Forecasting Seasonal Time Series with Multilayer Perceptrons – an empirical Evaluation of Input Vector Specifications for deterministic Seasonality

Sven F. Crone, Nikolaos Kourentzes

Abstract - **Research in forecasting with Neural Networks (NN) has provided contradictory evidence on their ability to model seasonal time series. Several empirical studies have concluded that time series should be deseasonalised prior to modelling, with contradictory evidence pointing to adequate selection of input vectors. However, the nature of seasonality itself has not been considered: econometric theory suggests that deterministic seasonality and stochastic seasonality need to be modelled differently, with only the latter requiring deseasonalisation. As prior research has failed to take the conditions of the underlying seasonality into consideration, this study explores how deterministic seasonality should be best modelled with NN to achieve accurate and robust forecasts. We consider different forms of modelling seasonality as autoregressive lags, through different encodings of explanatory variables and deseasonalisation. We evaluate the results regarding empirical accuracy and the parsimony of the input vector in order to limit the degrees of freedom, develop robust models and simplify the training of NN. Our findings are consistent with econometric literature, i.e. that no deseasonalisation is required for deterministic seasonality, and contributes to current research in NNs by identifying a more parsimonious coding of seasonality based on seasonal indices.**

I. Introduction

NEURAL networks (NN) are recognised as a potent forecasting tool with research and applications in a wide variety of disciplines [1, 2]. Given their theoretical properties, NNs are universal approximators, permitting them to approximate any measurable function to any desired degree of accuracy (and to generalise on unseen data as well) [3], a desirable property in forecasting. NN have proven their capabilities to forecast linear as well as nonlinear time series of synthetic and empirical data [4-6], and across a wide range of time series frequencies including monthly, weekly, daily data with their distinct data properties [7]. However, they are prone to criticism due to the lack of a sound methodology to determine valid and reliable models in empirical evaluations [8-10]. For instance, no consensus exists on how to select a relevant set of input variables and lags for NNs given the different time series components and properties of the dataset [1]; a recent literature survey identified 72% (out of 95) published papers model NNs based purely on invalid and unreliable trial and error approaches. This has a significant impact on the consistency of NN performance and also hinders our understanding of how to model them [11]. Consequently it remains an important task to empirically evaluate competing methodologies for NN modelling with scientific rigor, in order to derive insight on best practices.

One open research questions currently under discussion [1, 12] is to determine the most efficient and effective modelling approach for NNs on seasonal time series, with the related set of questions whether seasonality should be modelled directly or should be removed by deseasonalising the time series first. Hill et al. [6] showed that NNs using deseasonalised time series from the M1-competition outperformed statistical benchmark models, promising significant improvements in NN performance. Nelson et al. [13] (from the same research group) verified the effect of deseasonalisation using M1-data. They repeated the experiments of Hill et al. without deseasonalising, and provided further evidence that the forecasting performance of NN significantly deteriorate on seasonal data, therefore concluding that deseasonalisation is a necessary step in NN modelling. They postulate that deseasonalising time series allows NNs to focus on learning the trend and the cyclical components; to learn seasonality at the same time would require larger networks, resulting in a larger input vector, which may cause detrimental overfitting. Zhang and Qi [14] reach the same conclusion that deseasonalising helps. However, their research suggest that deseasonalising time series reduces long and dynamic autocorrelation structures that would make the choice of the input vector for NNs more difficult, thus leading to smaller, more parsimonious and hence robust models. Curry [12] examines the ability of NN to model seasonality from a theoretical perspective using data with different properties. His research suggests that NN require adequately long input vectors in order to capture the seasonal effects; ill selected and underspecified input vectors on the other hand can impair the ability of a NN to capture and forecast seasonality, implying that the results by Zhang and Qi can potentially hide input misspecification errors.

However, none of the preceding research studies on NNs distinguishes between the different forms of seasonality, and hence the conditions under which one approach outperforms another. Reflecting upon the statistical literature, both deterministic seasonality and stochastic seasonality (seasonal unit roots) theoretically require different modelling approaches [15-17], which has been ignored in NN literature and the debate on how to model seasonality to date.

