Forecasting high-frequency Time Series with Neural Networks - an analysis of modelling challenges from increasing data frequency

Nikolaos Kourentzes and Sven F. Crone

Prior research in forecasting economic time series with artificial Neural Networks (NN) has provided only inconsistent evidence on their predictive accuracy. In management science, NN were routinely evaluated on a set of well established benchmark time series of monthly, quarterly or annual frequency. In contrast, NN are accepted as a potent method for electrical load using time series that essentially display similar archetypical patterns of seasonality, trends, level shifts, outliers and calendar effects, but are of higher complexity and frequency. While this discrepancy has been attributed to the lack of a reliable methodology to determine the model parameters, recent research originating from econometrics and finance has indicated that high frequency data may pose different modelling problems compared to their low frequency counterpart and may hence benefit the use of dissimilar methods. This analysis aims to identify and contrast the challenges in modelling NN for low and high frequency data in order to develop a methodology tailored to the properties of the dataset. We conduct a set of experiments in three different frequency domains of daily, weekly and monthly empirical data of the same time series of cash machine withdrawals, using a consistent modelling procedure. The comparison against the naive model and exponential smoothing family models provides evidence that NN are suitable to predict high frequency data. Our analysis identifies a set of problems in modelling NN that arise in high frequency domain, mainly in specifying the input vector. To address these problems a different modelling approach is required between the low and high frequency data. Identifying these problems provides a starting point for the development of a unified methodology to forecast high frequency data with NN and facilitate revisions of the NN modelling approaches employed for low frequency data in management science.

I. Introduction

THERE has been an increased interest in artificial neural networks (ANN) research and applications in forecasting over the last years [1]. ANN have been successfully applied in several forecasting problems [2, 3] demonstrating their abilities both in practice [4] and in academic literature [5, 6]. In a literature survey conducted by the authors the vast majority of business forecasting papers (74 out of 102) make use of low frequency data, i.e. weekly, monthly, quarterly, annual, etc. However, there is an increasing interest in shorter time intervals time series, with examples from domains like electricity forecasting [7-9], traffic predictions [10, 11], finance [12-14] and macroeconomics [15]. On the business forecasting domain what constitutes low and high frequency data is not strictly defined and is relevant to the common practice, techniques and computational resources. Time series of daily or shorter time intervals are accepted as high frequency data [16]. It is argued in literature that high frequency data pose a new set of forecasting problems, making conventional methods inappropriate [17] and demanding new approaches [7]. There are indications that ANN can perform well on high frequency data due to the properties of the dataset [18, 19] which is supported by empirical evidence [4]. The aim of this study is to explore what modelling challenges this increase in the data frequency causes for ANN. This is done by using a time series from the NN5 competition, which is modelled in monthly, weekly and daily time granularity. This way it is explored how the transition from low to high frequency data affects the modelling process and what problems arise. The ANN are compared against benchmarks in each frequency domain and furthermore a bottom-up accuracy comparison is performed to evaluate the effect in accuracy of using higher frequency time series.

Section II presents the forecasting models that are used in this analysis. Section III provides information on the time series and the experimental design. Section IV discusses the results in each different frequency domain and across, using a bottom-up comparison. In section V the identified modelling challenges are discussed. Section VI concludes this analysis and indicates potential further research.

Sven F. Crone is a lecturer in the Management Science Department, Lancaster University, Lancaster, LA1 4YX, UK. (e-mail: s.crone@lancaster.ac.uk).

Nikolaos Kourentzes is a PhD student at the Management Science Department, Lancaster University, Lancaster, LA1 4YX, UK. (e-mail: nikolaos@kourentzes.com).

II. Methods

A. Artificial Neural Networks

For this analysis a basic multilayer perceptron (MLP) will be used, which is the most commonly employed form of ANN [1]. The advantage of neural networks is that they can flexibly model nonlinear relationships without any prior assumptions about the underlying data generation process [20]. In univariate forecasting MLP is used as a regression model, capable of using as inputs a set of lagged observations of the time series to predict its next value [21]. Data are presented to the network as a sliding window over the time series history. The neural network tries to learn the underlying data generation process during training so that valid forecasts are made when new input values are provided [22]. In this analysis single hidden layer neural networks are used, based on the proof of universal approximation [23]. The general function of these networks is given in (1).

