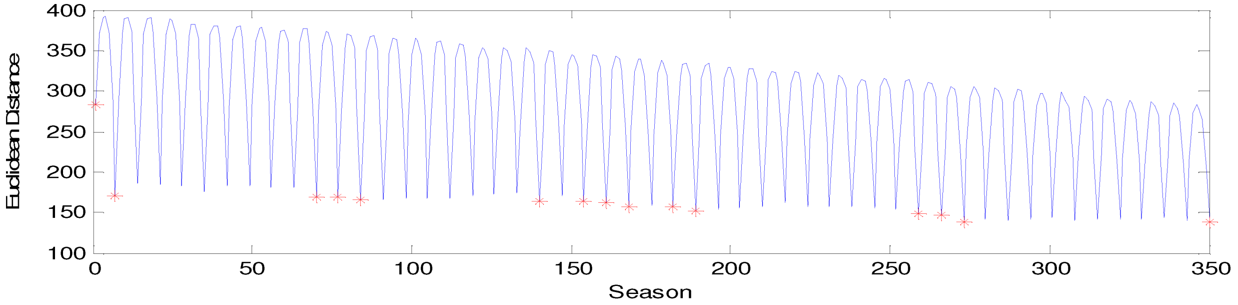

Fig. 1. Seasonal length plotted against the Euclidean distance between seasons. The asterisks signify the local minima.

substantial impact of this meta-parameter, the issue has not been explored, with the exception of Balkin and Ord [31] for the cases of yearly, quarterly and monthly time series of low frequency. A common practice resorts to heuristic trial and error approaches, or setting an arbitrary number of lags as multiples of a full season. While these (suboptimal) heuristics may be feasible for low frequency data, they fail for high frequency data as the increased length of the time series and the multiple overlapping patterns can lead to a large number of significant lags and very long input vectors. This mirrors a common challenge of statistical tests in data mining: the sheer size of the available dataset leads to most test becoming statistically significant [32]. In forecasting high-frequency time series, the large number of data points will induce low significance levels and hence indicate most lags in the ACF and PACF as significant, creating inflated input vectors. This aspect warrants further investigation. In addition, ACF and PACF information may be masked through multiple overlying seasonalities, that require an iterative model building and residual analysis for valid identification.

To limit iterative modelling and identify an efficient suitable maximum lag of the input-variables we propose a simple approach to identify the true seasonality and the maximum seasonal length of the input vector by measuring the Euclidian distance of a seasonal plot for arbitrary seasonal lengths. A time series of length 1, ..., n is split into seasons of increasing length s, with s=1, ...,S and $S \leq n / 2$, and the Euclidean distance between all n / s seasonal time series is calculated. The seasonal length that minimises the Euclidean distance indicates the minimum possible deviation (in squared error terms) of the seasons, thus providing a possible seasonality and a suitable input variable to capture the seasonality. In fig. 1 the development of the Euclidean distance of the time series NN5-035 (used in section III in the experimental evaluation) is plotted against the seasonal length, identifying multiple suitable seasonalities through local and global minima as multiples of the weekly seasonality. We may utilise this additional information to specify suitable lags and a maximum lag-length in addition to the lags identified through the four conventional methodologies for input-variable specification.

III. Experimental Design

A. Time Series Data

The experiments evaluate the effect of increasing time frequency on a single time series of daily cash withdrawals from cash machines in the UK, taken from the recent NN5 competition dataset of 111 time series (ID# NN5-035). The daily time series consists of two years of data, beginning March 18th 1996 and ending March 22nd 1998. In order to avoid the creation of inconsistencies from the aggregation of the data, the first and last incomplete months that cannot be aggregated are trimmed from the time series, leaving a time series of 23 months or 699 days, just less than two full years of data. The trimmed time series contains 14 missing values, which are imputed by the average of the neighbouring observations. To run experiments on weekly and monthly data of lower frequency the adjusted daily time series is aggregated by summing cash withdrawals over weeks and calendar months respectively. A plot of the daily time series, with both the trimmed and missing values displayed as shaded observations, and the series aggregated to weekly data and monthly data is provided in fig. 2.