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

In this analysis we will argue that an adequate distinction implies a different modelling procedure – both from a theoretical perspective but also supported by the empirical evidence. Modelling deterministic seasonality is impaired by prior deseasonalisation of the time series, so that different modelling practises should be considered. To provide evidence we conduct an empirical evaluation of competing approaches to model seasonality using univariate (autoregressive) time series approaches, the use of external variables and differencing on simulated time series. We find that using a set of dummy variables can significantly improve forecasting accuracy over the current NN modelling 'best practise' suggested by prior research, where seasonality is either removed or modelled using seasonal lags of the time series. Finally, we propose a parsimonious coding based on seasonal indices, which outperforms other candidate models in accuracy while keeping the degrees of freedom to a minimum in order to construct parsimonious NN architectures.

The paper is organised as follows: section II discusses the different types of seasonality from a theoretical perspective. Section III introduces the methods that will be used to model deterministic seasonality, followed by information on the time series and the experimental design in section IV. Section V discusses the results comparing the different approaches to modelling deterministic seasonality, followed by conclusions and further research objectives in section VI.

II. Forecasting Seasonal Time Series

A. *Deterministic and Stochastic Seasonality*

Seasonality is generally defined as a reoccurring time series pattern within the calendar year, where its structure is defined by the frequency of the times series. A time series is said to have deterministic seasonality when its unconditional mean varies with the season and can be represented using seasonal dummy variables,

$$y_t = \mu + \sum_{s=1}^{S} m_s \delta_{st} + z_t \,, \qquad (1)$$

where y_t is the value of the time series at time t, μ is the level of the time series, m_s is the seasonal level shift due to the deterministic seasonality for season s, δ_{st} is the seasonal dummy variable for season s at time t, z_t is a weak stationary stochastic process with zero mean and S is the length of the seasonality. Furthermore, the level of the time series μ can be generalised to include trend.

In contrast to deterministic seasonality, seasonality can also be the result of an integrated autoregressive moving average (ARIMA) process,

$$\phi(L)\Delta_S y_t = \gamma + \theta(L)\varepsilon_\tau \,, \qquad (3)$$

resulting in a stochastic seasonality (or seasonal unit root), where L is the lag operator, Δ_S is the seasonal difference operator, $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ are the coefficients of the autoregressive and moving average process respectively, γ is a drift, and $\varepsilon_t \sim$ i.i.d. with zero mean and standard deviation σ^2. Of course this can be expressed as a SARIMA model. The variance of y_t under the case of deterministic seasonality is constant over t and the seasonal period s, which is not true here. This stochastic seasonal process can be viewed as a seasonal unit root process, i.e. for each s there is a unit root, which in turn requires seasonal differencing. More details about the seasonal unit root process can be found in [15-17].

B. *Modelling NNs for Deterministic Seasonality*

Seasonal information can be included in NNs in a variety of ways, including various forms of explanatory variables. Note that deterministic seasonality as in (1) is defined as a series of seasonal level shifts m_s, which describe the seasonal profile and are constant across time, i.e. $m_s = m_{st}$. Also note that as deterministic seasonality is defined in (1) the $\sum m_s = 0$ over a full season. This implies that with the appropriate transformations of μ and m_s a set of S-1 or S binary seasonal dummy variables can be used to code the seasonality. Furthermore, due to z_t each value of the time series deviates over its respective seasonal mean with a constant variance over both s and t, which means that the deterministic seasonal process forces the observations to remain close to their underlying mean [17]. Modelling (1) with S seasonal dummies and $\mu \neq 0$ using a linear model (e.g linear regression or a linear NN) introduces multicollinearity, hence S-1 dummies must be used [18]. For nonlinear methods such as NNs with only linear transfer functions and $H > 1$ multicollinearity can exist even for S-1 dummies, since the dummies provide input into several hidden nodes. This hinders inference from a NN, but does not necessarily harm its predictive power [1, 18]. As this also holds for the case of nonlinear transfer functions, both modelling using S-1 or S seasonal dummies may be adequate for NN models.