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

$X = [x_0, x_1, \ldots, x_n]$ is the vector of the lagged observations (inputs) of the time series and $w = (\beta, \gamma)$ are the network weights. I and H are the number of input and hidden units in the network and g(.) is a non-linear transfer function [24]. For this analysis the hyperbolic tangent function is used.

How to select the input vector of a MLP and the number of hidden units in the hidden layer remains debatable in research [1]. For this analysis four different methodologies to select the input vector are used. Neural networks are autoregressive model, as it is seen in (1); hence the partial autocorrelation function can be used to identify significant time lags in a time series. These time lags can be used as the input vector in an ANN model. This approach has been used in literature with promising results [22, 25]. The standard approach to calculating the PACF is the Yule-Walker algorithm, which estimates the true PACF by minimising the forward regression error in the least squares sense. Research suggests that Burg algorithm for calculating the PACF, which minimises both the forward and backward error, provides more accurate estimation of the autoregressive structure of the time series [26]. Both approaches are used in this analysis. An expansion of these methods is to use the ACF in addition to PACF, as suggested in [22]. An even more popular approach in literature is to use a stepwise linear regression to identify the significant time lags in a time series and use those as the input vector [27-29], which is also employed in this analysis. Under this approach a stepwise model is fitted to the data and the significant time lags are used as inputs to the ANN. To select the correct number of hidden units the most widely used approach to find the best number is through simulations [1]. A MLP is trained using different number of hidden units and the most accurate indicates the correct number of hidden units.

B. *Benchmarks*

A comparison with benchmark models is necessary to strengthen the validity and the contribution of any analysis in forecasting research, which is often overlooked in ANN literature [2]. A set of benchmark models is used. The simplest is the naive forecasting model, which assumes that the forecast will be equal to the last observation. When there is seasonal information a seasonal naive model is employed, which assumes that the forecast is the same as the last seasonal observation [30]. Another well established set of models that are proven to be very robust are the exponential smoothing family models [31]. This is also used as a benchmark. The selection of the appropriate exponential smoothing model follows the standard approach, i.e. the components of the time series dictate which form of exponential smoothing should be used [32].

III. Experimental Design

A. *Time series*

The time series used in this analysis originates from the NN5 competition dataset, which contains ATM money withdrawal information. In the full 111 time series competition dataset the selected time series is dubbed as NN5-035. This time series is daily and starts from the 18th of March 1996 and ends at the 22nd of March 1998. To run the experiments in lower frequency time series, i.e. weekly and monthly, the time series needs to be aggregated. This will be done be summing the daily observations in calendar weeks and months respectively. Summation is preferred to averaging, since the later would smooth the time series. To avoid the introduction of artificial outliers the first and last incomplete months are trimmed, leaving the time series to exactly 23 months or 699 days, just below two full years of data. In the trimmed time series there are 14 missing values. These are imputed by the average of the neighbouring observations. A plot of the time series with the trimmed and the missing values is provided in fig. 1. The processed time series is then aggregated in a weekly and a monthly time series (fig. 2).

A visual inspection of fig. 1 and fig.2 does not provide

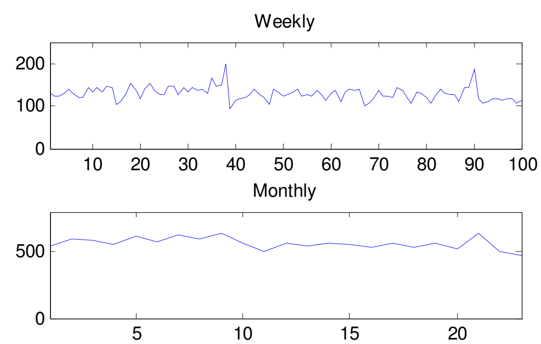

Fig. 2. NN5-035 weekly and monthly aggregated plots

any hints of a trend and Phillips-Perron test provides the same result for all three time series. To identify any possible seasonality several different tools were used, these being