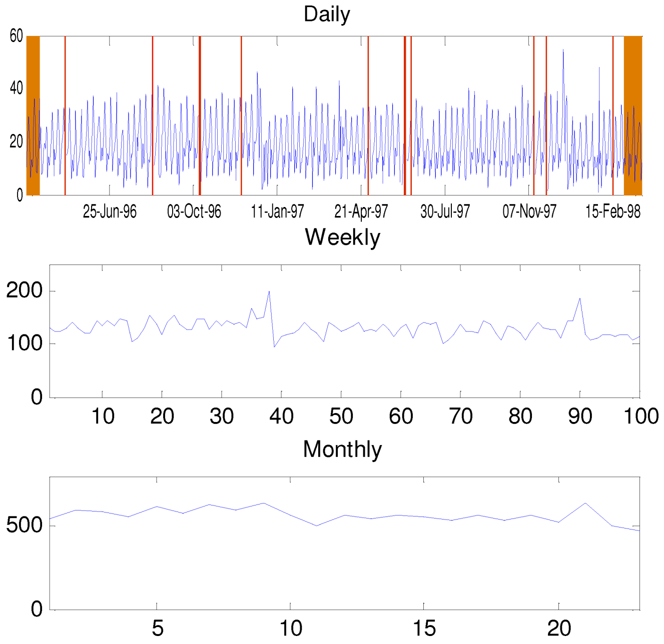

Fig. 2. NN5-035 weekly and monthly aggregated plots

A visual analysis of the three time series reveals various seasonal patterns around a constant level without any trends, as confirmed by a Phillips-Perron test on all three time series. In order to identify single or multiple seasonalities of different length on the time series of different frequency, an analysis of ACF/PACF-plots, periodograms and visual inspections of seasonal year-on-year diagrams were used, of which fig. 3 shows the seasonal plot for the daily time series.

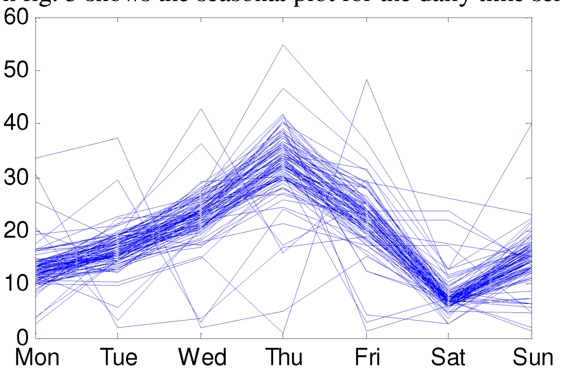

Fig.3. Seasonal week-on-week diagram for the daily time series

The seasonal plot indicates a strong day-of-the-week seasonal pattern, plus some slight instationarity of the level of the stacked weekly lines, which can be attributed to a second seasonality of week in the year. Both periodogram and ACF/PACF-analysis confirm these patterns, with the day-of-the-week pattern obviously missing in the data with lower frequencies of weekly and monthly observations. The yearly cycle provides some challenges in identification from the truncated time series, as no data on two full years is available, which will equally make any modelling difficult.

All time series show a set of systematic seasonal pulses of Christmas and the New Year's Eve during the last 1.5 weeks of each year (18 of December until 31 of December), which are reflected with different intensities in all frequencies of the 13 last daily, last 2 weekly and last monthly observations. These time periods are modelled by using an integer dummy variable as an additional input during MLP training. In addition, the aggregation from daily to weekly and monthly frequencies introduces asymmetries due to the varying number of working days per month (31 days in January vs. 28 days in February) and the potentially different number of days and weeks per year, which is reflected in a binary dummy variable for the February of each year.

B. Experimental setup

The experimental setup of forecasting horizon, error metrics, and test dataset is guided by the design of the original NN5 competition. The forecasting horizon is h=1, 2, ..., 56 days into the future, or the equivalent of 1 to 8 weeks and 1 to 2 months for the lower time frequencies respectively in order to allow a bottom-up comparison of the accuracy across a homogeneous test set despite different time frequencies.