An alternative way to code deterministic seasonality as in (1) is through its trigonometric representation:

$$y_t = \mu + \sum_{k=1}^{S/2}\left[\alpha_k \cos\left(\frac{2\pi kt}{S}\right) + \beta_\kappa \sin\left(\frac{2\pi kt}{S}\right)\right] + z_t \,, \qquad (2)$$

where α_k and β_k create linear combinations of $S/2$ sines and cosines of different frequencies following the ideas of spectral analysis of seasonality. Equations (1) and (2) have μ and z_t expressed as separate components in both cases, allowing separate modelling of seasonality and the remaining time series components [17]. Note that if less than $S/2$ linear combinations of sines and cosines are used the representation of seasonality is imperfect and it is approximated with some error, the size of which is related to the number of combinations used. Deterministic seasonality as expressed in (2) can equally be modelled in NNs using explanatory variables. Note that an alternative is to approximate (2) using less frequencies by increasing the number of hidden nodes H in a network [3].

Following the same procedure, based on the increase of H, NN can approximate seasonal patterns by combining seasonal dummies in a single integer dummy defined as $\delta = [1, 2 \ldots S]$ [19], which is neither permissible nor explored in linear models. Alternatively the set of m_s can be combined to form a series of interval scaled seasonal indices that can be used as an explanatory variable for the NN, with the requirement to predetermine the m_s beforehand.

C. Misspecifying NNs for Deterministic Seasonality

For our analysis it is interesting to examine what occurs if deterministic seasonality is misspecified as a seasonal unit root process. Considering seasonal differences (1) becomes

$$\Delta_S y_t = \Delta_S z_t \,. \tag{4}$$

Essentially in (4) seasonality has been removed, i.e. a deseasonalised form of y_t is modelled. Comparing (1) and (4) we can deduce that it is now impossible to estimate m_s and furthermore $\Delta_S z_t$ is overdifferenced [17]. Therefore, it is preferable to keep deterministic seasonality and model it appropriately.

However, neglecting the properties of the deterministic seasonality NNs are frequently modelled as a misspecified stochastic seasonal unit root process, with the problems implied above. One alternative is to use seasonal integration to remove seasonality and another alternative would be to use an adequate AR structure to model the seasonality as discussed in [12]. Note that much of the debate in literature regarding the deseasonalisation of time series (see section I) falls in the latter two alternatives, which in theory are not advisable for deterministic seasonality. However, for practical applications with small samples it can be shown that it is difficult to distinguish between deterministic and stochastic seasonality [17], therefore these alternatives are still viable options that warrant experimental evaluation.

III. Experimental Design

A. Time Series Data

We employ eight synthetic time series to evaluate the different approaches to model deterministic seasonality using NN. The time series are constructed using as a data generating process the dummy variable representation of deterministic seasonality (1). All time series have $S = 12$, i.e. simulate monthly data, and are 480 observations long. Two different sets of m_s are modelled, reflecting two different seasonal patterns (A & B) derived from retail sales with $\mu = 240$ and z_t is ε_t ~ i.i.d. with $(0, \sigma_j^2)$. These are superimposed with four levels of increasing random noise σ_j^2: for time series without noise $\sigma = 0$ is used (i.e. zero error for all t). For low, medium and high noise we use $\sigma=\{1, 5, 10\}$ respectively. Note that the synthetic time series are constructed in a stricter way than required by (1) in order to create time series in which only the effect of the deterministic seasonal pattern requires modelling, simplifying the specification of the input vector of the NN to focus solely on the effects of the different seasonal coding schemes. For the empirical evaluation all series are split in three equal subsets of ⅓rd training, ⅓rd validation and ⅓rd test data of 160 observations. The first 72 observations of both time series patterns and 4 noise levels are plotted in fig. 1:

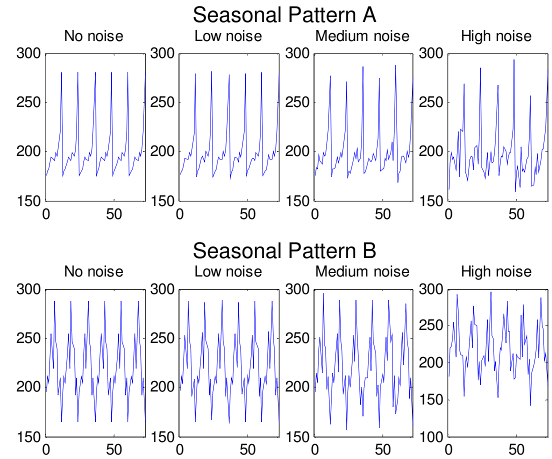

Fig. 1. Plot of the first 72 observations of each synthetic time series.