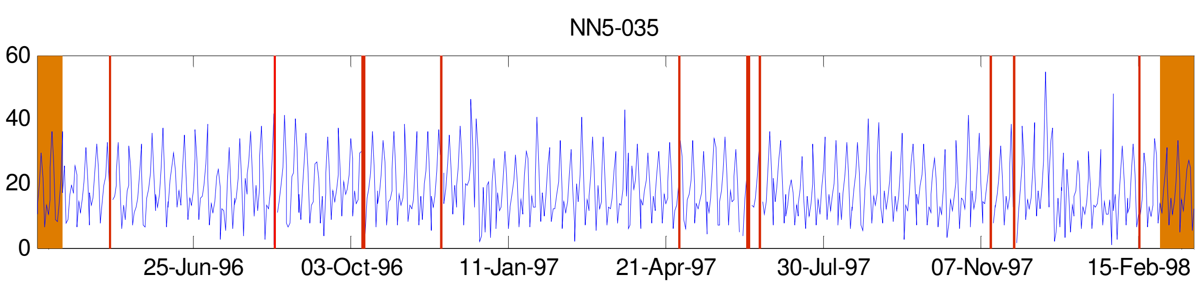

Fig. 1. NN5-035 plot, highlighting the observations that are trimmed and the missing values.

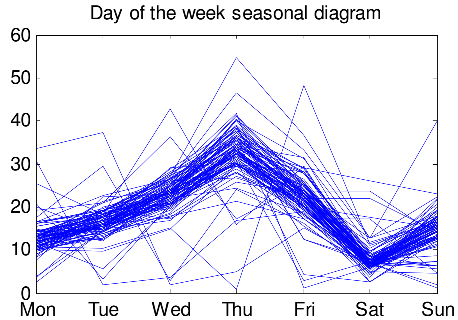

Fig.3. Seasonal diagram for the daily time series indicating a strong day of the week pattern

visual inspection, the ACF/PACF and the periodogram. The seasonal plot for the daily time series is provided in fig. 3 and indicates a strong day of the week seasonal pattern. The periodogram provides the same information.

Using the ACF/PACF for the daily time series the weekly pattern is found, but in addition to that a yearly cycle can be identified. If the untrimmed original NN5-035 time series is used, which is longer than 2 full years, the annual cycle is even more evident. As for the weekly and the yearly time series a yearly pattern can be identified through the ACF/PACF analysis. Evidently there is a yearly pattern that is difficult to model due to the lack of enough data.

All three time series have some apparent outliers. Examining the daily time series, unordinary consumer behaviour can be observed the last 1.5 weeks of each year (18 of December until 31 of December). This is associated with the effect of the Christmas and the New Year's Eve. These are marked with an integer dummy variable. Examining the weekly aggregated time series the last two weeks of the year show an aberrant behaviour, which is mirrored in the last month for the monthly time series. The dummy time series will be used during the training as an additional input for the neural network models. Specifically for the monthly time series an artificial outlier is introduced by the aggregation. February having fewer days is constantly below the level of the other months, which is also coded.

B. Experiment set up

Most of the experiments parameters originate from the NN5 competition. The forecasting horizon is 56 days, 8 weeks and 2 months for each time relevant time series frequency. These are selected like that to facilitate a bottom-up comparison of the accuracy across different frequencies.

The error measure used for comparison of the models is the symmetric MAPE, as set by the competition. The SMAPE calculates the symmetric absolute error in percent between the actuals X and the forecast F across all observations t of the test set of size n as shown in (2).

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

The validation and the test set are both 84 days, 12 weeks and 3 months for each time series frequency. The model selection is performed by selecting the model with the minimum error in the validation set. The evaluation of the models is done using rolling origin evaluation, being superior than the widely used fixed origin evaluation [33]. The comparison of the competing neural network models is done using the nonparametric Friedman test and the Nemenyi test. This combination is suggested in literature as adequate to evaluate nonparametric models, like neural networks, without the need to relax assumptions of ANOVA or similar parametric tests [34]. All tests are performed at 5% significance level.

C. Neural Network Models

All the neural networks used have a similar setup with the exception of the input vector and the number of hidden units. The neural networks are initialised 40 times, to provide an adequate error distribution and enough sample for the statistical tests. Gradient descent backpropagation is used for the training. The learning rate is set to 0.5 with a cooling factor per epoch of 0.01. Momentum is set to 0.4 and the networks are trained for 1000 epochs or until an early stopping criterion is satisfied. For the early stopping criterion the mean squared error is evaluated every epoch, after the first hundred epochs. The data are presented to the neural networks scaled between [-0.6, 0.6] using random sampling without replacement. All neural networks have a single output with an identity function.