The symmetric mean absolute percent error (sMAPE) is used to evaluate and compare the competing modelling approaches, as in the NN5. It computes the absolute error in percent between the actuals X_t and the forecast F_t for all periods t of the test set of size $n=h$ for each time origin:

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{\left|X_t - F_t\right|}{(X_t + F_t)/2}\right)100 \quad . \tag{2}$$

Both the validation and test datasets contain 84 days (or the equivalent of 12 weeks or 3 months for different time frequency). All models are evaluated using a rolling time origin evaluation, evaluating the average accuracy across 28 daily time origins (and 4 origins for weekly and 2 for monthly data respectively) in order to increase the validity and reliability of the results in contrast to a single fixed origin evaluation [33]. The accuracy of the competing NN models is evaluated for statistically significant differences (5%) using the nonparametric Friedman test and the Nemenyi test, to facilitate an evaluation of nonparametric models without the need to relax assumptions of ANOVA or similar parametric tests [34].

C. Neural Network Architectures

The Evaluation encompasses MLP models using different input-vector specifications and statistical benchmarks to compare the predictive accuracy of different approaches. All MLP models use identical architecture and training parameters, with the exception of varying the number of input nodes and hidden nodes. To evaluate the accuracy of the competing methodologies for input-vector specification, we will identify lags using the four methodologies of stepwise regression, PACF Burg-algorithm, PACF Yule-Walker-algorithm, and the combined PACF & ACF algorithm using the Yule-Walker equations. For the weekly and the monthly time series, the stepwise regression based models could not identify any significant lags, so no NN was trained. In addition to these four methodologies, we create a second set of input-vectors that combine the time lags specified by each of the methods with the time lags identified using the Euclidean distance, introducing a maximum seasonal lag to each. The NN architectures of input and hidden nodes for the three aggregation levels of daily, weekly and monthly data are summarised in table I. Each input vector methodology is identified by the name of the underlying algorithm, with the suffix '*S*' denoting those extended with the lags from the Euclidian distance chart.

Each MLP is trained with the corresponding number of nodes as specified through the input-vector methodology. All topologies have a single output node with the identity function as a linear activation function with iterative predictions for multiple-steap ahead forecast. An adequate number of hidden nodes for each time series frequency is pre-determined from a set of H_i= 1, ... , 12 nodes through experimentation for each of the time series using errors from the in-sample data only, selecting a set of H_d=8, H_w=5 and H_m=9 hidden nodes for the daily (d), weekly (w) and monthly (m) time series respectively, all using the hyperbolic tangent (TanH) activation function.

For training, we apply a standard gradient descent learning using backpropagation with momentum for the MLP training, applying an initial learning rate of η=0.5 which is

TABLE I
SUMMARY OF NEURAL NETWORK ARCHITECTURES

Frequency	Daily Data			Weekly Data			Monthly Data		
		# of nodes			# of nodes			# of nodes	
Model	Lags*	Input	Hidden	Lags*	Input	Hidden	Lags*	Input	Hidden
PACF Yule	1, 3-9, 11, 13-16, 21, 28-29, 36, 65, 108	19	8	1-2	2	5	1	1	9
PACF Burg	1, 3, 5-9, 11, 13-15, 20, 22, 29, 36	15	8	1	1	5	1	1	9
PCAF+ACF-Yule	1-189	189	8	1-2	2	5	1	1	9
Stepwise Regression	1, 7, 35, 56, 83-84, 99, 169, 174, 182, 189	12	8	-	-	-	-	-	-
PACF-Yule-S	1, 3-9, 11, 13-16, 21, 28-29, 36, 65, 108, 189	17	8	1-2, 25	3	5	1, 7	2	9
PACF-Burg-S	1, 3, 5-9, 11, 13-15, 20, 22, 29, 36, 189	6	8	1, 25	2	5	1, 7	2	9
PACF+ACF-Yule-S	1-189	189	8	1-2, 25	3	5	1, 7	2	9
Stepwise Regression-S	1, 7, 35, 56, 83-84, 99, 169, 174, 182, 189	12	8	-	-	-	-	-	-