B. Evaluating Empirical Accuracy

We conduct a fixed horizon, rolling origin evaluation in order to assess the error of forecasting $t+h = 1, .., 12$ months in the future across multiple time origins. This evaluation scheme is preferred because it provides a reliable estimation of the out of sample error [20]. Two established and robust error measures are used: the mean absolute error (MAE) allows a direct comparison of the predictive accuracy and the known noise level by evaluating the absolute deviation of actuals X_t and the forecast F_t for all periods t in the test set. In addition the symmetric mean absolute percent error (sMAPE) is used as a scale independent accuracy measure to facilitate comparisons across time series. The accuracy of the competing NN models is evaluated for statistically significant differences 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 [21]. To compare the NN models against the benchmark the best NN initialisation in the validation set is used.

C. Neural Networks for Time Series Prediction

We limit our evaluation to the multilayer perceptron (MLP), the most widely employed NN architecture [1]. The advantage of MLP 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 [22] without any prior assumptions about the underlying data generating process [23], providing a powerful forecasting method for linear or non-linear, non-parametric, data driven modelling. In univariate forecasting MLP is used similarly to a regression model, capable of using as inputs a set of lagged observations of the time series to predict its next value [7]. Data are presented to the network as a sliding window over the time series history. The NN tries to learn the underlying data generation process during training so that valid forecasts are made when new input values are provided [24]. In this analysis single hidden layer NN are used, based on the proof of universal approximation [22]. The general function of these networks is

$$f(X,w)=\beta_0+\sum_{h=1}^{H}\beta_h g\left(\gamma_0+\sum_{i=0}^{I}\gamma_{hi}x_i\right). \qquad (5)$$

$X = [x_0, x_1, \ldots, x_n]$ 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 $g(\cdot)$ is a non-linear transfer function [25]. In this analysis the hyperbolic tangent (tanh) transfer function is used, which is frequently used for NNs [26].

D. *Seasonal Encodings for Neural Networks*

We develop and compare MLPs that code the deterministic seasonality of the time series using the seven alternative encodings described in section II.B and C.

The common coding of deterministic seasonality through monthly seasonal dummy variables is implemented in models *Bin11* and *Bin12* which use S-1=11 and S=12 seasonal binary dummy variables respectively. The integer dummy variable representation uses only a single integer explanatory variable that repeats values from 1 to 12 (named *Int*). The trigonometric representation is modelled through the use of two dummy variables, one for $\sin(2\pi t/12)$ and one for $\cos(2\pi\tau/12)$ and is named *SinCos*. Finally, seasonal indices for the time series are identified by calculating the average value for each period of the season in the training set. This is an adequate estimation since the time series exhibit no trend or irregularities. The seasonal indices are repeated to create an explanatory variable which is then used as the only input to the MLP model *SIndex.* No time series lags are used for these models.

To model deterministic seasonality as stochastic process a univariate MLP model is used that employs lag t-1 and t-12 (named *AR*). To model seasonality as a seasonal unit root process the time series is used after seasonal integration, replicating the common approach of removing seasonality prior to inputting the time series to the MLP. No lags are used and the correct level is estimated by the MLP by assigning the correct weights to the bias terms in the different nodes (named *SRoot*). An overview of the inputs for each model is provided in table I.

TABLE I
Summary of MLP Inputs

Model	Lags*	Explanatory variables**	No of inputs
AR	1, 12	-	2
Bin11	-	11 Seasonal Dummies	11
Bin12	-	12 Seasonal Dummies	12
Int	-	Integer Dummy [1,2...12]	1
SinCos	-	$\sin(2\pi t/12)$, $\cos(2\pi t/12)$	2
Sindex	-	Seasonal Indices	1
SRoot	-***	-	0

* The Lags specify the time lagged realisations t-n used as inputs; ** For all explanatory variables only the contemporary lag is used; *** Time series is modelled after seasonal integration, i.e. $\Delta_S y_t$.