The number of hidden units is re-specified for every time series. After simulation experiments the ideal number of hidden units is found to be 8, 5 and 9 for the daily, weekly and monthly time series respectively. The possible options that were evaluated are from 1 to 12 hidden units.

For the input vector four different methods are used to model the ANN automatically for each time series. These are

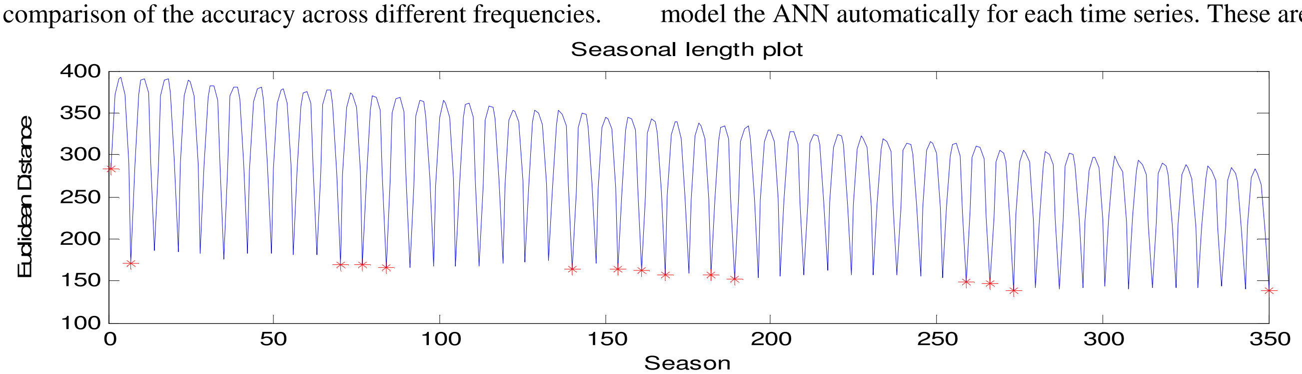

Fig. 4. Seasonal length plotted against the Euclidean distance between seasons. The asterisks signify the local minima.

TABLE I
SUMMARY OF NEURAL NETWORKS

Frequency	Daily			Weekly			Monthly		
Model	**Input Vector**	**#**	**Hidden Units**	**Input Vector**	**#**	**Hidden Units**	**Input Vector**	**#**	**Hidden Units**
Yule	1,3-9,11,13-16,21,28-29,36,65,108	19	8	1-2	2	5	1	1	9
Burg	1,3,5-9,11,13-15,20,22,29,36	15	8	1	1	5	1	1	9
ACF-Yule	1-189	189	8	1-2	2	5	1	1	9
Regression	1,7,35,56,83-84,99,169,174,182,189	12	8	-	-	-	-	-	-
YuleS	1,3-9,11,13-16,21,28-29,36,65,108,189	17	8	1-2,25	3	5	1,7	2	9
BurgS	1,3,5-9,11,13-15,20,22,29,36,189	6	8	1,25	2	5	1,7	2	9
ACF-YuleS	1-189	189	8	1-2,25	3	5	1,7	2	9
RegressionS	1,7,35,56,83-84,99,169,174,182,189	12	8	-	-	-	-	-	-

The input vector contains the time lags that are used as inputs. For example 1, 25 means that t-1 and t-25 are used.

PACF calculated by both the Yule-Walker and Burg algorithms, the combination of ACF and PACF and stepwise linear regression as described in section II.A. A problem that arises here is related to how many lags should one evaluate to include in the input vector. This has not been explored in literature, with the exception of a sole reference for low frequency data [35]. The common practice involves trial and error approaches or setting an arbitrary number of lags. This approach may be feasible for low frequency data, but for high frequency it is not, as the options become too many, due to the large sample size. A method that derives this information from the time series is necessary. A method based on the Euclidean distance is proposed. Assuming no prior seasonal information the time series is split into "seasons" of different length and the Euclidean distance between the seasons is calculated. The seasonal length that minimises the Euclidean distance essentially gives the minimum possible deviation (in squared error terms) of the seasons in a seasonal plot, thus providing the best possible seasonality. Apparently for each seasonal length that is tested an Euclidean distance is calculated and these can be plotted showing all the local (global) minima in terms of distance (fig. 4). The local minima in this sense are the minimum distances found as the seasonality increases. The seasonality is additional modelling information that is argued as potentially necessary for the input vector [36]. From fig. 4 it can be seen that multiples of the weekly seasonality result in smaller distances. It is suggested that this can act as a guideline how many seasons of data to evaluate to identify the input vector. The compromise that must be made is to use a local minimum of the Euclidean distance that will leave enough data for the training of the ANN.