*The Lags specify the time lagged realisations *t-n* used as inputs, with the number of lags equalling the number of input nodes.

reduced each epoch by 1% and a constant momentum term of φ=0.4. Each MLP is trained for a maximum of 1000 epochs using early stopping, where training is aborted if the MSE on the validation data does not improve by 1% within 100 epochs. The weight-vector with the lowest MSE on the validation set during training is saved and used as the final set of weights. To facilitate learning for the MLPs, all input and output data is linearly scaled between [-0.6, 0.6] using the maxima and minima from training and validation data, and is presented to the MLP randomly without replacement.

Each MLP is initialised 40 times to account for randomised starting weights and to provide an adequate sample to estimate the distribution of the forecast errors in order to conduct the statistical tests. The MLP initialisation with the lowest sMAPE on the validation dataset is selected to predict all values of the test data.

D. Statistical Benchmark Methods

Any empirical evaluation of time series methods requires the comparison of their accuracy with established statistical benchmark methods, in order to assess the increase in accuracy and its contribution to forecasting research (which is often overlooked in NN experiments [2]). We compare the accuracy of MLPs with different input vectors against a set of statistical benchmark models for level and seasonal time series (due to the absence of trends), including the Naive level, Naive season, single Exponential Smoothing (EXSM) and seasonal EXSM. The method of Naive level,

$$\widehat{x}_{t+h|t} = x_{t-1} \quad , \qquad (3)$$

assumes that the forecast $\widehat{x}_{t+h|t}$ for period $t+h$ will be equal to the last observation x_t. For time series with seasonality of length s, the seasonal Naive method computes a forecast $\widehat{x}_{t+h|t}$ equal to the last observation x one season $t+h-s$ ago, with s depending on the seasonality inherent in the time series frequency [35]:

$$\widehat{x}_{t+h|t} = x_{t+h-s} \quad . \qquad (4)$$

The benchmark methods of EXSM with optimised parameters for stationary and seasonal time series are well established, due to their proven track record in univariate time series forecasting [36]. Model selection of the EXSM method is conducted based on the identified time series components [37]. Depending on the level of seasonality, different benchmarks are computed: For daily time series three naive models are used: Naive level, seasonal Naive with s=7 for weekly seasonality, and seasonal Naive with s=189 using the seasonal lag identified from the Euclidean distance algorithm. In analogy, three EXSM Variants are evaluated: one single EXSM without seasonality, seasonal EXSM with s=7 and with s=189. For weekly and monthly time series of different seasonal length the Naive level and single EXSM methods are evaluated.

IV. RESULTS

E. Results on individual time series

Table II provides the SMAPE errors on the test dataset for the best MLPs, selected as the candidate with the lowest SMPAE validation error, and the statistical benchmark methods. The best MLP candidate with a PACF Yule input vector outperforms all statistical benchmarks on daily and weekly time frequency, but not on monthly time series where the statistical benchmark of single EXSM outperforms all other methods.

Examining table II reveals that the addition of the seasonal lag identified through the Euclidean distance approach (-S suffix models) in the input vector affects positively the accuracy of both the ANN and the benchmarks. Furthermore, for the daily time series the *ACF-Yule* and *ACF-Yule-S*

TABLE II
TEST SUBSET SMAPE

	Time Series Frequency		
MLP Models	**Daily**	**Weekly**	**Monthly**
PACF-Yule	0.3244	*0.0717*⅄	*0.2545*
PACF Burg	0.3882	0.0728	*0.2545*
ACF-PACF-Yule	1.3439	*0.0717*⅄	*0.2545*
Stepwise Regression	*0.2800*⅄	-	-
PACF Yule -S	**0.2674**⅄	**0.0515**⅄	0.2308
PACF Burg-S	0.2705	0.0843	0.2308
ACF-PACF-Yule-S	1.3439	**0.0515**⅄	0.2308
Stepwise Regression-S	0.2800⅄	-	-
Friedman Test	0.000**	0.000**	0.106
Benchmarks			
Naive (s=1)	0.6610	0.1415	0.2307
Naive (s=7)	0.4644	-	-
Naive (s=189)	0.2883	-	-
EXSM (s=1)	-	0.1264	**0.1680**
EXSM (s=7)	0.2806	-	-
EXSM (s=189)	0.3313	-	-