All the other parameters of the MLPs are held constant for all models. This allows attributing any differences in the performance of the models solely to the differences in modelling seasonality. All MLP use a single hidden layer with six hidden nodes and are trained using the Levenberg-Marquardt algorithm. This algorithm requires setting the μ_{LM} and its increase and decrease steps. Here $\mu_{LM}=10^{-3}$, with an increase step of $\mu_{inc}=10$ and a decrease step of $\mu_{dec}=10^{-1}$. For a detailed description of the algorithm and the parameters see [27]. The maximum training epochs are set to 1000. The training can stop earlier if μ_{LM} becomes equal of greater than $\mu_{max}=10^{10}$ or the validation error increases for more than 50 epochs. This is done to avoid over-fitting. When the training is stopped the network weights that give the lowest validation error are used. Each MLP is initialised 50 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 error for each time series on the validation dataset is selected to predict all values of the test set. Lastly, the time series and all explanatory variables that are not binary are linearly scaled between [-0.5, 0.5].

E. *Statistical Benchmark Methods*

Any empirical evaluation of time series methods requires the comparison against established statistical benchmark methods, in order to assess the increase in accuracy and its contribution to forecasting research (a fact often overlooked in NN experiments [11]). In this analysis we use seasonal exponential smoothing models (*EXSM*) with appropriately selected additive seasonality as a benchmark. The smoothing parameters are identified by optimising the one step ahead in-sample mean squared error. This model is selected as a benchmark due to its proven track record in univariate time series forecasting [8]. For more details on exponential smoothing models and the guidelines that were used to implement them in this analysis see [28].

IV. RESULTS

A. *Nonparametric MLP Comparisons*

All competing MLPs are tested for statistically significant differences using the Friedman and the post-hoc Nemenyi tests based on the mean rank of the errors. As MAE and sMAPE provided the same ranking, and both tests provided consistent results regardless of error measure, the results using a single measure are provided in table II.

The Friedman test indicates that across all time series, across different noise levels and for all time series separately there are statistically significant differences among the MLP models. Inspecting the results of the Nemenyi tests in table II we get a more detailed view on the ranking of each individual model and among which models there are statistically significant differences. It can be observed that across all different noise levels and across all time series the *SIndex* outperforms all other models with a statistically significant difference from the second best model. In all cases the *Bin11* and *Bin12* perform equally with no statistically significant differences both ranking second after *SIndex*. When all time series or only the no and low noise time series are considered, the *SinCos* has no statistically significant differences with the seasonal binary dummies *Bin11* and *Bin12* models. For the case of medium and high noise time series the *SinCos* ranks third after the *SIndex* and seasonal binary dummy variables models. This demonstrates that although the *SinCos* model is not equivalent to the trigonometrical representation of deterministic seasonality as expressed in (2) it is able to approximate it and in many cases with no statistically significant differences from the equivalent seasonal dummy coding. Furthermore, this representation is S/4 times more economical in inputs compared to (2) and S-2 and S-1 inputs more economical from (1) or *Bin11* and *Bin12* respectively. For the low, medium and high noise the *Int* model follows fifth in ranking. Although this model performs worse than the previous seasonality encodings it still outperforms the misspecified seasonal models *AR* and *SRoot*. This is not true for the no noise time series, which also affects the overall ranking across time series as well. The *AR* model follows. This demonstrates that it is better to code the deterministic seasonality through explanatory dummy variables, than as an autoregressive process, as it would be fitting for stochastic seasonality. Furthermore, in agreement to the discussion in II.C, removing the seasonality through seasonal integration, as in *SRoot*, performs poorly and ranks last in most cases. The reason for this is that the NN are not able to estimate directly the m_s and $\Delta_S y_t$ is overdifferenced. Note that in the case of no noise all models with the exception of *Int* are able to capture the seasonality perfectly with no error.

TABLE II
Summary of MLP nonparametric comparisons

Time series	All	No noise	Low noise	Medium noise	High noise
Friedman p-value	0.000	0.000	0.000	0.000	0.000
Nemenyi Mean Model Rank*⅄					
AR	1521.7	**<u>330.0</u>**	520.3	522.6	555.0
Bin11	**<u>1300.5</u>**	**<u>330.0</u>**	**<u>251.1</u>**	**<u>234.9</u>**	**<u>239.3</u>**
Bin12	**<u>1302.3</u>**	**<u>330.0</u>**	**<u>247.5</u>**	**<u>238.5</u>**	**<u>246.3</u>**
Int	1548.9	473.5	461.3	422.2	346.0
SinCos	**<u>1310.0</u>**	**<u>330.0</u>**	**<u>254.8</u>**	250.7	257.2
Sindex	1241.1	**<u>330.0</u>**	143.6	162.3	182.5
SRoot	1579.0	**<u>330.0</u>**	575.0	622.3	627.3
Nemenyi Ranking at 5% significance level*					
AR	5	**<u>1</u>**	6	6	6
Bin11	**<u>2</u>**	**<u>1</u>**	**<u>2</u>**	**<u>2</u>**	**<u>2</u>**
Bin12	**<u>2</u>**	**<u>1</u>**	**<u>2</u>**	**<u>2</u>**	**<u>2</u>**
Int	6	7	5	5	5
SinCos	**<u>2</u>**	**<u>1</u>**	**<u>2</u>**	4	4
Sindex	1	**<u>1</u>**	1	1	1
SRoot	7	**<u>1</u>**	7	7	7