These four input vector identification methods are used in two different implementations. Firstly, the time series are used to identify the input vector without any pre-processing. The second implementation is the same with the difference that the last "seasonal" lag identified using the Euclidean distance is forced in the input vector. The models are named after the input vector methodology used and for the second implementation variant an "S" suffix is added. For the weekly and the monthly time series the regression based models cannot identify any significant lags. In these cases a neural network is not trained. The neural networks used are summarised in table I.

D. *Benchmarks*

For both the weekly and the monthly time series the naive and the simple exponential smoothing models are used as benchmarks. For the daily time series three naive models are used. The first naive uses the last observation, the second uses the weekly seasonal lag and the third uses the Euclidean distance "seasonal" lag. Two seasonal exponential smoothing models are used. The first uses a weekly seasonality and the second, like the naive model, the Euclidean distance identified "season".

IV. RESULTS

A. *Individual time series*

The SMAPE of the best ANNs, according to validation set errors, is provided in table II. The same table provides the benchmarks errors as well. The daily and the weekly ANN models outperform all the benchmarks, but the same is not true for the monthly ANN models. 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-YuleS* 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.

In table III statistical test that compare the different ANN models are provided. The Friedman test can be used to identify significant differences within groups of ANN models. Once the Friedman test establishes differences, the Nemenyi test can be used to identify which models do not have significant differences in that group of models. Starting from the monthly time series the tests show that no models are significantly different and all outperformed by the benchmarks. For the weekly time series the *Yule*, *YuleS,* *ACF-Yule* and *ACF-YuleS* perform identical showing that the inclusion of the extra input does not offer statistically significant improvements, in spite of the improvement observed in table II. For the daily time series *Regression*, *YuleS* and *RegressionS* are not significantly different, showing that the marginal difference between *Regression*

TABLE II
TEST SUBSET SMAPE

	Frequency / Model	Daily	Weekly	Monthly
ANN Models	Yule	0.3244	**0.0717**	**0.2545**
	Burg	0.3882	0.0728	**0.2545**
	ACF-Yule	1.3439	**0.0717**	**0.2545**
	Regression	**0.2800**	-	-
	YuleS	0.2674	0.0515	0.2308
	BurgS	0.2705	0.0843	0.2308
	ACF-YuleS	1.3439	0.0515	0.2308
	RegressionS	0.2800	-	-
	Selected Model	Regression(S)	Yule/ACF-Yule	Yule/Burg/ACF-Yule
Benchmarks	Naive-1	0.6610	0.1415	0.2307
	Naive-7	0.4644	-	-
	Naive-189	0.2883	-	-
	EXSM-1	-	0.1264	0.1680
	EXSM-7	0.2806	-	-
	EXSM-189	0.3313	-	-

SMAPE for all the models. The selected ANN models are those that perform best in the validation subset. The numbers following the benchmark models indicate the seasonality, i.e. EXSM-1 is a simple exponential smoothing and EXSM-7 is a seasonal exponential smoothing with a seasonality of 7.

and the benchmarks can be even larger as *YuleS* shows. Interestingly, *BurgS* outperforms the benchmarks in the test set but is significantly worse than the first group of models. This indicates that several different setups of ANN outperform the benchmarks for the daily time series and that for the high frequency time series the *–S* models are significantly better than the other models. Combining the information in tables I, II and III one can see that as the data frequency increases there is more autoregressive information for the models to capture, resulting in better accuracy in comparison to the benchmarks. Furthermore, there is evidence that the Euclidean distance minimisation approach offers improvement in accuracy, especially in the high frequency domain.