lowest validation error in *italics*, lowest test error (in table) in **bold;** ** = significant Friedman-test at the 0.01 level; * = significant Friedman-test at 0.05 level; ⅄ = no significant differences by Nemenyi-test at 0.05 level; no * / ⅄ = Friedman-test / Nemenyi-test insignificant models, which have 189 inputs, do not perform well. One explanation can be that the degrees of freedom are so high that the training of the network is no longer efficient. The more parsimonious models perform much better.

To establish the significance of the results beyond a comparison of the best method, we conduct nonparametric statistical tests of significance between the error distributions of the different methods. First, we employ the Friedman test to identify significant differences within groups of the MLP models. Once the Friedman test has established significant differences of the input-vector candidates, the Nemenyi test is employed at a 0.05 significance level in order to identify which models do not have significant differences within that group. The test-results are indicated in table II. On monthly time series, no MLP model shows significantly different errors, and all MLP candidates are outperformed by both Naive and EXSM benchmarks. On weekly time series the *PACF-Yule*, *PACF-Yule-S,* *ACF-PACF-Yule* and *ACF-PACF-Yule-S* input vector methodologies perform identical, showing that the inclusion of extra inputs with a maximum lag-length offer no statistically significant improvements, in spite of the slightly better results of *ACF-PACF-Yule* and *ACF-PACF-Yule-S*. For daily time series *Stepwise Regression*, *PACF-Yule-S* and *Stepwise-Regression-S* outperform all other MLPs and benchmarks, demonstrating the superiority of the additional maximum lag from the seasonal diagram for daily data. The statistical test indicate no significant differences between these methods, but significant differences to the conventional input-vector methodologies without '*S*'.

This indicates that MLPs with different input vectors outperform statistical benchmarks on daily and weekly time series of high-frequency, while they underperform on low-frequency data. As the data frequency increases, and more autoregressive information is captured in the models, MLPs achieve a better accuracy in comparison to the benchmarks. Furthermore, there is evidence that the Euclidean distance minimisation approach offers selected improvements in accuracy across all low and high frequency domains.

F. Results from a bottom-up comparison

Using the best MLP for each time series in the validation subset (table II) we create fixed-origin forecasts for 84 days, 8 weeks and 2 months for each time series frequency respectively. In order to facilitate a valid comparison, forecast values are aggregated in a bottom-up manner into time-buckets of lower frequency. For example, we employ a daily forecasting model using daily data. The 84 daily forecast values are aggregated across the forecasting horizon to form forecasts for 8 calendar weeks and 2 months by mapping days to weeks and months. We then estimate the accuracy of these weekly and monthly aggregated forecasts from the weekly and monthly actuals, and compare it to forecasting directly with a weekly model built on using weekly data. Weekly forecasts are aggregated to monthly forecasts accordingly. The results using SMAPE errors are provided in table III.

TABLE III
SMAPEs of Bottom-up Aggregated Forecasts

	Model used to create forecasts		
Time Series	Daily	Weekly	Monthly
Daily time series	**0.2800**		
Weekly time series	0.1160	**0.0743**	-
Monthly time series	**0.0828**	0.0858	0.1853

For daily time series no bottom-up aggregation is feasible. For weekly time series the bottom-up approach, forecasting on a daily level and aggregating to weekly forecasts yields higher forecasts errors of 11.60% in comparison to 7.43% when using a weekly forecasting model directly.

However, for monthly time series the forecasts conducted on a higher time series frequency of either daily or weekly level yield lower forecast errors of 8.28% and 8.58% respectively, in contrast to 18.53% when building conventional forecasting models using monthly data. This discrepancy may be caused by the presence of short termed calendar effects (e.g. Christmas) in the test set, which may be easily captured and extrapolated on higher time series frequencies of daily and weekly data than on monthly data.