* In each column MLP with no statistically significant differences under the Nemenyi test at 5% significance are **<u>underlined</u>**; ⅄ the critical distance for the Nemenyi test at 1% significance level is 10.5, at 5% significance level is 9.0 and at 10% significance level is 8.2.

It is apparent that the best method to model the deterministic seasonality is to use the seasonal indices as an explanatory variable for the MLP. Not only this method performs best with statistically significant difference from the second best model, but also it is very parsimonious, requiring a single input to model the deterministic seasonality, as it can be seen in table I. This is an important finding, since this coding of seasonality is not widely used, although it is a direct extension of (1).

B. *Comparisons against Benchmarks and Noise Level*

Taking advantage of the synthetic nature of the time series we can compare directly the error of each forecasting model with the artificially introduced error level and derive how close is each model to an ideal accuracy. The ideal accuracy is when the model's error is exactly equal to the noise, since that would mean that the model has captured perfectly the data generating process and ignores completely the randomness. On the other hand, a lower error than the noise level would imply possible overfitting to randomness. The comparison is done in MAE for each time series individually. The results are presented in fig. 2. Moreover the benchmark accuracy in MAE for each time series is provided in the same figure.

In fig. 2 it is clear than when there is no noise, for both seasonal patterns, all MLP models and the benchmark forecast the time series perfectly with zero error. Comparing the MLP models to the benchmark the misspecified *AR* and *SRoot* models perform worse than *EXSM*, with the *SRoot* model performing consistently last. This demonstrates that for the case of deterministic seasonality deseasonalising the time series, here through seasonal integration, hinders the NN to forecast the time series accurately. For both seasonal patterns for the low noise time series 2 and 6 all MLP perform worse than the benchmark. The opposite is true for the *Bin11, Bin12, Int, SinCos* and *SIndex* MLP models for the higher noise level time series. This implies that NN perform better than the statistical benchmark in high noise time series, being able to capture the true data generating process better.

Comparing the models accuracy with the known error due to noise it is interesting that all the MLP models, with the exception of the misspecified *AR* and *SRoot*, for all time series are very close to the ideal accuracy, i.e. having error only due to randomness. Note that for the validation set, on which the best performing initialisation for each of the NN models was chosen, their error is practically only due to noise. The benchmark error consistently increases as the noise level increases. For the case of low noise time series *EXSM* manages to forecast the time series with the error being solely due to randomness, implying a very good fit to the data generating process. The results are consistent across both seasonal patterns.

TABLE III
Summary sMAPE across all time series

Model	Training subset	Validation subset	Test subset
AR	<u>1.90%</u>	<u>1.94%</u>	<u>1.72%</u>
Bin11	1.60%	1.59%	1.45%
Bin12	**1.58%**	**1.58%**	1.46%
Int	1.62%	1.61%	1.49%
SinCos	1.59%	1.59%	1.47%
Sindex	1.60%	**1.58%**	**1.44%**
SRoot	<u>2.36%</u>	<u>2.21%</u>	<u>1.91%</u>
EXSM	1.86%	1.68%	1.52%

The best performing model in each set is marked with bold numbers. The models that are outperformed by the EXSM benchmark are underlined

Evaluating the performance of all models across the three training, validation and test subsets the models perform consistently and there are no clues of overfitting to the training set and all models are able to generalise well on the test set. Using MAE it is impossible to aggregate the results across different time series. Instead, the scale independent sMAPE is used. Summary accuracy sMAPE figures for all time series are provided in table III.