B. *Bottom-up comparisons*

Using the best neural network for each time series in the validation subset (table II) forecasts are created for 84 days, 8 weeks and 2 months for each time series respectively. The

TABLE III
STATISTICAL TESTS FOR TEST SUBSET

Frequency	Daily	Weekly	Monthly
Friedman Test	0.000	0.000	**0.106**
Model		nemenyi	
Yule		Same	
Burg			
ACF-Yule		Same	
Regression	Same	-	-
YuleS	Same	Same	
BurgS			
ACF-YuleS		Same	
RegressionS	Same	-	-

The different neural network models are tested against each other using Friedman test and if significant differences are found then the Nemenyi test is employed to explore in more details which models are significantly different. Same here means that there are no statistical differences. All tests are performed at 5% significance level.

TABLE IV
BOTTOM-UP COMPARISON SMAPE

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

The SMAPE produced by the each model for each time series is provided in the above table. For the weekly time series the daily model forecasts are aggregated into calendar weeks. For the monthly time series the daily and the weekly models forecasts are aggregated into the last two calendar months of the test set.

daily forecasts can be aggregated to weekly and monthly forecasts, using calendar information, to compare the accuracy with the weekly and monthly models. Essentially bottom-up forecasts in the dimension of time are created. To keep the experiment complexity controllable this evaluation is a fixed origin evaluation and SMAPE is used. The results are provided in table IV.

For the weekly time series the bottom-up approach does not produce better forecasts. The aggregated daily forecasts are less accurate than the weekly forecasts, for the weekly time series. However when examining the monthly forecasts the more detailed (high frequency) forecasts used, the better the accuracy of the bottom-up forecasts. This discrepancy is caused by the presence of the outliers in the test subset. The daily forecasts behave very well only outside the peak of the Christmas, where the weekly forecasts model the outlier better. This reduced performance in Christmas is evident on weekly level, but on monthly level the normal periods are so many that dampen the effect of the outlier, showing that high detail (frequency) forecasts can be useful. Furthermore, the need for a more sophisticated coding of outliers for high frequency data is revealed, as it will be discussed in the next section.

V. DISCUSSION

A. *Effect of sample size*

Basic characteristic of the high frequency data is large datasets. For this experiment the daily time series is 7 and 30 times longer than the weekly and the monthly time series. This difference in the sample size creates several differences in the three frequency domain even though exactly the same modelling procedure is followed.

From table I one can observe that most of the input vector identification methodologies used behaved differently for the three data frequencies with significant differences between the daily time series and the other two. Higher frequency data can provide extra detail, which is lost in the lower frequencies, that aids in the creation of better forecasts, as the bottom-up comparison indicates (table IV). However this comes at a price. From the conducted experiments it was made apparent that a more sophisticated way to code the outliers is needed, since in high frequency data an “outlier” can last for several observations. Coding through a binary dummy variable is not adequate, since this would essentially code a level change and not an abnormal behaviour. Integer dummies seem to be unable to model the outlier satisfactory,

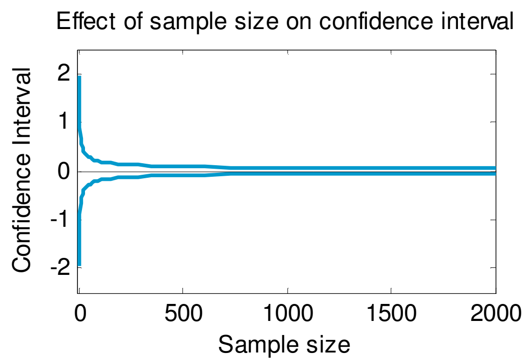

Fig.5. As the sample size increases the confidence interval becomes tighter. After a point it is so tight that nearly everything becomes significant.

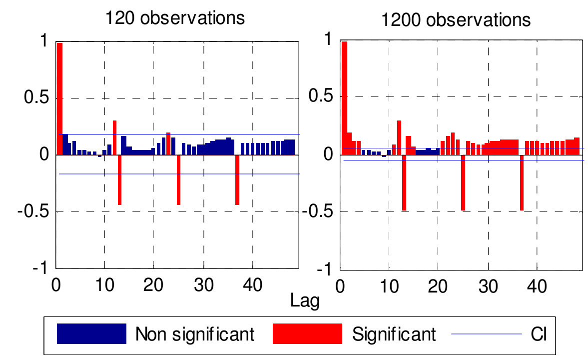

Fig.6. PACF plots of a synthetic time series. The first plot which is calculated using only 120 observations has far less significant lags than the second which uses 10 times more observations.

though their use increases accuracy. This analysis, using a consistent modelling across all frequency levels does not address the need for a new approach to outlier coding.