Higher frequency data can provide extra detail which may be lost in the lower frequencies, that aids in the creation of better forecasts, as the bottom-up comparison in table III indicates. As a consequence, one may consider forecasting on higher frequency data even if our decision domain is on a lower time series frequency. This further raises the importance of robust modelling of MLPs on high frequency data, in particular in the light of calendar effects and outliers.

III. Discussion

A. Computational resources

In modelling high-frequency time series we identified a number of particular challenges that warrant discussion to facilitate further research. A fundamental characteristic of high frequency data – for a given time span of history – are large datasets. In our preceding experiments, the daily time series is 700% longer than the weekly time series and 3000% longer than monthly time series.

Due to the increased size of the datasets, modelling MLPs for high frequency data require additional computational resources. In our experiments an identical methodology was used to find the number of the hidden units for a given input vector across all three frequency domains, so that all differences in processing time were solely caused by the amount of data resulting from the different time frequencies. The processing times for the topology-search are provided in table IV indicating that specifying topologies to forecast daily data requires 371.1% more computational time than for monthly data.

TABLE IV
Computational time for Topology Search in seconds

Time series	Computational Time (in seconds)	% increase of time on monthly data
Daily	2181	371.1%
Weekly	554	19.7%
Monthly	463	-

All experiments to identify the number of hidden units were conducted for each time series using the software 'Intelligent Forecaster' on an IntelCore2 T7500 processor with 2.2Ghz, 3GB memory using a 32-bit Windows Vista. Valid and reliable experiments require large scale experiments. Simulations on high-frequency data will require substantial computational resources and the development of robust methodologies to specify the input vector for NN modelling.

B. Impact of sample size on statistical test

In addition to the increase in computational time required for training and testing of NN for high-frequency time series, the increased size of the datasets creates further challenges.

In particular, the increased length of the time series impacts the validity of statistical tests required e.g. for input variable selection, by lowering the confidence limits and hence tightening the confidence intervals. Fig. 4 illustrates the positive correlation of the tightness of the confidence intervals of the ACF/PACF with the sample size.

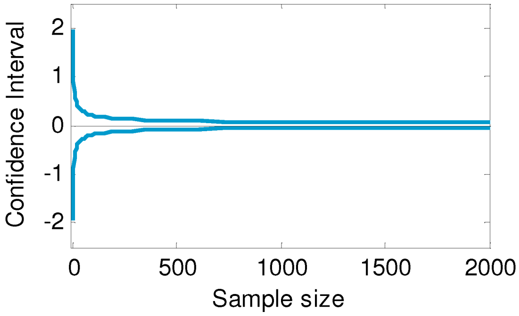

Fig.4. Effect of sample size on confidence interval

As the individual autocorrelations and partial autocorrelations of a time series will exhibit a constant magnitude, this results in more lags of the ACF and PACF becoming statistically significant. After some size of the dataset, the confidence intervals become so tight that nearly every lag becomes significant, an effect that would equally hold for the test of statistical significance used in stepwise regression. As a result, the length of the input vector would rise drastically with the magnitude of the dataset.

To exemplify the effect of sample size while controlling for effects of the information content, we create a synthetic time series of 120 and 1200 observations, the later being ten replications of the first sample. The results for the two PACFs calculated on short low frequency dataset A and the large high-frequency dataset B are provided in figure 5.

It is evident that the ACF of the shorter, low-frequency time series of 120 observations has far less significant lags than the ACF of the second sample, which uses 10 times more observations to represent the increased data of a high-frequency time series with a consistent pattern. This effect is equally apparent in the specified input-vectors of table I.