The results are consistent with fig. 2. The *AR* and *SRoot* models are outperformed by the benchmark, which is turn is outperformed by all other MLP models. In agreement with the results in table II the *SIndex* model is overall the most accurate, followed by the *Bin12* and *Bin11*. Note that the small sMAPE figures imply that all the models managed to capture the seasonal profile in all the time series and a visual inspection of the forecasts would reveal very small if no differences at all. Finally, the overall error level seems to be different between the three subsets. This is due to the random noise. Although each set contains 160 observations, which simulates in total 40 years of data, it was not enough to ensure equal noise distribution across all subsets.

III. CONCLUSIONS

We have evaluated different methodologies to model time series with deterministic seasonality. By exploring the theoretical properties of deterministic seasonality we show that the current debate in literature on how to model seasonality with NN does not address the problem correctly, omitting the properties of the type of seasonality. Seven competing approaches to model deterministic seasonality are evaluated and compared against a seasonal exponential smoothing model. *SIndex*, which uses the estimated seasonal indices as input to a NN, outperforms all competing MLP models and the benchmarks with statistically significant differences. Moreover, this approach to model seasonality is very parsimonious, requiring only one additional input. This may have particularly significant implications for high frequency data with long seasonal periods, often resulting in a large dimensionality of the input vector that can cause problems in training NN models.

This study does not thoroughly address the issue of how to best estimate the seasonal indices. In this study it proved relatively easy to obtain a good estimation of the seasonal indices for all time series, while this may become more problematic in the presence of irregularities, trends and multiple seasonalities. For future evaluations it is important to evaluate the robustness of the findings with different approaches to estimate the seasonal indices. Similarly, this study should be extended to more time series patterns and empirical time series that exhibit deterministic and stochastic seasonality, which will allow us to validate the findings and provide a reliable solution for practical applications.

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, no. 1, pp. 35-62, Mar, 1998.

[2] 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, pp. 425-434, Jul-Sep, 2005.

[3] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators " *Neural Networks,* vol. 2, no. 5, pp. 359-366, 1989.

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

[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, pp. 381-396, Apr, 2001.

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

[7] N. Kourentzes, and S. F. Crone, "Automatic modelling of neural networks for time series prediction – in search of a uniform methodology across varying time frequencies."

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

[9] J. S. Armstrong, "Findings from evidence-based forecasting: Methods for reducing forecast error," *International Journal of Forecasting,* vol. 22, no. 3, pp. 583-598, 2006.

[10] J. L. Callen, C. C. Y. Kwan, P. C. Y. Yip *et al.*, "Neural network forecasting of quarterly accounting earnings," *International Journal of Forecasting,* vol. 12, no. 4, pp. 475-482, Dec, 1996.

[11] 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, pp. 481-495, Sep-Nov, 1998.

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

[13] M. Nelson, T. Hill, W. Remus *et al.*, "Time series forecasting using neural networks: Should the data be deseasonalized first?," *Journal of Forecasting,* vol. 18, no. 5, pp. 359-367, Sep, 1999.

[14] G. P. Zhang, and M. Qi, "Neural network forecasting for seasonal and trend time series," *European Journal of Operational Research,* vol. 160, no. 2, pp. 501-514, Jan, 2005.

[15] D. R. Osborn, S. Heravi, and C. R. Birchenhall, "Seasonal unit roots and forecasts of two-digit European industrial production," *International Journal of Forecasting,* vol. 15, no. 1, pp. 27-47, 1999.

[16] A. Matas-Mir, and D. R. Osborn, "Does seasonality change over the business cycle? An investigation using monthly industrial production series," *European Economic Review,* vol. 48, no. 6, pp. 1309-1332, 2004.

[17] E. Ghysels, and D. R. Osborn, *The Econometric Analysis of Seasonal Time Series*, Cambridge: Cambridge University Press, 2001.

[18] A. H. Kvanli, R. J. Pavur, and K. Keeling, *Introduction to Business Statistics*, 6th ed., Mason, Ohio: Thomson/South-Western, 2002.

[19] 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." pp. 195-205.

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

[21] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.,* vol. 7, pp. 1-30, 2006.

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

[23] 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, pp. 666-680, Aug, 2001.