Another aspect of problems associated with the data size is how the input vector identification methods behave. All the methods used here employ some form of confidence intervals. Focusing to ACF-PACF approaches, the tightness of the confidence intervals is positively correlated with the sample size, as is illustrated in fig. 5.

Although the sample size effects the calculation of ACF/PACF the effect on the confidence intervals is more prominent. Using a synthetic time series, to keep the information content under control, the PACF is calculated for 120 observations and 1200 observations, the later being ten times the first sample. The results are provided in figure 6 and are quite illustrative of the problem, indicating that a "high frequency data" correction is necessary, to avoid overparameterisation of the models in the input vector.

One other problem that derives from the resulting large input vectors is that the degrees of freedom become so high that it is very hard to train (optimise) the neural network. This is evident in tables I and II, where the *ACF-Yule* and *ACF-YuleS* models for the daily time series have as inputs all lags from t-1 to t-189 and the optimiser cannot train the network efficiently, resulting in the worst model in terms of accuracy. In theory the optimiser should be able to assign nearly zero weights to the nonessential inputs, which in practice does not happen due to the high degrees of freedom.

B. Computational resources

High frequency data, due to the large datasets associated require a lot of computational resources. In the performed experiment the same layout was used to find the number of the hidden units for all three frequency domains. The

TABLE V
COMPUTATIONAL TIME IN SECONDS

Time series	Computational Time (seconds)	% more from monthly
Daily	2181	371.1%
Weekly	554	19.7%
Monthly	463	-

The recorded time corresponds to the identical experiments run for each time series to identify the correct number of hidden units. The experiments were run in a Intel(R) Core(TM)2 T7500 processor at 2.2Ghz with 3GB of memory using Windows Vista 32-bit operating system.

difference in the computational time was solely due to the frequency of the data. Inspecting table V one can see that the daily data require 371.1% more time than the monthly data. Large scale simulations will require immense computational resources and given the present computational capabilities parallel computing seems to be the way forward. This also has significant implications for heuristic approaches used to identify the input vector and modelling the ANN. As the frequency of the data increases the required computational resources for simulation studies become very high, demanding a smart modelling approach that will make the most of the information contained in the time series minimising the need to do simulation/run heuristics to find the best setup.

C. Calendar Problems

Increasing the frequency in business data means that one has to take into considerations weeks, days, hours, etc. From weeks and on the aggregation of data can be very challenging, this is because there is no set number of weeks every month/year has. The same is true for days. Furthermore, every four years there is a leap year, which creates seasonal shifts. Aggregation using calendar information is a solution, which is used in these experiments, but as far as the data exploration and input vector identification/coding is concerned most methods are unable to handle the above calendar properties. This needs further research so that a standardised solution can be developed that will allow statistical tools handle this information.

D. Necessary amount of information for model building

The modeller has to decide how many time lags to include in the analysis for the input vector. When seasonality is present this can be a good guideline. In the case of high frequency data it is common to expect multiple seasonalities to be present, like here, where a day of the week and a day of the year pattern exist. The problem becomes even trickier when there is no seasonality. In this analysis an approach using the minimum seasonal Euclidean distance was used, but further research is necessary to reach a consistent mathematical rule to decide how many lags should be evaluated. What is clear for high frequency data is that the number of lags that need to be evaluated is much higher than for the low frequency data, as it is apparent from the input vectors is table I.

VI. Conclusions

The experiments carried out in this analysis aimed at revealing the difficulties involved in modelling high frequency data with neural networks. Neural networks are found to be able to model this kind of data, with promising results, suggesting it is worthwhile to pursue further research on this specific problem. Furthermore, a set of problems associated with the dataset size, the available statistical tools, the optimisation routines, outlier identification, computational needs and modelling problems associated with time aggregation and manipulation were revealed. This analysis offers some initial solutions to these problems, but its main contributions are that it identified the set of problems that need to be addressed before forecasting high frequency data with neural networks becomes a routine procedure and that it gives more evidence that neural networks are a promising approach to high frequency data forecasting.