As a result, the methodologies based upon statistical test would construct non-parsimonious models that depend not on the structure of the data generating process, but merely the sample size. In addition, the impact of sample size on

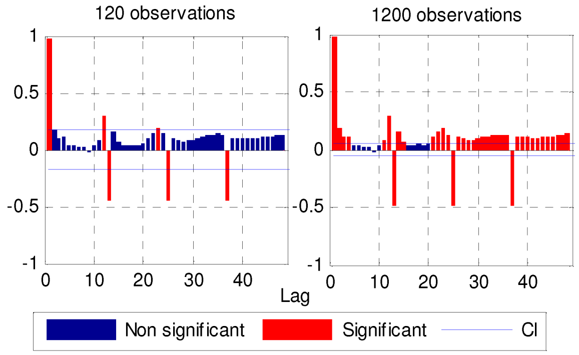

Fig.5. PACF plots of a short (a) and a long sample of a time series (b)

developed for low-frequency data for high-frequency time series despite similar time series patterns. Here additional research is needed to explore corrections to conventional methodologies, in order to extend the use of statistical test as filters in the modelling process to high frequency data.

IV. Conclusions

We evaluate different methodologies to specify the input vector of MLPs across a single time series of increasing time frequency. The experiments indicate that MLPs are well suited to predict high-frequency data of weekly and daily observations and outperform established statistical benchmark methods, while they fail to outperform on low-frequency data of monthly observations.

As a consequence, we provide evidence that NN may be better suited to forecast high frequency data rather than the low-frequency data stemming from the popular M1-, M3-or NN3-forecasting competitions on which they are routinely evaluated in the academic forecasting domains. This may provide an initial explanation of the apparent gap between their limited merit in empirical evaluations and academic competitions using low frequency data, and their corporate success in applications of electrical load forecasting which routinely employs high-frequency data. These findings are further supported by external evidence in comparing the increasing performance of contenders of computational intelligence from the monthly NN3-competition to the daily NN5-competition in comparison to a consistent statistical benchmark method (see the competition website at www.neural-forecasting-competition.com for details).

Our experiments further identify a number of challenges in the modelling process of MLPs for high- and low-frequency data, associated with the dataset size, the available statistical tools, the optimisation routines, outlier identification, computational needs and modelling problems associated with time aggregation and calendar effects. While our analysis offers mere initial solutions to these problems, we consider the identification of these challenges a valuable contribution as they must be resolved to establish NN as a valid and reliable method to routinely forecast low- as well as high-frequency data. The initial results – despite their limited reliability stemming only from a single time series – may facilitate revisions of existing modelling approaches employed for low frequency data in management science, and also to serve as a starting point to for the development of a unified methodology to accurately forecast high- as well as low-frequency data with MLPs. At the same time, experiments must find a way to control for the increased amount of data available for MLP training, which may create interaction effects with the architecture, capacity and accuracy of a NN on datasets of similar data generating processes. Furthermore, experiments must be extended towards causal models using external explanatory variables, although the challenges and costs of additional data acquisition or even prediction of uncontrollable, exogenous variables often warrant the use of mere univariate time series models. Ideally, such methodologies should scale equally well towards additional time series of explanatory input-variables as towards additional data points. Possibly more important still, issues of understanding the models derived from selected input variables in order to infer properties of the data generating process require consideration and may benefit such methodologies with increased interpretability.

In the future, the analysis must be extended to additional time series, time series of different patterns including multiple seasonality and trends, and to additional methodologies of input vector specification to provide a coherent, valid and reliable picture of the relative performance of NN on high- and low-frequency data.

References

[1] G. Q. 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.

[2] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *Journal of Forecasting*, vol. 17, pp. 481-495, 1998.

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

[4] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers & Operations Research*, vol. 28, pp. 381-396, 2001.

[5] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers & Operations Research*, vol. 28, pp. 1183-1202, 2001.

[6] H. S. Hippert, D. W. Bunn, and R. C. Souza, "Large neural networks for electricity load forecasting: Are they overfitted?," *International Journal of Forecasting*, vol. 21, pp. 425-434, 2005.

[7] J. W. Taylor, L. M. de Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, pp. 1-16, 2006.

[8] G. A. Darbellay and M. Slama, "Forecasting the short-term demand for electricity - Do neural networks stand a better chance?," *International Journal of Forecasting*, vol. 16, pp. 71-83, 2000.

[9] M. Cottrell, B. Girard, and P. Rousset, "Forecasting of curves using a Kohonen classification," *Journal of Forecasting*, vol. 17, pp. 429-439, 1998.

[10] H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, pp. 253-261, 2001.