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

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

[26] T. P. Vogl, J. K. Mangis, A. K. Rigler *et al.*, "Accelerating the convergence of the backpropagation method," *Biological Cybernetics,* vol. 59, pp. 257-263, 1988.

[27] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, Boston: PWS Publishing, 1996.

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

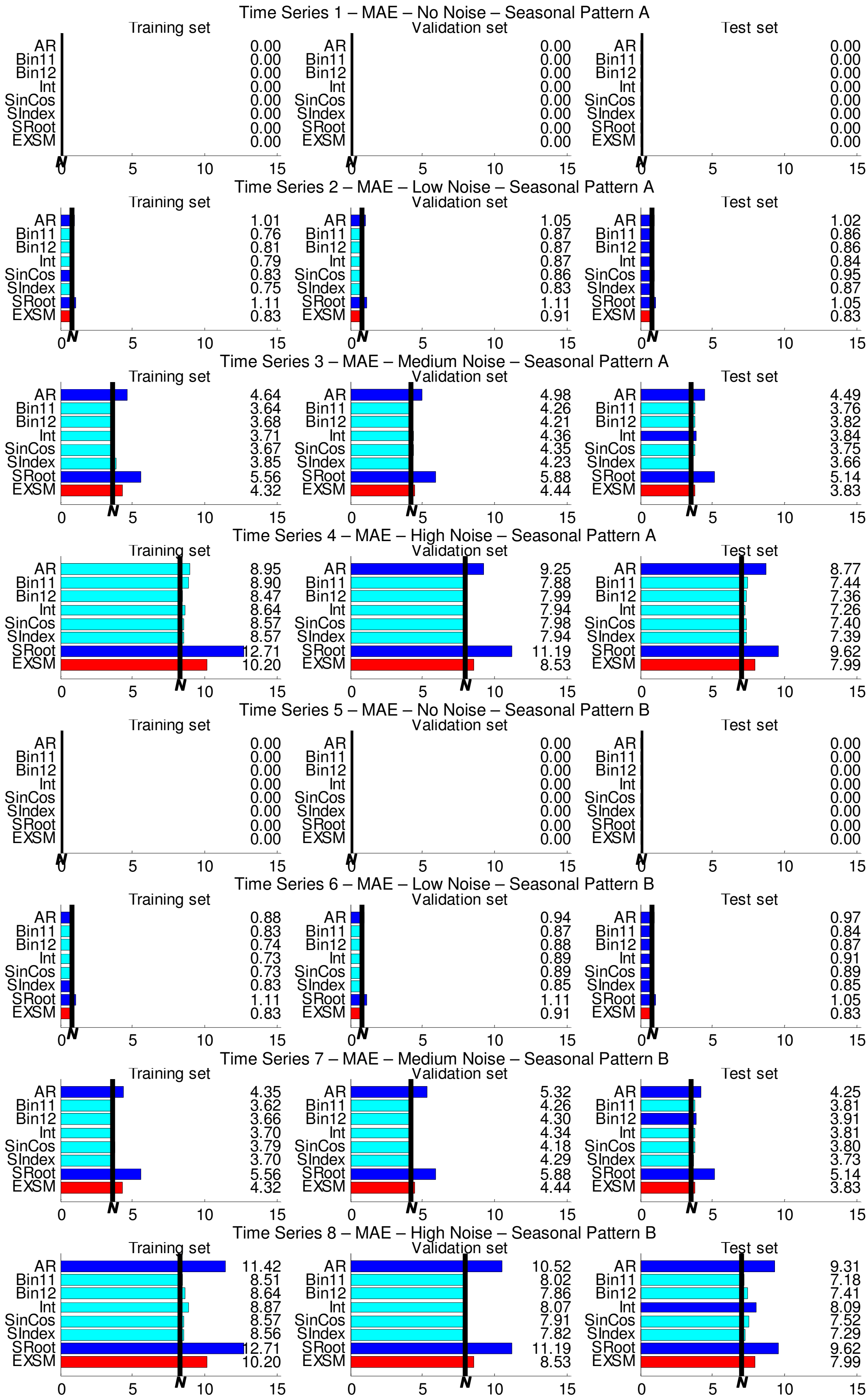

Fig. 2. MAE for each time series for each subset for all models. The noise level is marked by a thick black vertical line. Light coloured bars are models which are better than the benchmark (*EXSM*). The value of each error is provided at the right side of each row.