References

[1] 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, no. 1, 1998, pp. 35-62.

[2] 2. M. Adya, and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *Journal of Forecasting*, vol. 17, no. 5-6, 1998, pp. 481-495.

[3] 3. T. Hill, M. O'Connor, and W. Remus, "Neural network models for time series forecasts," *Management Science*, vol. 42, no. 7, 1996, pp. 1082-1092.

[4] 4. 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, no. 3, 2005, pp. 425-434.

[5] 5. 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, no. 4, 2001, pp. 381-396.

[6] 6. G.P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 12, 2001, pp. 1183-1202.

[7] 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, no. 1, 2006, pp. 1-16.

[8] 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, no. 1, 2000, pp. 71-83.

[9] 9. M. Cottrell, B. Girard, and P. Rousset, "Forecasting of curves using a Kohonen classification," *Journal of Forecasting*, vol. 17, no. 5-6, 1998, pp. 429-439.

[10] 10. H. Dia, "An object-oriented neural network approach to short-term traffic forecasting," *European Journal of Operational Research*, vol. 131, no. 2, 2001, pp. 253-261.

[11] 11. M.S. Dougherty, and M.R. Cobbett, "Short-term inter-urban traffic forecasts using neural networks," *International Journal of Forecasting*, vol. 13, no. 1, 1997, pp. 21-31.

[12] 12. H. Amilon, "A neural network versus Black-Scholes: A comparison of pricing and hedging performances," *Journal of Forecasting*, vol. 22, no. 4, 2003, pp. 317-335.

[13] 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, no. 10, 2005, pp. 2499-2512.

[14] 14. K. Lam, and K.C. Lam, "Forecasting for the generation of trading signals in financial markets," *Journal of Forecasting*, vol. 19, no. 1, 2000, pp. 39-52.

[15] 15. N. Gradojevic, and J. Yang, "Non-linear, non-parametric, non-fundamental exchange rate forecasting," *Journal of Forecasting*, vol. 25, no. 4, 2006, pp. 227-245.

[16] 16. R.F. Engle, "The econometrics of ultra-high-frequency data," *Econometrica*, vol. 68, no. 1, 2000, pp. 1-22.

[17] 17. C.W.J. Granger, "Extracting information from mega-panels and high-frequency data," *Statistica Neerlandica*, vol. 52, no. 3, 1998, pp. 258-272.

[18] 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, no. 4, 1998, pp. 251-263.

[19] 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, no. 1, 1999, pp. 197-216.

[20] 20. 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, no. 3, 2001, pp. 666-680.

[21] 21. 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," *European Symposium on Time Series Prediction*, pp. 195-205.

[22] 22. G. Lachtermacher, and J.D. Fuller, "Backpropagation in Time-Series Forecasting," *Journal of Forecasting*, vol. 14, no. 4, 1995, pp. 381-393.

[23] 23. K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, 1991, pp. 251-257.

[24] 24. U. Anders, O. Korn, and C. Schmitt, "Improving the pricing of options: A neural network approach," *Journal of Forecasting*, vol. 17, no. 5-6, 1998, pp. 369-388.

[25] 25. 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, no. 2, 2005, pp. 341-362.

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

[27] 27. 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, no. 4, 1997, pp. 439-461.

[28] 28. M. Qi, and G.S. Maddala, "Economic factors and the stock market: A new perspective," *Journal of Forecasting*, vol. 18, no. 3, 1999, pp. 151-166.

[29] 29. C.M. Dahl, and S. Hylleberg, "Flexible regression models and relative forecast performance," *International Journal of Forecasting*, vol. 20, no. 2, 2004, pp. 201-217.

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

[31] 31. S. Makridakis, and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, 2000, pp. 451-476.

[32] 32. E.J. Gardner, "Exponential smoothing: The state of the art--Part II," *International Journal of Forecasting*, vol. 22, no. 4, 2006, pp. 637-666.

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

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

[35] 35. S.D. Balkin, and J.K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, no. 4, 2000, pp. 509-515.

[36] 36. B. Curry, "Neural networks and seasonality: Some technical considerations," *European Journal of Operational Research*, vol. 179, no. 1, 2007, pp. 267-274.