[11] M. S. Dougherty and M. R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *International Journal of Forecasting*, vol. 13, pp. 21-31, 1997.

[12] H. Amilon, "A neural network versus Black-Scholes: A comparison of pricing and hedging performances," *Journal of Forecasting*, vol. 22, pp. 317-335, 2003.

[13] Q. Cao, K. B. Leggio, and M. J. Schniederjans, "A comparison between Fama and French's model and artificial neural networks in predicting the Chinese stock market," *Computers & Operations Research*, vol. 32, pp. 2499-2512, 2005.

[14] K. Lam and K. C. Lam, "Forecasting for the generation of trading signals in financial markets," *Journal of Forecasting*, vol. 19, pp. 39-52, 2000.

[15] N. Gradojevic and J. Yang, "Non-linear, non-parametric, non-fundamental exchange rate forecasting," *Journal of Forecasting*, vol. 25, pp. 227-245, 2006.

[16] R. F. Engle, "The econometrics of ultra-high-frequency data," *Econometrica*, vol. 68, pp. 1-22, 2000.

[17] C. W. J. Granger, "Extracting information from mega-panels and high-frequency data," *Statistica Neerlandica*, vol. 52, pp. 258-272, 1998.

[18] I. S. Markham and T. R. Rakes, "The effect of sample size and variability of data on the comparative performance of artificial neural networks and regression," *Computers & Operations Research*, vol. 25, pp. 251-263, 1998.

[19] M. Y. Hu, G. Q. Zhang, C. Z. Jiang, and B. E. Patuwo, "A cross-validation analysis of neural network out-of-sample performance in exchange rate forecasting," *Decision Sciences*, vol. 30, pp. 197-216, 1999.

[20] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, pp. 251-257, 1991.

[21] M. Qi and G. P. Zhang, "An investigation of model selection criteria for neural network time series forecasting," *European Journal of Operational Research*, vol. 132, pp. 666-680, 2001.

[22] S. F. Crone and N. Kourentzes, "Input variable selection for time series prediction with neural networks-an evaluation of visual, autocorrelation and spectral analysis for varying seasonality," presented at European Symposium on Time Series Prediction, Espoo, Finland, 2007.

[23] G. Lachtermacher and J. D. Fuller, "Backpropagation in Time-Series Forecasting," *Journal of Forecasting*, vol. 14, pp. 381-393, 1995.

[24] U. Anders, O. Korn, and C. Schmitt, "Improving the pricing of options: A neural network approach," *Journal of Forecasting*, vol. 17, pp. 369-388, 1998.

[25] B. Curry, "Neural networks and seasonality: Some technical considerations," *European Journal of Operational Research*, vol. 179, pp. 267-274, 2007.

[26] N. R. Swanson and H. White, "Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models," *International Journal of Forecasting*, vol. 13, pp. 439-461, 1997.

[27] M. Qi and G. S. Maddala, "Economic factors and the stock market: A new perspective," *Journal of Forecasting*, vol. 18, pp. 151-166, 1999.

[28] C. M. Dahl and S. Hylleberg, "Flexible regression models and relative forecast performance," *International Journal of Forecasting*, vol. 20, pp. 201-217, 2004.

[29] M. Ghiassi, H. Saidane, and D. K. Zimbra, "A dynamic artificial neural network model for forecasting time series events," *International Journal of Forecasting*, vol. 21, pp. 341-362, 2005.

[30] B. D. McCullough, "Algorithm choice for (partial) autocorrelation functions," *Journal of Economic and Social Measurement*, vol. 24, pp. 265-278, 1998.

[31] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, pp. 509-515, 2000.

[32] D. J. Hand, "Data mining - New challenges for statisticians," *Social Science Computer Review*, vol. 18, pp. 442-449, 2000.

[33] L. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International Journal of Forecasting*, vol. 16, pp. 437-450, 2000.

[34] D. Janez and ar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1-30, 2006.

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

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

[37] E. J. Gardner, "Exponential smoothing: The state of the art--Part II," *International Journal of Forecasting*, vol. 22, pp. 637-666, 2006.